Transportation Research Part C 96 (2018) 96–121

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Estimating multi-year 24/7 origin-destination demand using high-granular multi-source traffic data

Wei Ma[a], Zhen (Sean) Qian[a,b,*]

[a] Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States
[b] H. John Heinz III Heinz College, Carnegie Mellon University, Pittsburgh, PA 15213, United States

A B S T R A C T

Dynamic origin-destination (OD) demand is central to transportation system modeling and analysis. The dynamic OD demand estimation problem (DODE) has been studied for decades, most of which solve the DODE problem on a typical day or several typical hours. There is a lack of methods that estimate high-resolution dynamic OD demand for a sequence of many consecutive days over several years (referred to as 24/7 OD in this research). Having multi-year 24/7 OD demand would allow a better understanding of characteristics of dynamic OD demands and their evolution/trends over the past few years, a critical input for modeling transportation system evolution and reliability. This paper presents a data-driven framework that estimates day-to-day dynamic OD using high-granular traffic counts and speed data collected over many years. The proposed framework statistically clusters daily traffic data into typical traffic patterns using t-Distributed Stochastic Neighbor Embedding (t-SNE) and k-means methods. A GPU-based stochastic projected gradient descent method is proposed to efficiently solve the multi-year 24/7 DODE problem. It is demonstrated that the new method efficiently estimates the 5-min dynamic OD demand for every single day from 2014 to 2016 on I-5 and SR-99 in the Sacramento region. The resultant multi-year 24/7 dynamic OD demand reveals the daily, weekly, monthly, seasonal and yearly change in travel demand in a region, implying intriguing demand characteristics over the years.

1. Introduction

The increasing complexity and inter-connectivity of mobility systems call for large-scale deployment of dynamic network models that encapsulate traffic flow evolution for system-wide decision making. As an indispensable component of dynamic network models, time-dependent Origin-Destination (OD) demand plays a key role in transportation planning and management. Obtaining accurate and high-resolution time-dependent OD demand is notoriously difficult, though the dynamic OD estimation (DODE) problem has been intensively studied for decades. A number of DODE methods have been proposed, most of which aim at estimating dynamic OD demand for a typical day or even several hours on a typical day. To our best knowledge, there is a lack of research estimating dynamic OD demand for a long time period over the years. The OD demand and its behavior, though are generally repetitive in an aggregated view, can vary from day to day. The day-to-day variation of OD demand would need to be considered in estimate OD demand for a long period of many consecutive days. For example, estimating the dynamic OD demand for every 5-min in an entire year is computationally implausible using most of the existing DODE methods. In view of this, this paper presents an efficient data-driven approach to estimate time-dependent OD demand using high-granular traffic flow counts and traffic speed data collected over many years.

Dynamic OD demand represents the number of travelers departing from an origin at a particular time interval heading for a destination. It reveals traffic demand level, and is critical input for estimating and predicting network level congestion in a region. In addition, policymakers can understand the travelers' departure patterns and daily routines through the day-to-day OD demand. As a result, many Advanced Traveler Information Systems/Advanced Traffic Management Systems (ATIS/ATMS) require accurate time-

* Corresponding author at: Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, United States.
E-mail addresses: weima@cmu.edu (W. Ma), seanqian@cmu.edu (Z.S. Qian).

https://doi.org/10.1016/j.trc.2018.09.002
Received 6 March 2018; Received in revised form 2 September 2018; Accepted 2 September 2018
Available online 22 September 2018
0968-090X/ © 2018 Elsevier Ltd. All rights reserved.

dependent OD demand as an input. A tremendous number of studies estimate time-dependent OD demand using observed traffic data which includes traffic counts, probe vehicle data and Bluetooth data. Oftentimes those data collected over multiple days are taken daily average before being input to dynamic network models, which represent the average traffic pattern and OD demand on a typical day.

With the development of cutting-edge sensing technologies, many traffic data can be collected in high spatial and temporal granularity at a low cost. For example, traffic count and traffic speed for a road segment of 0.1 mile can be sensed and updated every 5 min throughout the year. This is a $12 \times 24 = 288$ dimension of counts/speed data for a single road segment on one day. Most of existing DODE methods become computationally inefficient or even implausible when dealing with large-scale networks with thousands of observed road segments and thousands of days of high dimensional data. How to efficiently obtain high-resolution OD demand on a daily basis over many years remains technically challenging. In this research, we estimate high-resolution dynamic OD demand for a sequence of many consecutive days over several years, referred to 24/7 OD demand throughout this paper.

Dynamic OD estimation (DODE) was formulated as either a least square problem or a state-space model. Cascetta et al. (1993) extended the concepts of static OD estimation problem and formulated a generalized least square (GLS) based framework for estimating dynamic OD demands. Tavana (2001) proposed a bi-level optimization framework which solves for a GLS problem in the upper level with a dynamic traffic assignment (DTA) problem in the lower level. The bi-level formulations for OD estimation problem were also discussed by Nguyen (1977), LeBlanc and Farhangian (1982), Fisk (1989), Yang et al. (1992), Florian and Chen (1995), Jha et al. (2004) for static OD demand. Zhou et al. (2003) extended the bi-level formulation to incorporate multi-day traffic data. To implement efficient estimation algorithms on real-time traffic management systems, Bierlaire and Crittin (2004) proposed a least square based real-time OD estimation/prediction framework for large-scale networks. Zhou and Mahmassani (2007), Ashok and Ben-Akiva (2000) established a state-space model for real-time OD estimation based on on-line traffic data feeds. Hazelton (2008) built a statistical inference framework using Markov chain Monte Carlo algorithm for generating posterior OD demand.

The bi-level OD estimation framework can be solved using heuristically computed gradient, convex approximation or gradient free algorithms. Yang (1995) proposed two heuristic approaches for the bi-level OD estimation problem, the iterative estimation-assignment (IEA) algorithms and sensibility-analysis based algorithm (SAB). Josefsson and Patriksson (2007) further improved the sensitivity analysis procedures adopted in SAB process. A Dynamic Traffic Assignment (DTA) simulator is also used to determine the numerical derivatives of link flows. Balakrishna et al. (2008), Cipriani et al. (2011) fitted such an estimation process into a stochastic perturbation simultaneous approximation (SPSA) framework. Lee and Ozbay (2009), Vaze et al. (2009), Ben-Akiva et al. (2012), Lu et al. (2015), Tympakianaki et al. (2015), Antoniou et al. (2015) further enhanced the SPSA based methods. Verbas et al. (2011) compared different gradient based methods to solve the bi-level formulation of DODE problem. Flötteröd et al. (2011) proposed a Bayesian framework that calibrates the dynamic OD using agent-based simulators. In addition to numerical solutions, research has been looking into computing the analytical derivatives for the lower-level formulations (Ghali and Smith, 1995; Frederix et al., 2011; Qian et al., 2012; Qian and Zhang, 2011). Other machine learning and computational technologies are also employed to enhance the efficiency of OD estimation methods (Kim et al., 2001; Kattan and Abdulhai, 2006; Huang et al., 2012; Xu et al., 2014).

The general bi-level formulation for OD estimation is proved to be non-continuous and non-convex, and thus its scalability is limited. Nie and Zhang (2008, 2010) formulated a single-level static and dynamic OD estimation framework that incorporates User Equilibrium (UE) path flows solved by the variational inequality, which is further improved by Shen and Wynter (2012) under the static cases. Recently, Lu et al. (2013) formulated a Lagrangian relaxation-based single-level non-linear optimization to estimate dynamic OD demand.

A large number of data sources are feeding to DODE methods. Zhang et al. (2008) evaluated the roles of count data, speed data and history OD data in the effectiveness of DODE. Van Der Zijpp (1997), Antoniou et al. (2004), Zhou and Mahmassani (2006), Rao et al. (2018) used automated vehicle identification (AVI) data together with flow counts to estimate dynamic OD demand. Emerging technologies such as Bluetooth (Barceló et al., 2010), mobile phone location (Calabrese et al., 2011; Iqbal et al., 2014), probe vehicles (Antoniou et al., 2006) data were also employed to estimate dynamic OD demands.

Two important issues are yet to be addressed. Firstly, many existing DODE methods (Ashok and Ben-Akiva, 2000; Josefsson and Patriksson, 2007; Nie and Zhang, 2008; Lu et al., 2013; Lu et al., 2015) require a dynamic traffic loading (DNL) process (either microscopic or mesoscopic) to endogenously encapsulate the traffic flow evolution and congestion spillover. As the DNL process requires relatively high computational budget, it can take hours to estimate dynamic OD demand on a network of thousands of links/nodes for a single day. Not only does it have hard time converging under the data fitting optimization problem, but estimating the 24/7 OD demand for several years becomes computationally impractical. The other issue is that most studies estimate OD demand for a few hours or a single day. OD demand varies from day to day, but is also repetitive to some extent. The day-to-day features of OD demand has not be taken into consideration of the DODE methods. For this reason, demand patterns that evolve daily, weekly, monthly, seasonally and yearly have not been explored, despite of high-granular data collected over many years.

In this paper, we develop a data-driven framework that estimates multi-year 24/7 dynamic OD demand using traffic counts and speed data collected over the years. The framework builds the relationship between dynamic OD demand and traffic observations using link/path indices matrix, dynamic assignment ratio (DAR) matrix, and route choice matrix. These three matrices enable the estimate framework to circumvent the bi-level formulation, since each of the matrices can be directly calibrated using high-granular real-world data rather than from complex simulation. The proposed framework utilizes data-driven approaches to explore the daily, weekly, monthly and yearly traffic patterns, and group traffic data into different patterns. The proposed estimation framework is computational efficient: 5-min dynamic OD demand for three years can be estimated within hours on an inexpensive personal computer.

In order to address computation issues, this paper uses a Graphics Processing Unit (GPU) which is currently attracting tremendous

research interests from various fields. Neural network models can be performed more deeply and widely (Szegedy et al., 2015) with GPU computing. It is also widely used in probabilistic modeling (Srivastava and Salakhutdinov, 2012) and finite element methods (Lu et al., 2014). To our best knowledge, this paper is among the first to design and implement GPU computing in the DODE method, since the traditional DODE methods are not suitable for GPU computing. We present a stochastic gradient projection method that well suits the GPU computing framework. As we will show in the case study, the proposed GPU friendly method is over 10 times more efficient than the state-of-art CPU based method. The implies that GPU computing makes possible to make full use of the massive traffic data comparing to traditional models.

The main contributions of this paper are summarized as follows:

(1) It proposes a framework for estimating multi-year 24/7 dynamic OD demand using high-granular traffic flow counts and speed data. It takes into account day-to-day features of flow patterns by defining and calibrating the dynamic assignment ratio (DAR) matrix using real-world data, which enables realistic representation and efficient computing of network traffic flow.
(2) It adopts t-SNE and k-means methods to cluster daily traffic data collected over many years into several typical traffic patterns. The clustering helps better understand typical daily demand patterns and improve the DODE accuracy.
(3) It proposes a stochastic projected gradient descent method to solve the DODE problem. The proposed method is suitable for GPU computation, which enables efficiently estimating high-dimensional OD over many years.
(4) A numerical experiment on a large-scale network with real-world data is conducted. 5-min dynamic OD demands for every day from 2014 to 2016 are efficiently estimated. As a result, OD demand evolution over the years can be presented and analyzed.

The remainder of this paper is organized as follows. Section 2 discusses the formulation. Section 3 presents the solution algorithm for the proposed framework. Section 4 proposes the entire DODE framework. In Section 5, a real-world experiment for estimating 5-min dynamic OD from 2014 to 2016 on a regional Sacramento Network is presented. Finally, conclusions are drawn in Section 6.

2. The model

In this section, we present a framework that utilizes the high-granular traffic counts and speed data to estimate 24/7 dynamic OD. We first model and discretize continuous-time traffic flow evolution on general networks. The dynamic assignment ratio (DAR) matrix is proposed to characterize the traffic flow evolution in discrete time. Unsupervised dimension reduction and clustering methods are adopted to group data of multiple years into several typical traffic patterns. We use the Logit-based route choice model to characterize travelers' behavior in each cluster. Finally, we formulate the DODE as a high-dimensional non-negative least square (NNLS) problem and propose an efficient solution algorithm.

Table 1
List of notations.

A	The set of all links
A^o	The set of links with flow observations
K_q	The set of all OD pairs
K_{rs}	The set of all paths between OD pair rs
δ_{rs}^{ka}	Path/link incidence for kth path in OD pair rs and link a
	Variables in continuous time
t_1	The departure time of path flow or OD flow
t_2	The arrival time at the tail of link
T_1	The set of all possible departure time from any path and link
T_2	The set of all possible arrival time at all links
$f_{rs}^k(t_1)$	The kth path flow rate for OD pair rs at time t_1
$x_a(t_2)$	The flow rate at the tail of link a at time t_2
$q_{rs}(t_1)$	The flow rate of OD pair rs at time t_1
$c_{rs}^k(t_1)$	The path cost for path k for OD pair rs departing at time t_1
$p_{rs}^k(t_1)$	The portion of choosing path k in all paths between OD pair rs at time t_1
	Variables in discrete time
h_1	The index of departure time interval of path flow or OD flow
h_2	The index of arrival time interval at the tail of link
$\bar{f}_{rs}^{kh_1}$	The kth path flow rate for OD pair rs in time interval h_1
$\bar{x}_a^{h_2}$	The flow rate at the tail of link a in time interval h_2
$\bar{q}_{rs}^{h_1}$	The flow rate of OD pair rs in time interval h_1
$\bar{p}_{rs}^{kh_1}$	The portion of choosing path k in all paths between OD pair rs in time interval h_1
$\rho_{rs}^{ka}(h_1, h_2)$	The portion of the kth path flow departing within time interval h_1 between OD pair rs which arrives at link a within time interval h_2 (namely, an entry of the DAR matrix)

2.1. Notations

Please refer to Table 1. The hat symbol, $\hat{}$, indicates the variable is an estimator for the true (unknown) variable.

2.2. Model the continuous time traffic flow

Before proposing the estimation method, we first formulate the model for continuous time traffic flow on general networks. We denote the path flow $f_{rs}^k(t_1)$ as the kth path flow rate for OD pair rs at time t_1 and link flow $x_a(t_2)$ as the flow rate at the tail of link a at time t_2. The relationship between path flow and link flow is presented by Eq. (1).

$$\begin{aligned} x_a(t_2) &= \int_{t_1 \in T_1} \left(\sum_{rs \in K_q} \sum_{k \in K_{rs}} \delta_{rs}^{ka}(t_1, t_2) f_{rs}^k(t_1) \right) dt_1 \\ &= \sum_{rs \in K_q} \sum_{k \in K_{rs}} \int_{t_1 \in T_1} \delta_{rs}^{ka}(t_1, t_2) f_{rs}^k(t_1) dt_1 \end{aligned} \tag{1}$$

where K_q is the set of all OD pairs, and K_{rs} is the path set for OD pair rs. T_1 is the set of possible departure time for any path and link. In this paper we always denote departure time of path flow or OD flow as t_1, and the arrival time at the tail of link as t_2, respectively. The time-dependent path/link indices matrix $\delta_{rs}^{ka}(t_1, t_2)$ is defined as follows:

$$\delta_{rs}^{ka}(t_1, t_2) = \begin{cases} 1 & \text{if path flow } f_{rs}^k(t_1) \text{ arrives at the tail of link } a \text{ at time } t_2 \\ 0 & \text{else} \end{cases} \tag{2}$$

Assuming the traffic flow is FIFO (First-In-First-Out) and continuous, the arrival time of all departure flows can be determined explicitly. Therefore, the time-dependent path/link indices matrix can be simplified as in Eq. (3).

$$\delta_{rs}^{ka}(t_1, t_2) = \begin{cases} \delta_{rs}^{ka} & \text{if } t_1 = \tau_{rs}^{ka}(t_2) \\ 0 & \text{else} \end{cases} \tag{3}$$

where δ_{rs}^{ka} is 1 if path k for OD pair rs passes link a and 0 otherwise. $\tau_{rs}^{ka}(\cdot)$ is the departure time function for kth path in OD rs, and $\tau_{rs}^{ka}(t_2)$ is the departure time of kth path in OD pair rs arriving at the tail of link a at t_2, $\tau_{rs}^{ka}(t_2) \in T_1$. Combining Eqs. (1) and (3) by replacing the time-dependent path/link indices matrix with a static path/link indices matrix, the relationship between link flow and path flow can be formulated as Eq. (4).

$$x_a(t_2) = \sum_{rs \in K_q} \sum_{k \in K_{rs}} \delta_{rs}^{ka} f_{rs}^k(\tau_{rs}^{ka}(t_2)) \tag{4}$$

Example 1 (*Link flow and path flow*). Consider a two-link network presented in Fig. 1. The path flow is $f_1(t)$, and the link flow for link 1 and 2 are $x_1(t)$ and $x_2(t)$, respectively. The travel time to traverse link 1 is constantly Δt. Then at the starting time t_0, we have

$$x_1(t_0) = f_1(t_0) \tag{5}$$

$$x_2(t_0) = 0 \tag{6}$$

After Δt, we have

$$x_1(t_0 + \Delta t) = f_1(t_0 + \Delta t) \tag{7}$$

$$x_2(t_0 + \Delta t) = f_1(t_0) \tag{8}$$

2.3. Objective function in discrete time

The objective function of DODE problem computes the ℓ^2 norm between the observed link flow $x_a(t_2)$ and the estimated link flow $\hat{x}_a(t_2)$. The estimated link flow is aggregated by the estimated path flows $\hat{f}_{rs}^k(t_1)$, then the optimization problem is presented in Eq. (9).

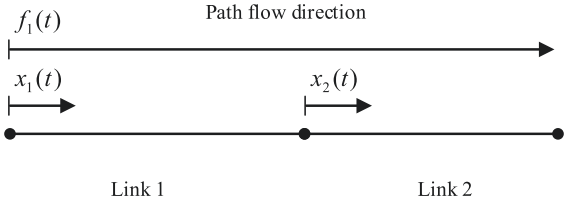


Fig. 1. Example of link flow and path flow.

$$\min_{\{\hat{f}_{rs}^k(\cdot)\}_{r,s,k}} \sum_{a\in A} \int_{t_2\in T_2} \|x_a(t_2)-\hat{x}_a(t_2)\|_2^2 dt_2$$
$$\text{s.t.} \quad \hat{f}_{rs}^k(t_1) \geqslant 0 \qquad \forall\, t_1 \in T_1, \forall\, rs \in K_q, \forall\, k \in K_{rs} \tag{9}$$

where T_2 is the set of possible arrival time for all links, which is usually the observation time period for all links. Eq. (9) formulates the objective function on the link set A, we can use the observed link set A^o to replace A if only a subset of links are observed. Based on Eq. (4), we rewrite the objective function as Eq. (10).

$$L(x,\hat{x}) = \sum_{a\in A} \int_{t_2\in T_2} \left\| x_a(t_2) - \sum_{rs\in K_q} \sum_{k\in K_{rs}} \delta_{rs}^{ka} \hat{f}_{rs}^k(\tau_{rs}^{ka}(t_2)) \right\|_2^2 dt_2 \tag{10}$$

Typically, the data collected from traffic sensors are discretized in terms of time intervals. Therefore, the objective function needs to be discretized as well. We divide the entire time period $T_1 \cup T_2$ into N time intervals, and the sequence of time intervals is denoted as $\{H_h\}_{h=1}^N$. We further denote $t^h = \sup_t'\{t'|t' \leqslant t, \forall\, t \in H_h\}$, which represents the beginning of each time interval.

Example 2 (*Time interval discretization***).** In Fig. 2, we discretize the whole time period into 4 intervals. H_1, H_2, H_3, H_4 are the time intervals and t^1, t^2, t^3, t^4 are time points denoting the starting time of each time interval.

The discretized objective function is presented in Eq. (11).

$$\begin{aligned}
L(x,\hat{x}) &= \sum_{a\in A} \int_{t_2\in T_2} \left| x_a(t_2) - \sum_{rs\in K_q} \sum_{k\in K_{rs}} \delta_{rs}^{ka} \hat{f}_{rs}^k(\tau_{rs}^k(t_2)) \right|_2^2 dt_2 \\
&\overset{LargeN}{\simeq} \sum_{a\in A} \sum_{h_2=1}^{N} \left(\left| \int_{t_2\in H_{h_2}} x_a(t_2)dt_2 - \sum_{rs\in K_q} \sum_{k\in K_{rs}} \delta_{rs}^{ka} \int_{t_2\in H_{h_2}} \hat{f}_{rs}^k(\tau_{rs}^{ka}(t_2))dt_2 \right|_2^2 \right) \\
&= \sum_{a\in A} \sum_{h_2=1}^{N} \left(\left| \bar{x}_a^{h_2} - \sum_{rs\in K_q} \sum_{k\in K_{rs}} \delta_{rs}^{ka} \sum_{h_1=1}^{N} \left(\int_{t_1\in H_{h_1}\cap \tau_{rs}^{ka}(H_{h_2})} \hat{f}_{rs}^k(t_1)dt_1 \right) \right|_2^2 \right) \\
&= \sum_{a\in A} \sum_{h_2=1}^{N} \left(\left| \bar{x}_a^{h_2} - \sum_{rs\in K_q} \sum_{k\in K_{rs}} \delta_{rs}^{ka} \sum_{h_1=1}^{N} (\rho_{rs}^{ka}(h_1,h_2)\hat{\bar{f}}_{rs}^{kh_1}) \right|_2^2 \right)
\end{aligned} \tag{11}$$

where

$$\bar{x}_a^{h_2} = \int_{t_2\in H_{h_2}} x_a(t_2)dt_2 \tag{12}$$

$$\hat{\bar{f}}_{rs}^{kh_1} = \int_{t_1\in H_{h_1}} \hat{f}_{rs}^k(t_1)dt_1 \tag{13}$$

We denote $\tau_{rs}^{ka}(H_{h_2})$ as the range of function $\tau_{rs}^{ka}(\cdot)$ with domain being H_{h_2}, $\tau_{rs}^{ka}(H_{h_2}) = \{t_1|t_1 = \tau_{rs}^{ka}(t_2), \forall\, t_2 \in H_{h_2}\}$. The cumulative link flow $\bar{x}_a^{h_2}$ and cumulative estimated path flow $\hat{\bar{f}}_{rs}^{h_1k}$ are integrated from $x(t_2)$ and $\hat{f}_{rs}^k(t_1)$ over time interval H_{h_1} and H_{h_2}, respectively. The weight function $\rho_{rs}^{ka}(h_1, h_2)$ denotes the portion of the kth path flow departing within time interval h_1 between OD pair rs which arrive at link a within time interval h_2.

$$\rho_{rs}^{ka}(h_1,h_2) = \frac{\int_{t_1\in H_{h_1}\cap \tau_{rs}^{ka}(H_{h_2})} f_{rs}^k(t_1)dt_1}{\bar{f}_{rs}^{h_1k}} \tag{14}$$

We can use this weight function to trace the discretized path flow $\bar{f}_{rs}^{h_1k}$ to link a, as presented in Eq. (15).

$$\bar{x}_a^{h_2} = \sum_{rs\in K_q} \sum_{k\in K_{rs}} \delta_{rs}^{ka} \sum_{h_1=1}^{N} \rho_{rs}^{ka}(h_1,h_2)\bar{f}_{rs}^{kh_1} \tag{15}$$

It can be seen that the discretized objective function approaches to the continuous objective function when $N \to \infty$. The weight function ρ_{rs}^{ka} reflects the link-level flow progression from time interval h_1 to h_2. The flow progression and evolution aggregated at the link level can be captured by the time-varying link-level traffic speed and counts. However, its evolution within each link, such as within-link shockwave, can be hardly calibrated or learned unless trajectory level data are available. In fact, link-level flow evolution is proven to be realistic, stable and efficient (Jin, 2012). Thus, in this research, we assume vehicles on the network are evenly spread in space and link flow rate at the tail of each link within each time interval is also constant (evenly spread in time), resulting the weight function ρ_{rs}^{ka} presented in Eq. (16).

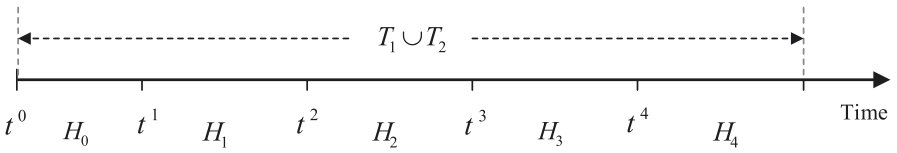


Fig. 2. Example of time interval discretization.

$$f_{rs}^{k}(t_1) = \frac{1}{|H_{h_1}|}\bar{f}_{rs}^{kh_1}, \forall\, t_1 \in H_{h_1} \tag{16}$$

The formulation (16) is further simpled using equal time intervals, as presented by $\Delta H := |H_h|, \forall\, h = 1, \cdots, n$. Then we are ready to present the dynamic assignment ratio (DAR) as in Eq. (18).

$$\rho_{rs}^{ka}(h_1, h_2) = \frac{|\tau_{rs}^{ka}(H_{h_2}) \cap H_{h_1}|}{|H_{h_1}|} \tag{17}$$

$$= \frac{|(\tau_{rs}^{ka})^{-1}(H_{h_1}) \cap H_{h_2}|}{|(\tau_{rs}^{ka})^{-1}(H_{h_1})|} \tag{18}$$

where $(\tau_{rs}^{ka})^{-1}(\cdot)$ is the inverse function of $\tau_{rs}^{ka}(\cdot)$ since $\tau_{rs}^{ka}(\cdot)$ is monotonically increasing based on the FIFO rule. $(\tau_{rs}^{ka})^{-1}(H_{h_1})$ represents the range of function $(\tau_{rs}^{ka})^{-1}$ with domain being H_{h_1}. For each path f_{rs}^{k}, Eq. (18) can be interpreted as the portion of vehicles arriving at link a in time interval h_2 among all the vehicles departing at interval h_1. As we assumed that the vehicles are spread evenly in time and space, the portion $\rho_{rs}^{ka}(h_1, h_2)$ can be computed either at departing time (17) or at arriving time (18). The DAR matrix is computed through the weight function $\rho_{rs}^{ka}(\cdot,\cdot)$.

Example 3 (*DAR matrix computation***).** As presented in Fig. 3, we demonstrate an example for computing the DAR matrix in a three link network. The path flow f_{rs}^{k} passes three links x_1, x_2, x_3 on the network. To compute non-zero entries of the DAR matrix with $h_1 = 1$, we derive the trajectories of path flow departing at time t^1 and t^2. The speeds of links are the slopes of the trajectory, which are denoted as ζ_1, ζ_2, ζ_1', ζ_2'. The probe vehicle speeds of links are available from various sources, such as HERE, INRIX and TomTom. We plot the two approximate trajectories of the leading vehicle departing from the origin at time t^1 and t^2, and measure the length of each time segment as ω_1, ω_2, ω_3, ω_4. Based on the definition of $(\tau_{rs}^{ka})^{-1}$, we have

$$|(\tau_{rs}^{k1})^{-1}(H_1)| = |H_1| \tag{19}$$

$$|(\tau_{rs}^{k2})^{-1}(H_1)| = \omega_1 + \omega_2 \tag{20}$$

$$|(\tau_{rs}^{k3})^{-1}(H_1)| = \omega_3 + |H_2| + \omega_4 \tag{21}$$

Then the DARs can be computed as follows based on Eq. (18).

$$\rho_{rs}^{k1}(1, 1) = 1 \tag{23}$$

$$\rho_{rs}^{k2}(1, 1) = \frac{\omega_1}{\omega_1 + \omega_2} \tag{24}$$

$$\rho_{rs}^{k2}(1, 2) = \frac{\omega_2}{\omega_1 + \omega_2} \tag{25}$$

$$\rho_{rs}^{k3}(1, 1) = \frac{\omega_3}{\omega_3 + |H_2| + \omega_4} \tag{26}$$

$$\rho_{rs}^{k3}(1, 2) = \frac{|H_2|}{\omega_3 + |H_2| + \omega_4} \tag{27}$$

$$\rho_{rs}^{k3}(1, 3) = \frac{\omega_4}{\omega_3 + |H_2| + \omega_4} \tag{28}$$

Given Eq. (18), the discrete time objective function is formulated as Eq. (29):

$$L(x, \hat{x}) \simeq \sum_{a \in A} \sum_{h_2=1}^{N} \left(\left\| \bar{x}_a^{h_2} - \sum_{rs \in K_q} \sum_{k \in K_{rs}} \sum_{h_1=1}^{N} \delta_{rs}^{ka} \rho_{rs}^{ka}(h_1, h_2) \hat{\bar{f}}_{rs}^{h_1 k} \right\|_2^2 \right) \tag{29}$$

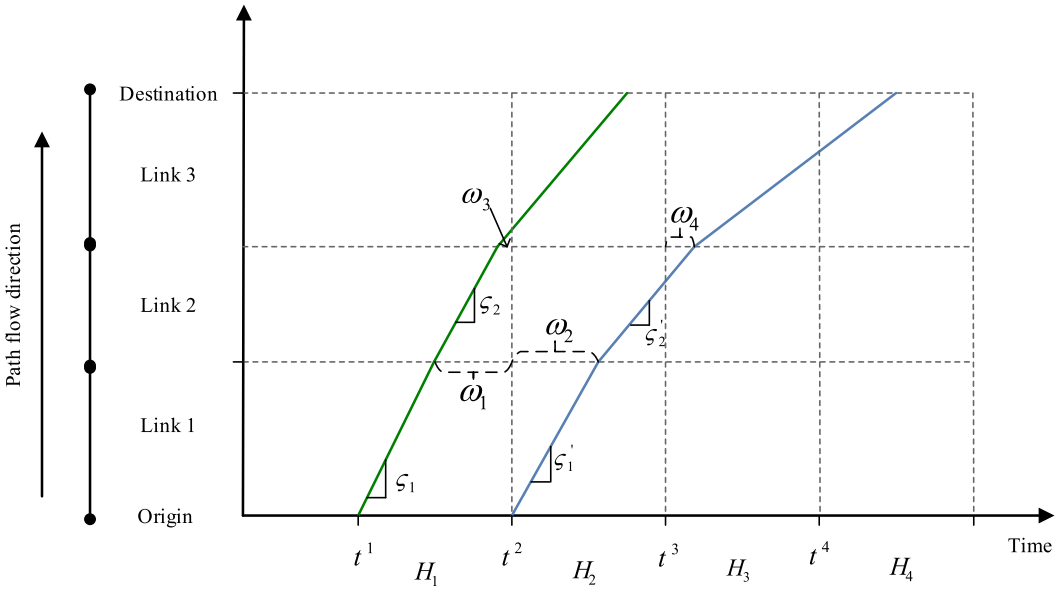


Fig. 3. Example of computing the DAR matrix.

2.4. Link/path travel time

In previous sections, we derive the objective function based on the DAR matrix. As shown in Example 3, the DARs are computed through ω_1, ω_2, ω_3, ω_4. These variables can be computed based on the link travel time, for example

$$\omega_1 = t^2 - (t^1 + c_1(t^1)) \tag{30}$$

In a general form, let $c_a(t)$ denote the travel time of link flow for a departing from the tail of link at time t. We denote $c_{rs}^k(t)$ as the travel time of path flow k in OD pair rs departing at time t. Let α_{rs}^k represent the sequence of links passed by flow f_{rs}^k, $\alpha_{rs}^k(a)$ represent the ath link in sequence α_{rs}^k , and β_{rs}^k represents the number of links passed by flow f_{rs}^k. Then $c_{rs}^k(t)$ can be calculated by Eq. (31).

$$c_{rs}^k(t_1) = c_{\alpha_{rs}^k(\beta_{rs}^k)}(c_{\alpha_{rs}^k(\beta_{rs}^k-1)}(\cdots(c_{\alpha_{rs}^k(1)}(t_1)))) \tag{31}$$

We note the link travel time can be obtained from either dynamic network loading models (traffic simulation) or the real-world data. In this research, we use the speed data from probe vehicles (such as INRIX or HERE) to circumvent the simulation process. The link/path travel time can be directly calibrated from the high-granular probe vehicle speed data.

2.5. Traffic pattern clustering

In the following sections, we will build the relationship between dynamic OD flow and dynamic path flow. Behavior models determine the route choice portions based on the traffic conditions and travelers perception errors, which are used to distribute OD flow onto different paths. Travelers' route choices are likely to be stable when traffic conditions are recurrent. In this research, we speculate that there exist several typical repetitive traffic conditions at the network level, each of which carries weekday/weekend, seasonal or other demand/supply characteristics. In each typical traffic pattern, we assume the network condition follows a statistical equilibrium defined by Ma and Qian (2017, 2018). Travelers will select their route based on the traffic pattern they observe historically, and their route choice portions remain stable for those days with the same typical traffic pattern. To estimate the route choice portions in each traffic pattern, we first cluster the traffic data into patterns using day-to-day traffic data in this section. Then the route choice portions for each pattern are estimated based on a generalized route choice model in the following section.

In addition to statistical equilibrium approach, the day-to-day traffic assignment model can also be used to utilize temporal correlation of traffic patterns, and the OD demand can be estimated by a filtering approach. One novelty that stems from the statistical equilibrium approach, to be further examined in the next step, is that the weekly/monthly/seasonal O-D variation can be learned directly from real-world data rather than being a prior to be imposed to the day-to-day dynamics model. In this paper we focus on the statistical equilibrium approach to modeling the temporal correlation of traffic patterns.

To cluster the traffic patterns, t-SNE (t-Distributed Stochastic Neighbor Embedding) is adopted to project high-dimensional traffic data points to low dimensional feature space. K-means method is then used to cluster the data points in the feature space. Each cluster obtained from k-means method represents traffic patterns under different traffic conditions.

2.5.1. Dimension reduction and data visualization

For a traffic state variable, e.g. link flow from all sensors on a network, we adopt state-of-art dimension reduction method t-SNE (t-Distributed Stochastic Neighbor Embedding) to project traffic state variables to low dimensional space. The dimension reduction process can significantly reduce the influence of noise and outliers to the clustering methods. The t-SNE method minimizes Kullback–Leibler divergence C between a joint probability distribution P in the high-dimensional space and a joint probability distribution Q in the low-dimensional space, as presented in Eq. (32).

$$C = KL(P||Q) = \sum_i \sum_j \mu_{ij} \log \frac{\mu_{ij}}{\nu_{ij}} \tag{32}$$

where i, j are the indices of the data. μ_{ij} and ν_{ij} measure the pair-wise similarity between data points, which are defined as:

$$\mu_{ij} = \frac{\exp(-||\chi_i - \chi_j||^2/2\sigma^2)}{\sum_{i' \neq j'} \exp(-||\chi_{i'} - \chi_{j'}||^2/2\sigma^2)} \tag{33}$$

$$\nu_{ij} = \frac{(1 + ||\psi_i - \psi_j||^2)^{-1}}{\sum_{i' \neq j'} (1 + ||\psi_{i'} - \psi_{j'}||^2)^{-1}} \tag{34}$$

where χ_i are data points from original high-dimensional space and ψ_i are data points from low-dimensional space that we want. ψ_i is assumed to follow a Student t-distribution with one degree of freedom as one heavy-tailed distribution in low-dimensional space. The computational and space complexity of t-SNE are $\emptyset(n^2)$, but it can be efficiently solved using stochastic gradient descent (SGD) methods with limited number of iterations.

In this research, t-SNE is used as the dimension reduction method, but other clustering methods, such as principal component analysis (PCA), can be potentially adopted as well for the same purpose (Chen et al., 2018). Among all the dimension reduction methods, t-SNE is able to handle the non-linear relationship between variables and hence form smaller groups compared to other methods (Fernández et al., 2013). Many studies have demonstrated the effectiveness of t-SNE in handling very high-dimensional datasets (Booth et al., 2016; Th et al., 2015). in the numerical example, we also compare the t-SNE with other PCA-based methods and demonstrate the effusiveness of t-SNE.

We set χ_i as the vector of observed traffic counts or traffic speed on each day and i denotes the index of the dates. χ_i is a one-dimensional vector with length $N \times O$, where N is the number of time intervals in a day and O is the number of observations per time interval. Then we minimize the objective function C to search for the low dimensional feature ψ_i, where i also denotes the index of dates. Then we are able to use the feature ψ_i to represent the high dimension variable χ_i for each day.

One important feature of the projected dimension by t-SNE is that it has state-of-art visualization properties of data. The low dimensional space not only retains the local structure of the data, but also reveals the global structure in the high dimensional space.

2.5.2. Clustering

Clustering methods group day-to-day traffic data into different patterns. Since t-SNE projects traffic data onto low dimensional feature space, which reflects the structure of high dimensional space. Even a simple clustering method works well on the feature space. In this research, we adopt k-means method to cluster the feature space.

We project traffic speed and traffic counts to feature space and build the clustering models, respectively. Suppose there are data available for D days, we will have U clusters for speed data and V clusters for count data after t-SNE and K-means. Then we define $U \times V$ clusters as $\{(u, v)|u \in U, v \in V\}$.

The intuition behind the clustering process is twofold: (1) Count data and speed data have different structures in the high dimensional space. Count data have larger variance than the speed data. Thus, parameter tuning for t-SNE should be different for count versus speed data. (2) Travelers' route choice is a combined decision process based on the traffic demand (count data) and traffic congestion (speed data) together. Hence we use the composite of count clusters and speed clusters to represent different patterns.

The clustering method we adopt is data-driven. Hard-coding the clusters using prior knowledge such as weekday/weekends or seasons is not necessary. Later we will show in the case study that the clustering results actually reflect not only weekday/weekend traffic patterns, but also other non-trivial factors such as incidents and events.

2.6. Route choice portions

For each traffic pattern, we compute the route choice portions for all OD pairs. Define route choice portion $p_{rs}^k(t_1)$ such that it distributes OD demand $q_{rs}(t_1)$ to path flow $f_{rs}^k(t_1)$ by Eq. (35).

$$f_{rs}^k(t_1) = p_{rs}^k(t_1) q_{rs}(t_1) \tag{35}$$

where $p_{rs}^k(t_1)$ represents the route choice portion of kth path flow in OD pair rs departing at time t_1. The time-dependent route choice portion $p_{rs}^k(t)$ can be determined through a generalized route choice model, as presented in Eq. (36).

$$(p_{rs}^{k}(t_1))_i = \Psi_{rs}^{k}(\mathscr{D}(i);i) \tag{36}$$

where $(p_{rs}^{k}(t_1))_i$ denotes the route choice portions for kth path in OD rs at time t_1 for pattern i. $\mathscr{D}(i)$ represents the traffic conditions (flow, travel time, speed, travel time reliability, etc.) of all those days within the pattern i. $\Psi_{rs}^{k}(\cdot)$ is a generalized route choice model that takes any information within the traffic pattern and compute the route choice portion for travelers in kth path in OD rs. To simplify the notation, we ignore the pattern index i in the rest of the paper.

For instance, we can use a Logit-based model based on mean travel time for each traffic pattern as shown in Eq. (37).

$$p_{rs}^{k}(t_1) = \frac{\exp(-\theta \tilde{c}_{rs}^{k}(t_1))}{\sum_{k \in K_{rs}} \exp(-\theta \tilde{c}_{rs}^{k}(t_1))} \tag{37}$$

where $\tilde{c}_{rs}^{k}$ represents the mean travel time of path flow k in OD rs departing at time t_1 for all days within the cluster (or pattern). θ is the dispersion factor in Logit model. To discretize the time, we further assume that the route choice portions stay the same in each time interval, then,

$$\bar{p}_{rs}^{kh_1} \coloneqq p_{rs}^{k}(t_1),\ \forall\ t_1 \in H_{h_1} \tag{38}$$

The discrete time link flow and path flow can be formulated as in Eq. (39).

$$\begin{aligned} \bar{f}_{rs}^{kh_1} &= \int_{t1 \in Hh_1} f_{rs}^{k}(t_1)dt_1 \\ &= \int_{t1 \in Hh_1} p_{rs}^{k}(t_1) q_{rs}(t_1)dt_1 \\ &= \bar{p}_{rs}^{kh_1} \int_{t1 \in Hh_1} q_{rs}(t_1)dt_1 \\ &= \bar{p}_{rs}^{kh_1} \bar{q}_{rs}^{h_1} \end{aligned} \tag{39}$$

2.7. Estimate the dynamic OD demand

Now we are ready to present the formulation for solving the DODE problem. Combining Eqs. (9), (29) and (39), the DODE formulation is presented in Eq. (40).

$$\begin{aligned} \min_{\{q_{rs}^{h_1}\}_{r,s,h_1}} \quad & \sum_{a \in A^o} \sum_{h_2=1}^{N} \left(\left\| \bar{x}_a^{h_2} - \sum_{rs \in K_q} \sum_{k \in K_{rs}} \sum_{h_1=1}^{N} \delta_{rs}^{ka} \rho_{rs}^{ka}(h_1, h_2) p_{rs}^{kh_1} \bar{q}_{rs}^{h_1} \right\|_2^2 \right) \\ \text{s.t.} \quad & \bar{q}_{rs}^{h_1} \geqslant 0 \qquad \forall\ rs \in K_q,\ 1 \leqslant h_1 \leqslant N \end{aligned} \tag{40}$$

In the formulation (40), link flows $\bar{x}_a^{h_2}$ are observed from sensors, path/link indices matrix δ_{rs}^{ka} is from network topology in Section 2.2, DAR matrix can be computed through real-time traffic speed data by Section 2.3 and route choice matrix p_{rs}^{kh} is determined by the clustering results in Section 2.5 and the route choice model in Section 2.6. We can formulate the multi-day 24/7 DODE problem as one large non-negative least square (NNLS) problem by viewing the $T_1 \cup T_2$ as the entire observation time period (e.g., 3 years in the case study). However, to ensure computational efficiency, a best practice is to decompose the NNLS problem of multiple years into subproblems for each of those days separately. This does not come without a price, though. The vehicles departing at the end of day 1 and arriving in the beginning day 2 are overlooked in this simplified process. This is still acceptable in practice since midnight OD is usually minimal and of less interest in general. One nice feature of solving NNLS on the daily basis is that it convenient to utilize the parallel computational power to estimate the dynamic OD of each day separately. In the reminder of this paper, the optimization Problem (40) applies for each day separately and we simply ignore the index for days.

In formulation (40), the link capacity constraints (the estimated link flow should be less and equal than the maximum flow capacity) are not explicitly enforced, since these constraints are usually satisfied by (1) achieving the minimum of the objective function close to zero; and (2) enforcing proper route choice models. As can be seen in the following case study, this is generally satisfied. In practice, if it is not the case, enforcing the link flow capacity as additional linear constraints to formulation (40) is straightforward under an iterative balancing framework (Zhang et al., 2008).

We denote B as the assignment matrix, the entries of B can be computed as in Eq. (41).

$$\mathrm{B}_{rs}^{ka}(h_1, h_2) = \delta_{rs}^{ka} \rho_{rs}^{ka}(h_1, h_2) \bar{p}_{rs}^{kh_1} \tag{41}$$

Formulation (40) is a non-negative least square (NNLS) problem in terms of x^{h_2} and B, which can be solved very efficiently in a low dimensional space (Lawson and Hanson, 1995) using the standard NNLS solver. But the standard method can be very inefficient in a high dimensional space, as it computes the inverse of $\mathrm{B}^T\mathrm{B}$ during the solving process. The dimension of $\mathrm{B}^T\mathrm{B}$ is usually in billions for a typical DODE problem that estimates daily dynamic OD. In the following section, we will propose a stochastic projected gradient descent method to solve the high-dimensional NNLS problem and implement it on GPU. The DODE problem on a single day can be solved in seconds using this proposed method.

Table 2
DODE framework variable vectorization.

Variable	Notations	Dimension	Type	Description
OD flow	q_{rs}^{h}	$\mathbb{R}^{N\|K\|}$	Dense	kth OD flow in time interval h is place at entry $(h-1)\|K\| + k$
Path flow	f_{rs}^{kh}	$\mathbb{R}^{N\Pi}$	Dense	kth path flow in time interval h is placed at entry $(h-1)\Pi + k$
Link flow	x_{a}^{h}	$\mathbb{R}^{N\|A\|}$	Dense	kth link flow in time interval h is placed at entry $(N-1)\|A\| + k$
DAR matrix	$\rho_{rs}^{ka}(h_1, h_2)$	$\mathbb{R}^{N\|A\|\times N\Pi}$	Sparse	Dynamic assignment ratio of kth path in OD rs in time interval h_1 for link a in time interval h_2 is placed at entry $[(h_2-1)\|A\| + a, (h_1-1)\Pi + k]$
Link/path indices matrix	δ_{rs}^{ka}	$\mathbb{R}^{\|A\|\times\Pi}$	Sparse	δ_{rs}^{ka} is 1 if path k for OD pair rs passes link a
Route choice matrix	p_{rs}^{kh}	$\mathbb{R}^{N\Pi\times N\|K\|}$	Sparse	Route choice for path k for OD pair rs in time interval h is placed at entry $[(h-1)\|\Pi\| + k, (h-1)\|K\| + rs]$

3. Solution algorithm

In previous section, we formulate the 24/7 DODE problem as a non-negative least square (NNLS) problem, as presented in Eq. (42).

$$\begin{aligned} \min_{\bar{q}} \quad & \|\bar{x} - \mathrm{B}\bar{q}\|_2^2 \\ \text{s.t.} \quad & \bar{q}_{rs}^{h_1} \geqslant 0 \quad \forall\, rs \in K_q,\ 1 \leqslant h_1 \leqslant N \end{aligned} \tag{42}$$

where $\bar{x}$ and $\bar{q}$ are the tensor representations of link flows and the OD flows in all time intervals, respectively. B is the assignment matrix. The construction of the tensor representations will be presented in the following section.

With the increasing granularity of traffic data, the dimensions of tensor x, q and matrix B grow quickly. Thus, we have to work on a high dimensional space for the proposed DODE framework. In this section, we discuss the technical details of each component of the solution algorithm that ensures computationally efficient implementation of the proposed framework.

3.1. Tensor representation

To enable tensor manipulation and computation during the DODE framework, all the variables involved need to be vectorized. For sparse matrices in the formulation, we use coordinate format sparse representation of the matrices.

For N intervals, denote total number path is $\Pi = \sum_{rs} |K_{rs}|$, $K = |K_q|$. The vectorized variables are presented in Table 2. Multiplications between sparse matrix and sparse matrix, sparse matrix and dense vector are very efficient, especially on multi-core CPUs or Graphics Processing Units (GPU).

3.2. Constructing the dynamic assignment ratio (DAR) matrix

The assignment matrix B is the multiplication of Link/path indices matrix, DAR matrix and route choice matrix. As shown in Table 2, the largest matrix among the three matrices is the dynamic assignment ratio (DAR) matrix. DAR matrix is constructed by network topology and speed data, and the construction process turns out to be the most time-consuming part in the DODE framework.

The construction process for DAR matrix requires iterations over all departure/arriving time intervals, paths and links. We find a way to construct DAR matrix by only iterating over departure time intervals and paths. The links and arriving time intervals will be iterated implicitly when we compute the travel time of each path. For specific time interval and path, we iterate over all the links in the path from origin to destination and compute the arrival time of each link. Using the arrival time, we can compute assignment ratio and put it to its corresponding entry in DAR matrix.

We can also use multi-process computing to construct DAR matrix for multiple days simultaneously. The parallel construction framework can significantly reduce the total computation time.

3.3. Non-negative least square on GPU

After constructing assignment matrix B, the 24/7 DODE problem is simplified to a non-negative least square problem presented in Eq. (42). However, solving such NNLS problem in high-dimensional space is non-trivial. For a general network, the dimension of OD vector is usually above ten thousand, and standard NNLS solver (Lawson and Hanson, 1995) is not able to handle such a high dimensional problem.

We propose a stochastic projected gradient descent method to solve the high dimensional NNLS problem. The process of the solution method is presented in Algorithm 1.

Algorithm 1. Stochastic Projected Gradient Descent (SPGD) method for NNLS

NNLS $(\mathrm{B}, y, b, \eta, E)$;
Input : matrix B, output y, batch size b, learning rate η, number of epoch E
Output: x such that $\mathrm{B}x = y, x \geq 0$
$(n, d) = \mathrm{B}.\mathrm{shape}$;
Initialize $x \in \mathbb{R}^n$;
for *iter* $\leftarrow 1$ **to** E **do**
  permuted_sequence = `permutate`(range(n));
  chunk_list = `make_chunk`(permuted_sequence, b);
  for *chunk* $\in$ *chunk_list* **do**
    $\mathrm{B}_o = \mathrm{B}[\mathrm{chunk}, :]$;
    $g = \mathrm{B}_o^T(\mathrm{B}_o x - y)$;
    $x = $ `Adagrad`(x, g, η);
    $x = \max(x, 0)$
  end
end

In the algorithm, the batch size b, learning rate η and number of epoch E are parameters for the SPGD method. Larger batch size implies better convergence rate but larger memory consumption; learning rate is dependent on the problem scale and larger learning rate implies better convergence rate; and larger number of epoch implies the better solution for the NNLS but longer computational time. The `permutate` function permutates the sequence in random order, `make_chunk` function divides a sequence to small chunks with same size. `Adagrad` is a variant of stochastic gradient (SGD) descent method, it outperforms the SGD during the experiments. Adagrad is an adaptive step size for SGD that is often used to optimize neural networks. Details of the Adagrad method can be found in Duchi et al. (2011).

We implemented the proposed Algorithm 1 in PyTorch, all the matrices multiplication can be evaluated on GPU. As we will show in later section, the implemented method can solve NNLS with a 10 thousand dimension in seconds.

4. Estimation framerwork

In this section, we present the proposed DODE pipeline given the network topology, speed data and count data. Path set of each OD pair needs to be generated prior to the estimation framework. For small networks, path enumeration is possible. When the networks are large, we can simply enumerate K shortest paths (Yen, 1971; Eppstein, 1998) for each OD pair and then search for the solution in the prescribed path set.

Count data and speed data need to be cleaned and imputed (if missing) before the estimation framework. Network topology and OD pairs will be converted to a directed graph with weighted edges. The entire DODE framework is summarized as follows,

DODE framework	
Step 0	*Data preparation.* Build directed graph representation for networks, enumerate paths for all OD pairs. Prepare link count data and speed data, attach data points to the edges of graph.
Step 1	*Constructing DAR matrix.* Construct DAR matrix using the graph and speed data by Sections 2.3 and 3.2.
Step 2	*Traffic data clustering.* Divide the data into different traffic patterns by clustering the speed data and count data using methods presented in Section 2.5.
Step 3	*Constructing route choice matrix.* Construct the route choice matrix for each traffic pattern using methods presented in Section 2.6.
Step 4	*Constructing observed link flow.* Construct the count data for each day using the notation presented in Table 2.
Step 5	*Stochastic Projected Gradient Descent for NNLS.* Specify learning rate and batch size based on different problem size, conduct Stochastic Projected Gradient Descent for NNLS presented in Algorithm 1 for each day.
Step 6	*Quality check.* Check the goodness of fit for the estimated dynamic OD demand and output the results.

5. Numerical experiment: a Sacramento Regional Network

In this section, we conduct a case study on I-5 and Hwy-99 towards Sacramento. 5-min count and speed data for the years of 2014 to 2016 are used to estimate 5-min dynamic OD demands over 3 years. Efficiency of the proposed methods and goodness of fit are evaluated. We visualize the evolution of estimated OD demand in several ways and discuss the benefits of the high-granular traffic

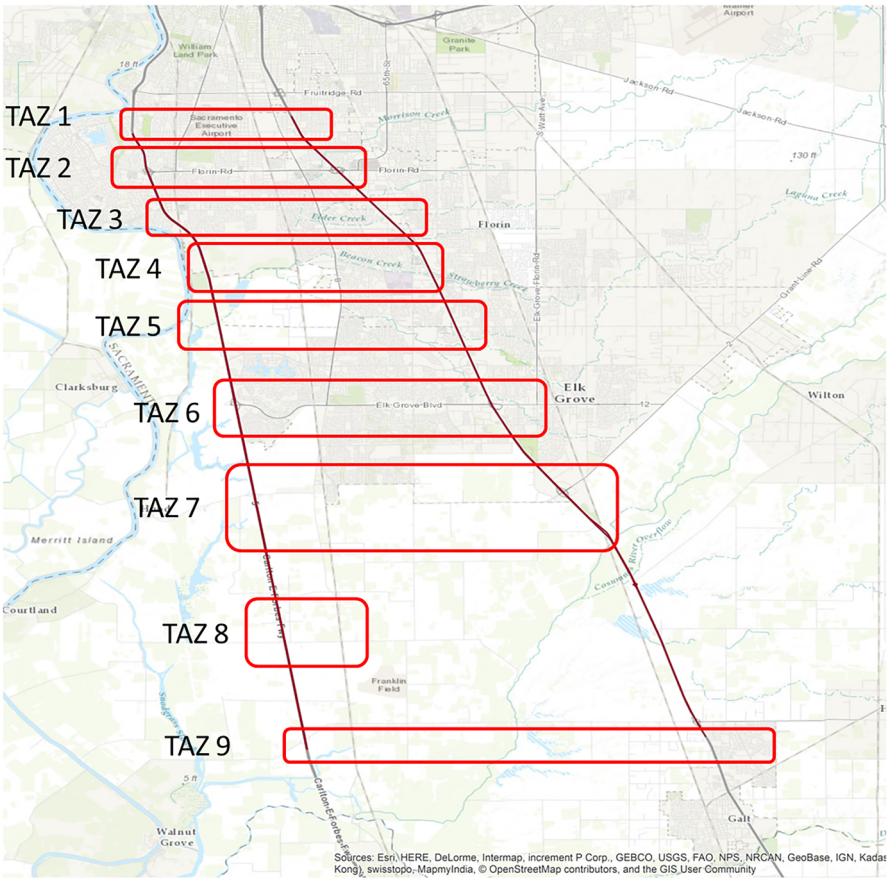


Fig. 4. Overview of network and TAZ zones.

data.

All the experiments below are conducted on a desktop with Intel Core i7-6700 K CPU @ 4.00 GHz × 8, 2133 MHz 2 × 16 GB RAM, GeForce GTX 1080 Ti/PCIe/SSE2, 500 GB SSD.

5.1. Data acquisition and preprocessing

We first describe the network, traffic count and speed data used in the case study. The data preprocessing involves the graph construction, data geocoding, data cleaning, data imputation and data interpolation.

5.1.1. Network

I-5 and SR-99 are the two highway corridors in this network. The OD connectors are constructed based on the residence region and interchanges/ramps of two highways. We divide the entire network into 9 traffic analysis zones (TAZs), and attach one origin and one destination to each TAZ. The overview of all 9 TAZs are shown in Fig. 4.

The 9 TAZs are across two major highways towards Sacramento downtown. The main purpose of this case study is to characterize the traffic demand in the southern region of Sacramento heading/leaving Sacramento downtown. Northern regions of TAZ 1 are not modeled since there are too many highway exits/entrances and local roads, our data are not rich enough to accurately model the demand profile in those regions. The north of TAZ 9 are not modeled since there is few resident area in this area. We further enumerate all paths to generate the path set for each OD pair.

5.1.2. Counts

The flow count raw data are obtained from Caltrans Performance Measurement System (PeMS), which is a combined source from various types of vehicle detector stations, including inductive loops, side-fire radar, and magnetometers. The count data contain the traffic counts from 94 locations in every 5 min for 3 years. There exist several sensors on the same road segment. In this case, we take the average of counts for that segment. On each day, there are 60min/5min × 24 h= 288 time intervals, thus the traffic count data for each day is a vector in $\mathbb{R}^{288}$. We randomly select 6 locations and visualize the day-to-day traffic counts. The average traffic counts over the 3 years for each time interval are also plotted in Fig. 5. Each grey time-of-day trace represents traffic counts over one day, and the blue line represents the average daily time-of-day traffic counts over three years.

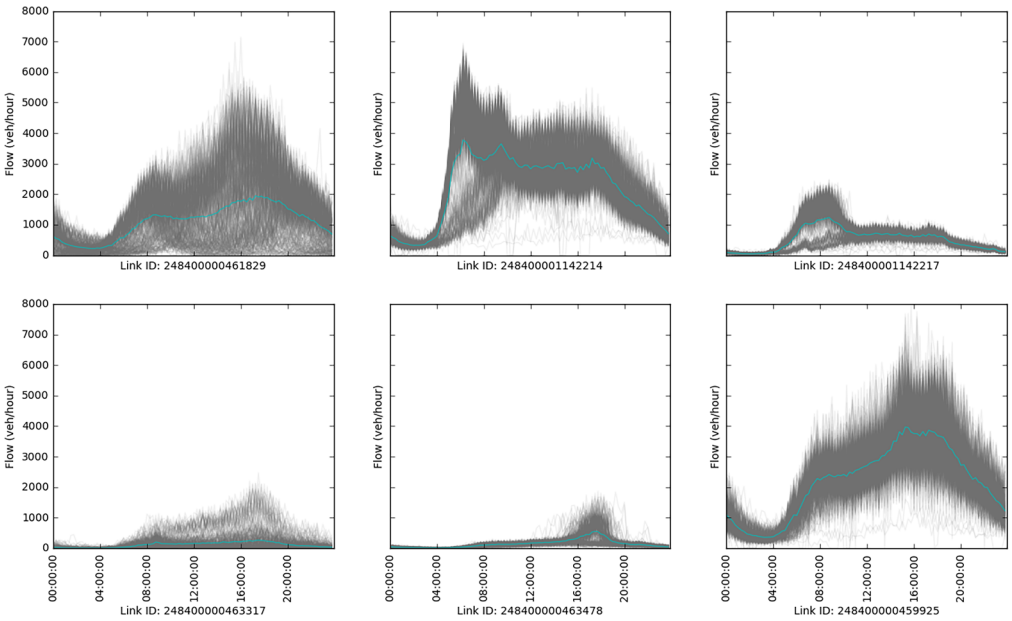


Fig. 5. Traffic counts for randomly selected 6 sensors.

As can be seen from Fig. 5, traffic counts data on most of days follow similar trends but contain large day-to-day variation. Some sensors pick up morning peaks and afternoon peaks, while others can only capture either or neither of the traffic peaks.

5.1.3. Speeds

Traffic speed data were obtained from National Performance Management Research Data Set (NPMRDS). The traffic speed data are provided at the geographic level of Traffic Message Channel (TMC), one of the geo-reference protocols. NPMRDS data contain traffic speed observations for 43 TMCs in every 5 min from 2014 to 2016. On each day, there are 288 time intervals, and thus the traffic speed data for each day is a vector in $\mathbb{R}^{288}$. We geocode the TMCs to the network and compute the time-dependent travel time for each road segment. There exist several TMCs attached to the same road segment, we take the average of the traffic speed over those TMCs for that road segment. We visualize the day-to-day traffic speed data for 16 randomly selected TMCs, as well as the mean time-of-day speed, plot in Fig. 6. Each grey time-of-day trace represents traffic speed over one day, and the blue line represents the

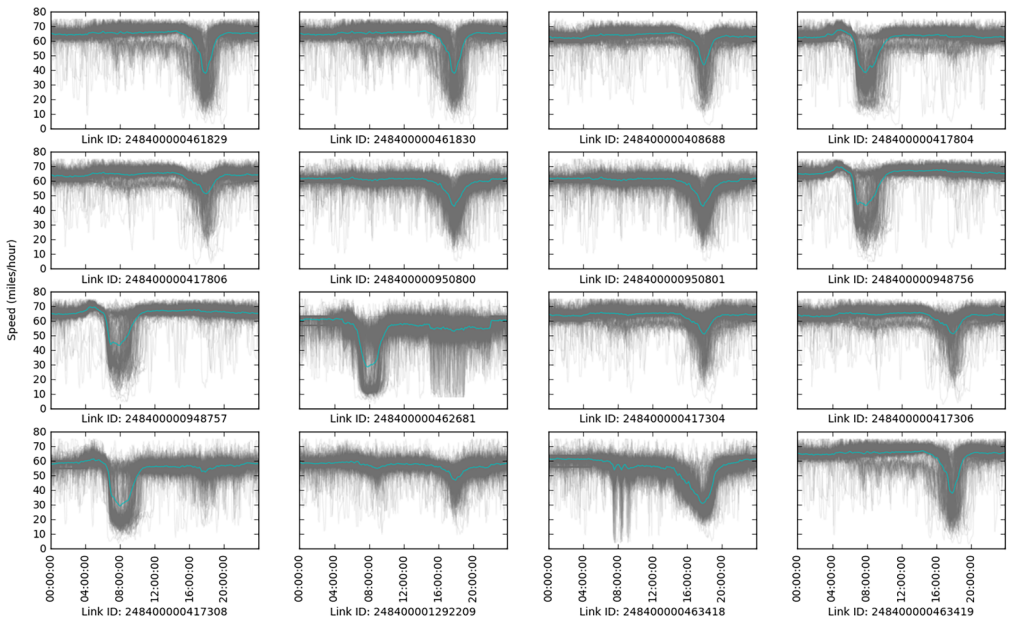
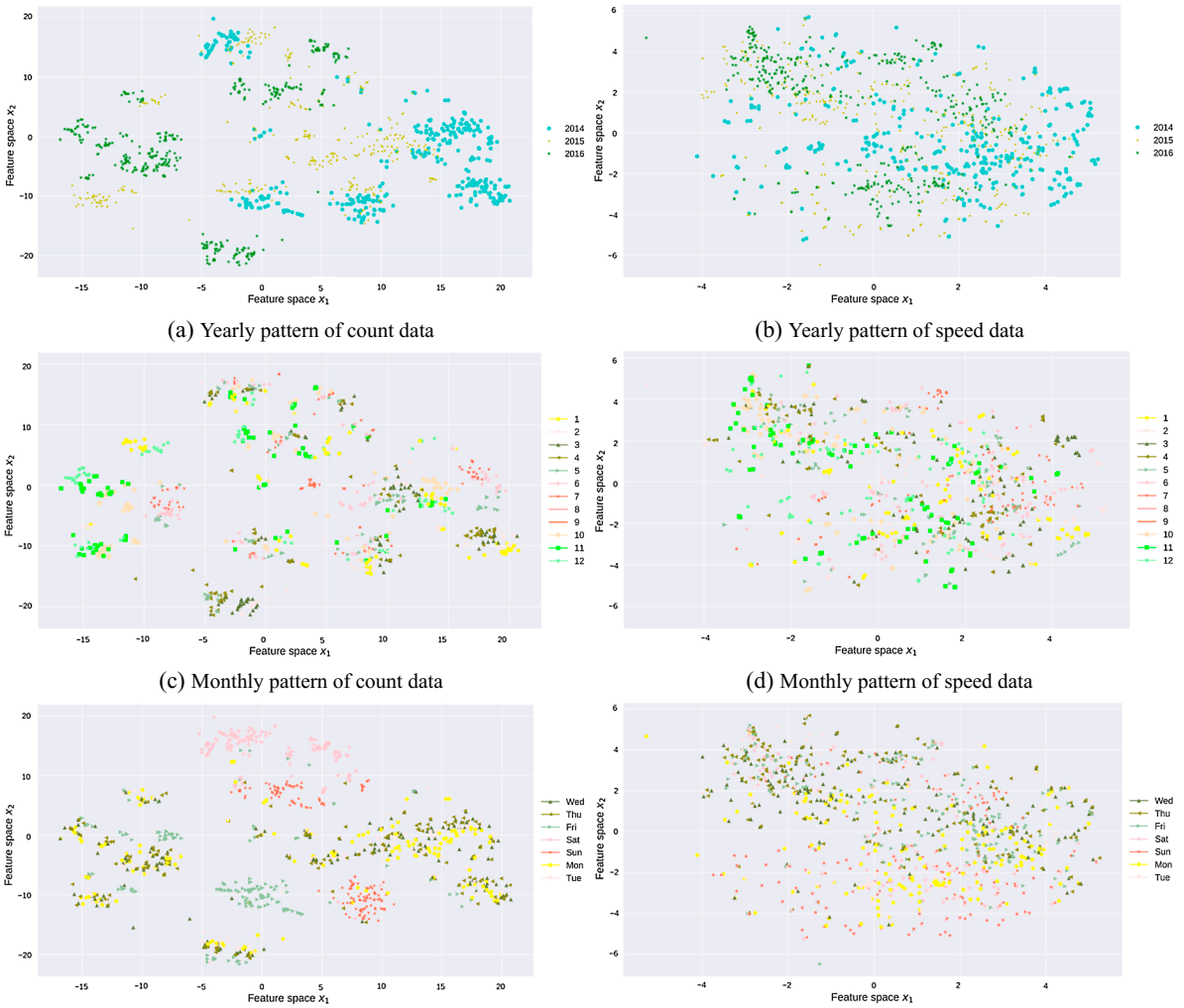


Fig. 6. Traffic speed for randomly selected 16 sensors.



(a) Yearly pattern of count data

(b) Yearly pattern of speed data

(c) Monthly pattern of count data

(d) Monthly pattern of speed data

(e) Weekday pattern of count data

(f) Weekday pattern of speed data

Fig. 7. Patterns on t-SNE feature space for count and speed data.

average traffic speed over three years. Similar pattern as in Fig. 5 can be observed in Fig. 6. Similar to counts data, traffic speeds show clearly patterns where speed drops during morning peaks or afternoon peaks, but day-to-day variations are quite large.

There are less than 1% data missing in the speed data. We use linear interpolation across different time intervals on one day and several neighboring days to impute data. For example, if the traffic speed at 10:00 is missing,then we take the average of traffic speed at 9:55 and 10:05 to impute the traffic speed at 10:00. If data for day 2 are missing, we take the average of traffic data for day 1 and day 3 as the imputed value. Note the former method is always preferred. Only when there are data missing in a large chunk of time intervals, the latter method will be used.

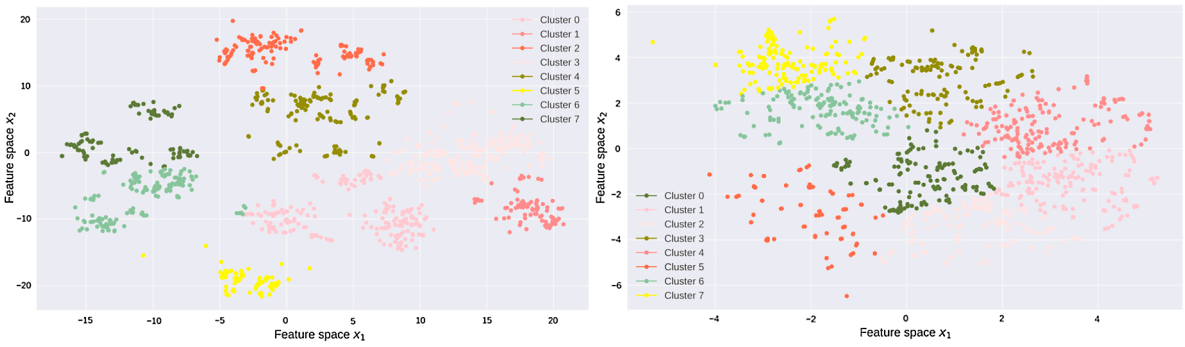
5.2. Clustering and route choice analysis

After processing the data, we use t-SNE to project the dimension of both traffic counts and traffic speed data to a lower dimensional feature space. Then a clustering method is adopted on this feature space to obtain traffic patterns.

5.2.1. Dimension reduction

We project both traffic data and speed data to a two-dimensional space so that we can visualize the data easily. TSNE package in scikit-learn is used to conduct t-SNE algorithm. The parameters for t-SNE are set as follows:

- *Count data:* perplexity 60, early exaggeration 12, learning rate 200
- *Speed data:* perplexity 20, early exaggeration 2, learning rate 80



(a) Count data (b) Speed data

Fig. 8. Clustering results for count and speed data.

The perplexity, early exaggeration and learning rate are parameters in the t-SNE algorithm. These parameters are data dependent and can be tuned through cross-validation. We visualize the count data and speed data in the feature space, respectively. Each point represents traffic data for one day, x-axis and y-axis represent the coordinates of the feature space. The absolute coordinates of each data point does not matter, while the relative positions of these data points matter. The relative positions of the data points indicate whether the data points are similar to each other and how the data points are clustered. We also colored each data point with respect to its year, month and weekday as in Fig. 7.

Feature space, like the principle component in PCA, is the base of the low-dimensional space extracted by t-SNE. As can be seen, the count data are more separable as the variance of count data is greater than the variance of speed data. The feature space reflects the yearly, monthly and daily pattern of traffic data. For example in Fig. 7a and b, traffic data in 2014 and 2016 are each grouped and far away between each other. Traffic data in 2015 lie in between groups of 2014 and 2016. In Fig. 7c, traffic flow in each month is grouped into several clusters, meaning traffic counts data has clearly monthly patterns. While in Fig. 7d, the speed data does not have very clear monthly patterns. Fig. 7e and f indicate both count data and speed data have strong weekly patterns, as Saturday/Sunday are clustered together and Wednesday/Thursday are clustered together.

We also apply the PCA, Latent Dirichlet Allocation (LDA) and kernel PCA with degree 3 polynomial kernel to the same count data and speed data, and the weekly/monthly/yearly patterns are not clear from those results. The figures similar to Fig. 7 can be found in the supplementary materials. The t-SNE tends to divide the data points into small groups, while other methods usually generate a cluttered visualization. To better cluster the data points, we use the results by t-SNE for the rest of the experiments.

5.2.2. Clustering

After dimension reduction, we use k-means to cluster the data points on the feature space. We choose the number of clusters $k = 8$ for both count and speed data, k-means method converges very quickly and the results are shown in Fig. 8.

Travelers can make different route choices based on traffic patterns related to both traffic volumes (traffic counts) or traffic congestion (traffic speed). We define $8 \times 8 = 64$ different traffic patterns to take into account characteristics of different count and speed clusters. The number of traffic data in each pattern are presented in Fig. 9. We drop all the patterns with no data point. There are in all 55 valid traffic patterns.

The outliers are also picked out during the clustering process. For example only one data point falls in the combination of count cluster 0 and speed cluster 0. This data point can be viewed as one outlier that does not share similarity with any other traffic patterns. We compute travelers' route choice portions of this outlier day using its unique traffic conditions.

For patterns with more than one data points (i.e., days), we compute the route choice portions using the average traffic speed of all days within each pattern, as discussed in Section 2.6. We adopt $\theta = 0.01$ since the magnitude of the travel time is around hundreds of seconds. In this demonstrative case study, θ is determined without careful calibration, which can be improved in the future research using methods proposed by Lu et al. (2015), Yang et al. (2001).

5.3. Dynamic OD estimation

Having the DAR matrix of each day computed by Section 2.3 and route choice portion matrix of each pattern computed by Section 2.6, we estimate the dynamic OD demand using the proposed stochastic projected gradient descent method.

5.3.1. Goodness of fit

In the stochastic gradient method, the configurations are set as follows:

- *number of epochs*: 300
- *batch size*: 8192
- *step size*: 5
- *use GPU*: True

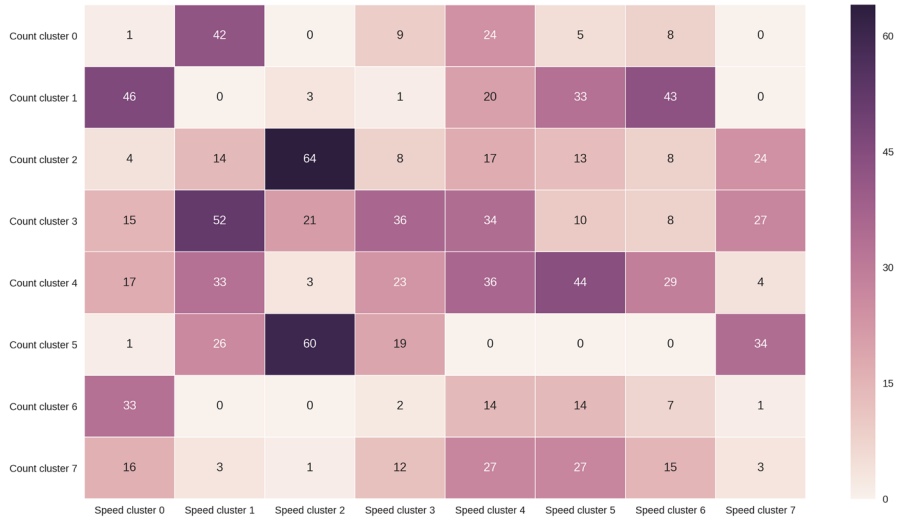


Fig. 9. Number of traffic data in each traffic pattern.

The entire estimation process for three years takes around 20 h, with an average of 1 min for each day. We randomly selected 16 days to visualize the observed traffic counts and estimated traffic counts in Fig. 10. The average R-square between the observed link flow and estimated link flow is 0.87 for three years. The estimated OD demands are able to reproduce the traffic counts observations, implying satisfactory results.

The true OD demand is difficult to obtain in real-world networks, so the comparison between the estimated OD demand and true OD demand is infeasible in the case study. To further validate the estimation results, we propose a novel interpretation of DODE formulation as follows: we view the observed link flow as the "data", the DAR matrix as the "model" and estimated OD as "target" in the DODE formulation. The terms "data", "model" and "target" are used to assimilate a typical machine/statistical learning task. Under this setting, the DODE formulation can be described as follows: given an observed "data", we train the "model" with the speed data and then compute the "target" by inputting the "data" to the "model". We first examine the stability of the "model". We compute

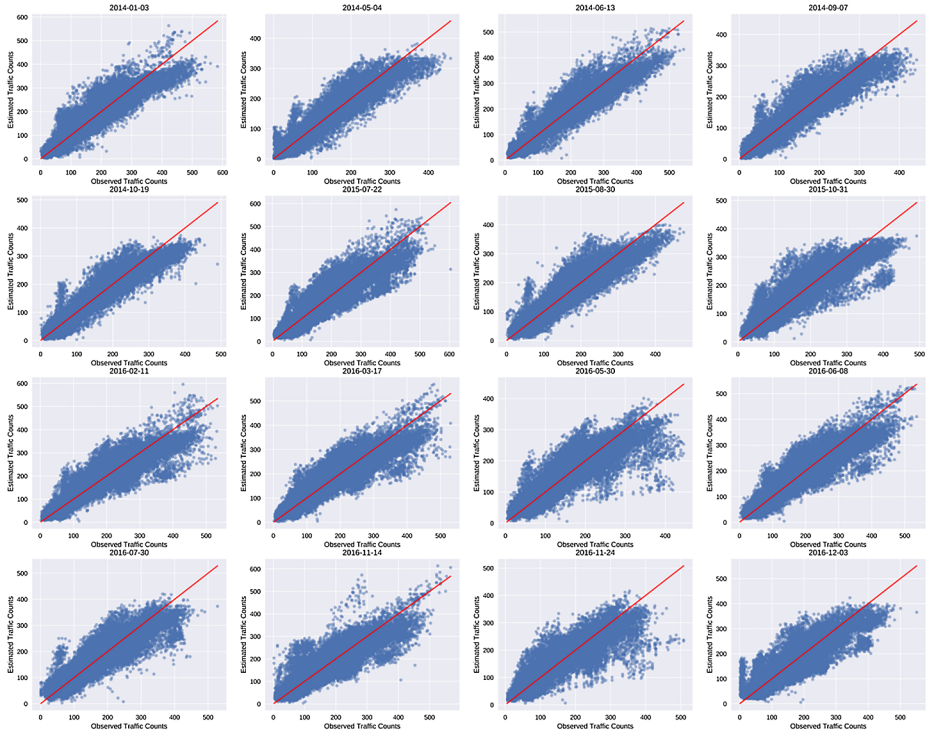


Fig. 10. Observed v.s. estimated traffic counts in 16 randomly selected days.

(a) ℓ^2 norm distance between the DAR matrix and average DAR matrix over three years

(b) R^2 between the observed link flow and estimated link flow across December, 2018

Fig. 11. Empirical test on the DAR matrix and OD estimation results.

the average DAR matrix across three years and plot the histogram of ℓ^2 distance between the DAR matrix on each day and the average DAR matrix in Fig. 11a. One can clearly see the distribution of ℓ^2 distance is unimodal, which implies the daily perturbation of traffic conditions has a bounded impact to the DAR matrix, thus the OD estimation results are robust to the observation errors and inaccurate DAR matrix. We also adopt a modified cross-validation approach as follows: we assume the DAR matrices ("model") in December 2018 are unknown and estimated by the average traffic conditions in the other 35 months. We compute the R^2 between the observed link flow and estimated link flow using the estimated DAR matrix and the true DAR matrix, respectively. The results are presented in Fig. 11b. The DODE with estimated DAR matrix (average R^2 is 0.794) slightly underperforms the DODE with true DAR matrix (average R^2 is 0.797), as expected. The estimation results are still satisfactory, indicating the robustness of the proposed DODE method.

5.3.2. *Algorithm efficiency*

We also conduct an experiment to demonstrate the computational efficiency of our proposed algorithm. To compare the CPU based SPGD method, GPU based SPGD and traditional active set based NNLS method (Lawson and Hanson, 1995), we random generate a matrix $B \in \mathbb{R}^{n\times n}$, $x \in (\mathbb{R}^+)^n$, we compute $y = Bx$ and solve NNLS(B, y) using these three methods. The number of iteration n is set from 100 to 6000. As a result, the time consumptions of the three methods are presented in Fig. 12.

The CPU based SPGD method is very slow so we have to terminate it early. As can be seen, the GPU based SPGD method is significantly the most efficient of all. The gap between standard NNLS method and GPU based gradient project method will increase rapidly as n increases.

In this case study, the dimension of B is (24768, 23328) for the Sacramento regional network. It only takes GPU based SPGD method around 1 min to solve it for each day, while the standard active set method will take more than one hour. In this case study, only the GPU based SPGD method can solve the problem of three years in an acceptable amount of time.

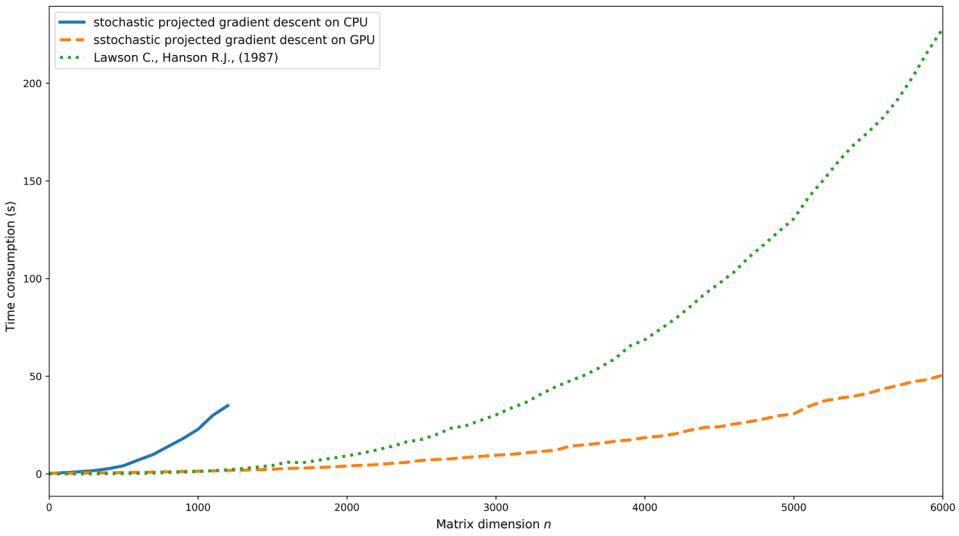
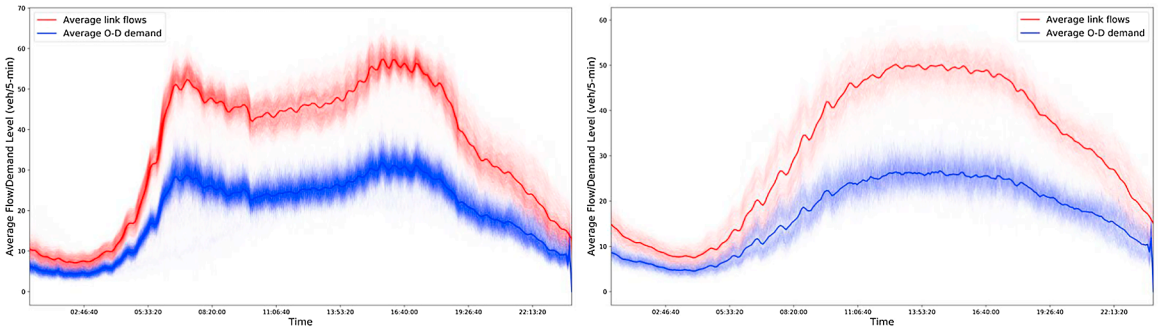


Fig. 12. Computation time of three methods with respect to matrix dimensions.



(a) Weekdays (b) Weekends

Fig. 13. Aggregated OD demand and counts by time of day, on weekdays and weekends (solid lines are the average of aggregated OD demand and counts taken over all weekdays and weekends, respectively).

5.4. Aggregated demand over all OD pairs

With the estimated 5-min dynamic OD demand over the three years, we now examine the characteristics of the traffic demand. We start with the aggregated demand over all OD pairs on each day of the three years.

5.4.1. Weekdays v.s. weekends

We first look at the differences in aggregated OD demands between weekdays and weekends. For each day, we compute the aggregated OD demand over all OD pairs at each 5-min time interval, and the aggregated traffic counts over all counting locations. Then daily average is computed over the three years. We plot time-of-day aggregated OD and counts for each day (in transparent colors), along with the daily average (in solid colors), in Fig. 13. Generally, dynamic OD demand patterns on weekdays and weekends are quite different, as expected. There are two clear spikes on weekdays corresponding to morning and afternoon peaks, respectively.

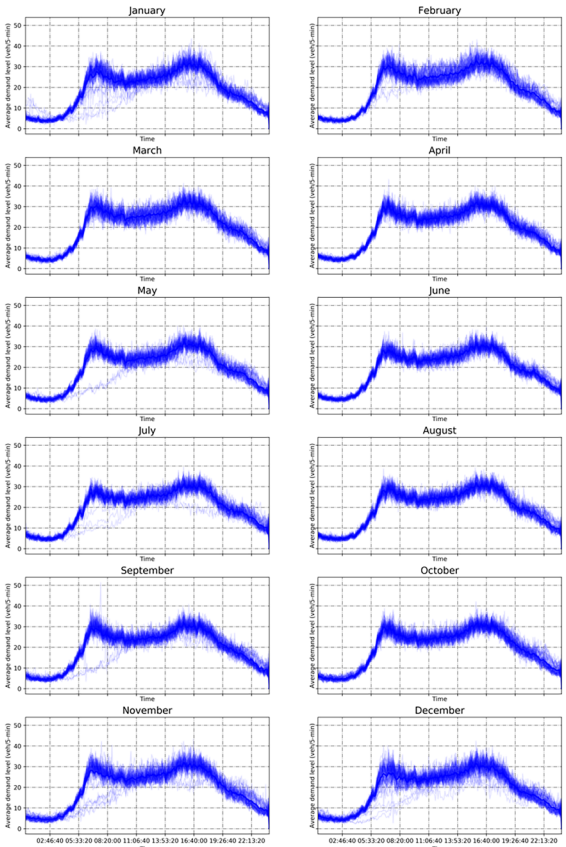


Fig. 14. Aggregated OD demand and counts, averaged over all working days in each month.

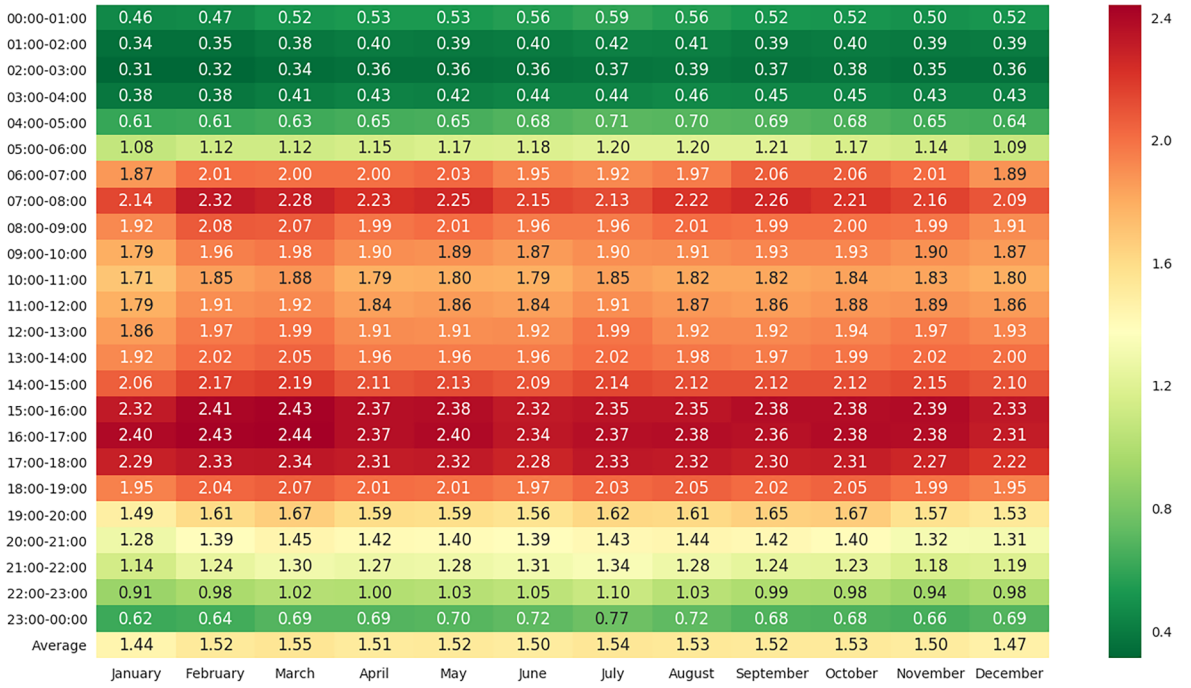


Fig. 15. Aggregated OD demand by hour, averaged over all working days in each month ($\times 10^3$ vehs).

There is only one spike on weekends, and the OD demand on weekends are fairly stable from 11:00am to 17:00 pm.

The results show that the aggregated OD demand and aggregated counts have similar time-of-day profiles, but in different scales. Total counts, as commonly used to approximate total demand level in practice, can substantially overestimate the demand level, since they tend to double count the same vehicles that pass through several counting locations. Though both generally follow similar time-of-day profiles, OD demand seems to have spikes and declines slightly earlier than what the total counts read. This indicates that

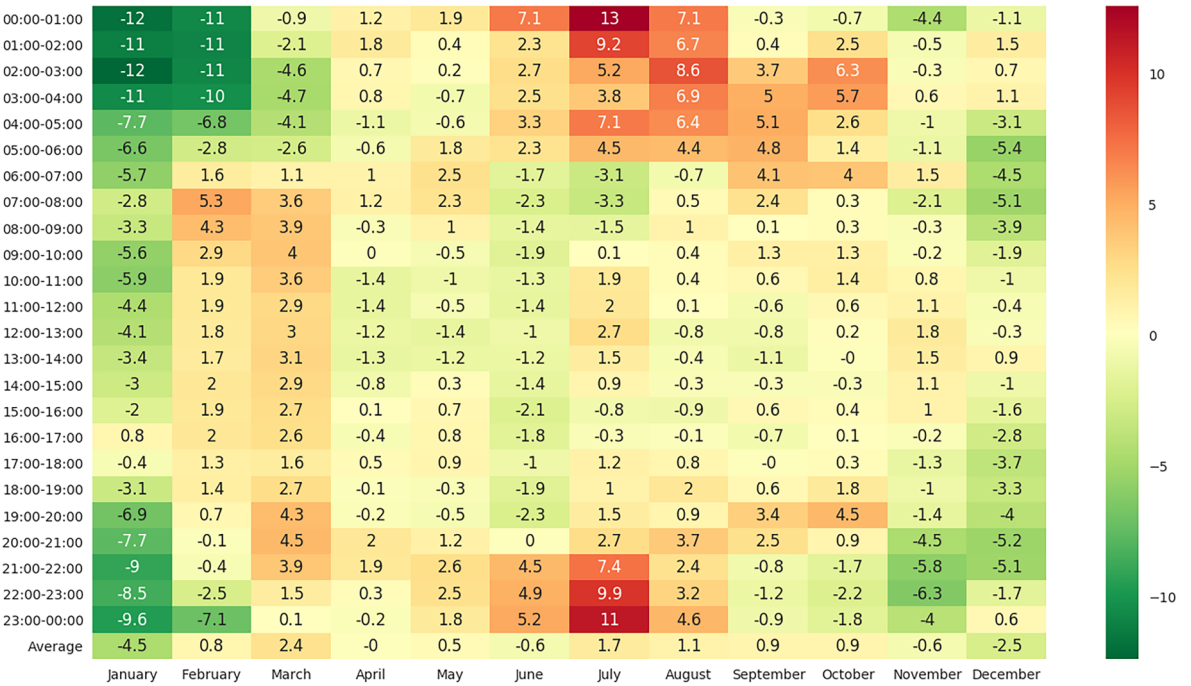


Fig. 16. Percentage change in aggregated OD demand by hour by month, comparing to the daily average of aggregated demand taken over all working days of all months (%).

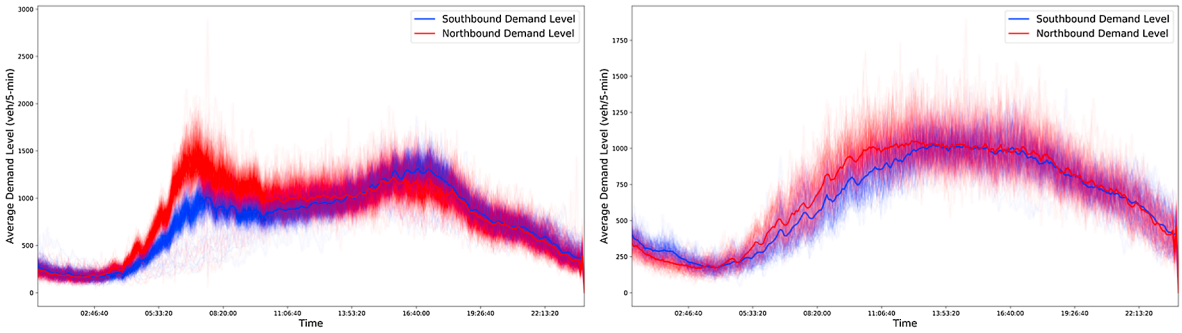


Fig. 17. Aggregated OD demand, by northbound and southbound.

spillover of congestion queues is not too long on both highway corridors, possibly only locally or in the vicinity of a bottleneck.

5.4.2. Monthly and seasonal effects on OD demand

For all working days (excluding any holidays on weekdays) in each month, we plot the daily aggregated OD demand over all OD pairs, total counts over all locations, along with their respective daily average for each month, in Fig. 14. The general time-of-day profiles are similar across different months. However, the day-to-day variation of OD demand in November, December and January are greater than other months, which may be largely attributed to the travel demands affected by holiday or winter seasons. We also compute the aggregated OD demand by hour, averaged over all working days in each month, in Fig. 15, as well as the percentage change in aggregated OD demand by hour in Fig. 16 where the base is set as the average of aggregated OD demand taken over all months.

OD demands during the morning peaks in June–August and December–January are slightly lower than other months, resulting less congestion during morning peaks. Among those, morning peak demand in July drops the most considerably compared to other months. On the other hand, summer time (from May to September) shows higher demand during off-peak hours, especially July and August. Overall, the total travel demand in December and January are the lowest throughout the years. Those monthly and seasonal demand change may be related to the summer/winter breaks of schools, and effects of summer/winter weather. These phenomena are consistent with our perception, and can be demonstrated and validated by three years' data, which cannot be discovered by examining speed/counts data directly.

5.4.3. Northbound v.s. southbound

We plot the aggregated OD demand by weekdays and weekends, and over all northbound and southbound OD pairs, respectively, in Fig. 14.

Northbound demand heads to the Sacramento downtown, and southbound demand heads to the southern region. On weekdays, the northbound OD demand is greater than southbound OD demand during morning peaks, and slightly less during afternoon peaks. Morning commute clearly shows more day-to-day variation than other time periods. One interesting observation is that the discrepancy between northbound/southbound OD demand in afternoon peaks is less than that in morning peaks. Congestion during the

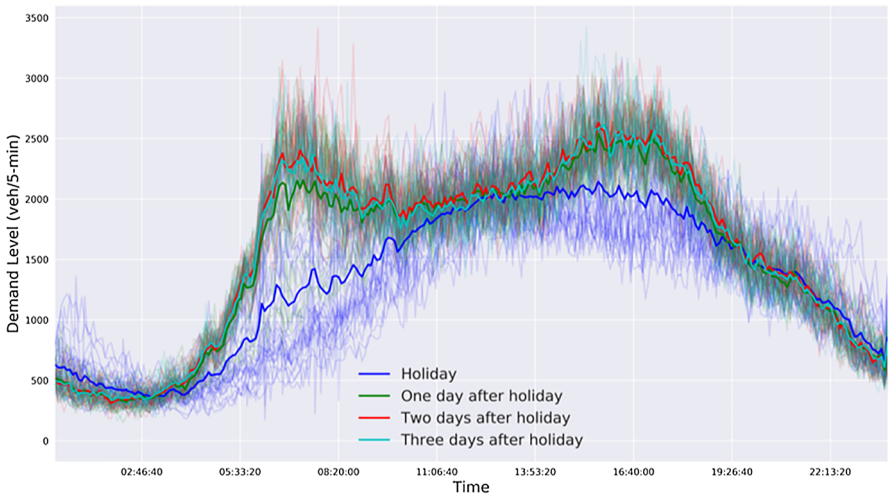


Fig. 18. Aggregated OD demand, on holidays and on weekdays immediately after holidays.

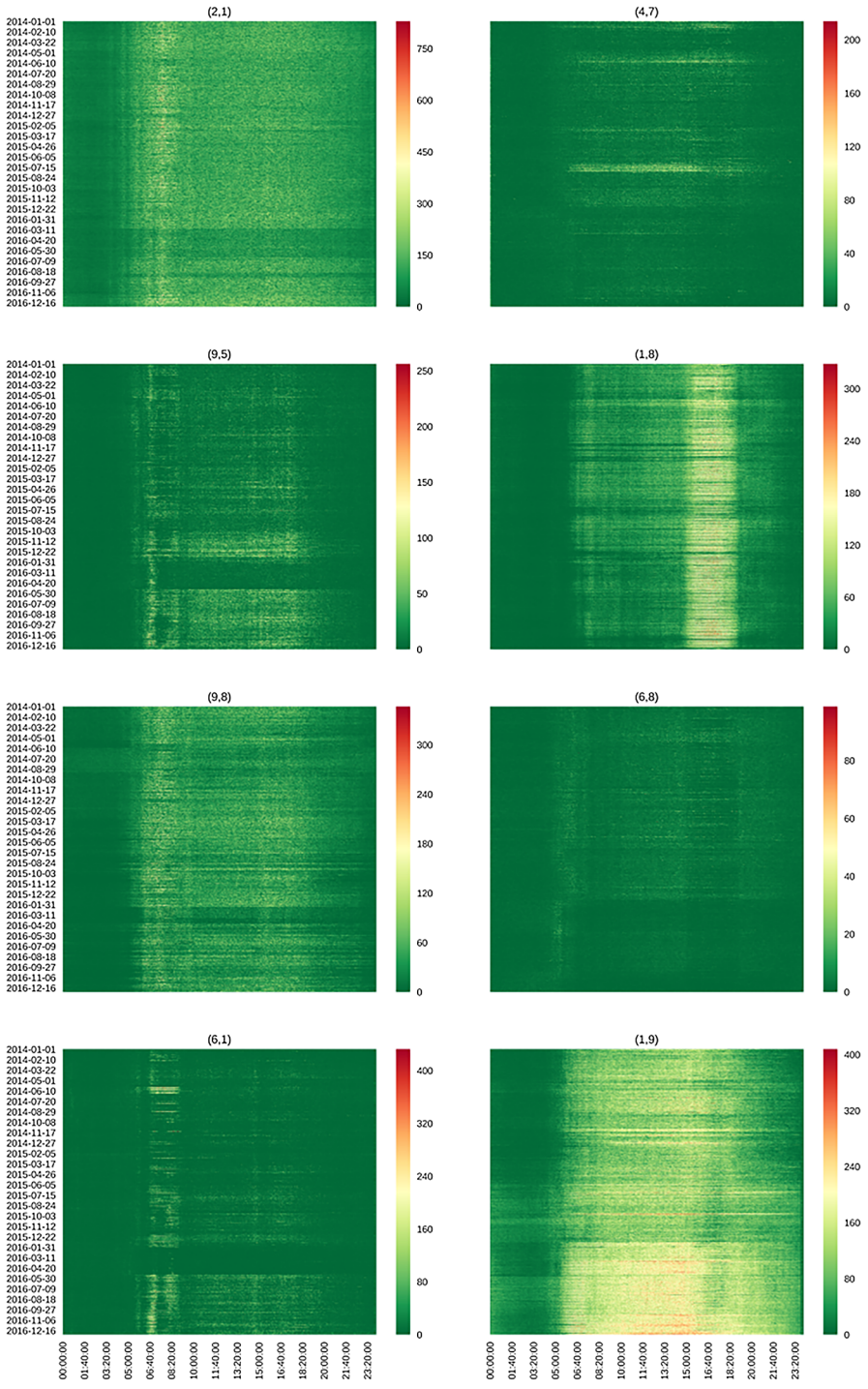


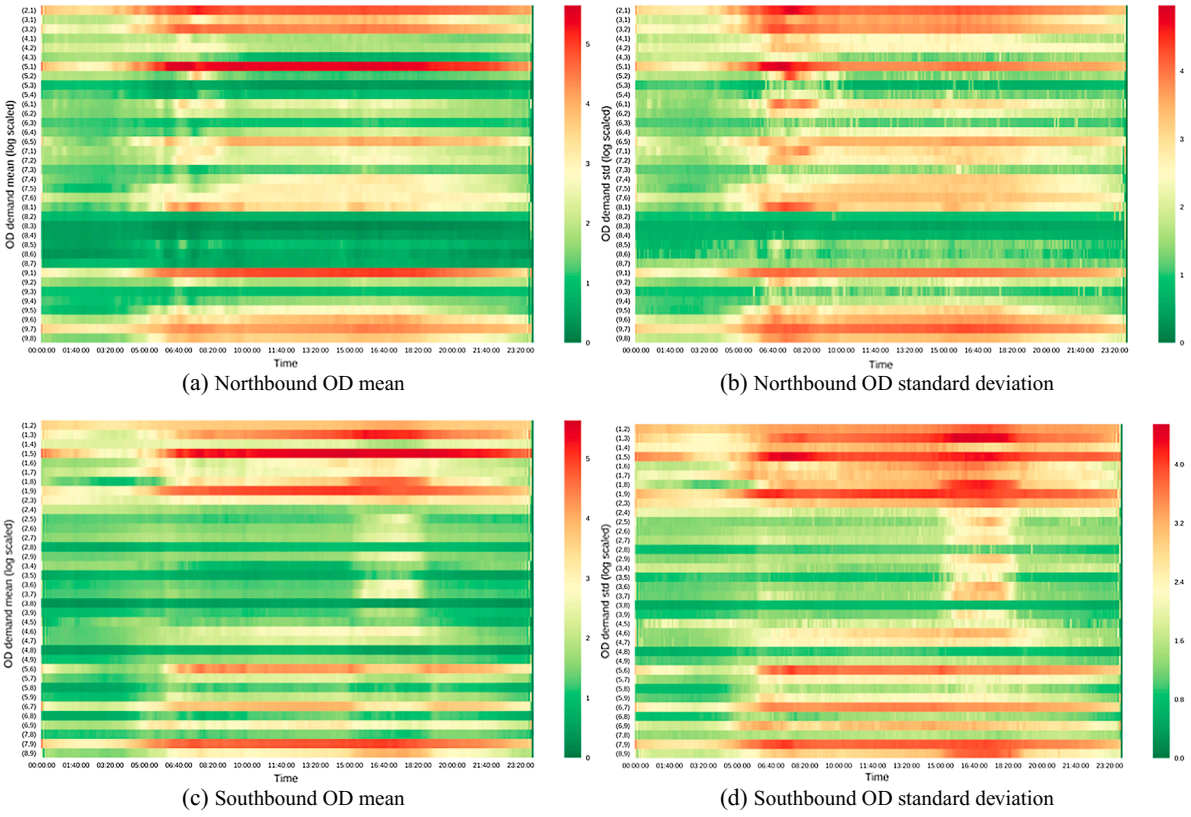
Fig. 19. Time-of-day OD demand profile for randomly selected northbound/southbound OD pairs.

day is usually more widely spread than morning commute congestion that mainly applies to northbound only (see Fig. 17).

On weekends, the OD demand per hour is considerably less than the demand rate during morning commute on weekdays. Northbound sees a higher demand level and earlier weekend peak than southbound. However, during midnight, more demand travels on southbound than northbound, possibly as a result of midnight activities in Sacramento Downtown.

5.4.4. Holidays v.s. weekdays immediately after holidays

OD demand during holidays appears quite different comparing to the regular weekdays and weekends. Thus, we pick out all the



(a) Northbound OD mean (b) Northbound OD standard deviation

(c) Southbound OD mean (d) Southbound OD standard deviation

Fig. 20. Mean and variation of OD demand, by OD pair and time of day.

holidays (excluding the weekends), and those working days immediately after holidays to visualize their respective demand patterns. For example, September 5 2016 is a Labor day on Monday, then September 6 2016 is one weekday immediately after the holiday. We compute the aggregated OD demand for the two types, and present the results in Fig. 18.

As can be seen from Fig. 18, holiday traffic patterns are closer to the weekend patterns then to the weekday patterns, with one big spike during the day. However, a small morning peak can exist for some holidays, possibly attributed to different nature of daytime activities from a regular weekend. Another interesting finding for the holiday OD demand pattern is that the midnight OD demand can be as high as 1250, almost half of the aggregated demand during morning peaks.

Though a morning commute peak resumes after holidays, we see that the peak on the weekday immediately after holidays is considerably lower than that of a regular weekday. OD demand patterns become normal from the second weekday after the holidays.

5.5. Disaggregated demand

Now we examine 24/7 OD demand of each OD pair over the 3 years.

5.5.1. Northbound v.s. southbound

We draw a figure with $(n \times m)$ pixels, n is the number of days and m is the number of time intervals on each day. We set y axis to be the dates from 2014 to 2016, and x axis to be the time of day from 00: 00 to 23: 59. Each pixel is color coded to indicate the OD demand level. This figure demonstrates the daily time-of-day demand change over the years for each OD pair in high granularity. We randomly selected 4 northbound and 4 southbound OD pairs, and plot them in Fig. 19. OD demand between the zone (1, 9) has increased substantially especially during the year of 2016, resulting an increased demand level throughout the entire 24 h. Also for OD pair (6, 1), there are clearly 3 spikes during morning commute, and demand for morning commute increases considerably in 2016. However, other OD pairs plot in Fig. 19 do not necessarily witness demand increase over time.

One can clearly see that there exist some strips with green color, implying temporary effects on travel demand for some OD pairs. For instance, OD demand is significantly reduced during January–April 2016 between the OD pair (6, 1), (9, 5). This could be possibly induced by construction projects in the regional networks that have more impacts on those OD pairs than others.

5.5.2. Mean and variance of dynamic OD demand

We compute the average and standard deviation of each OD pair for each 5-min time interval over 3 years, and plot them on a heatmap in Fig. 20. We set y-axis to be each OD pair, x-axis to be the time from 00: 00 to 23: 59. Each pixel is color coded to indicate

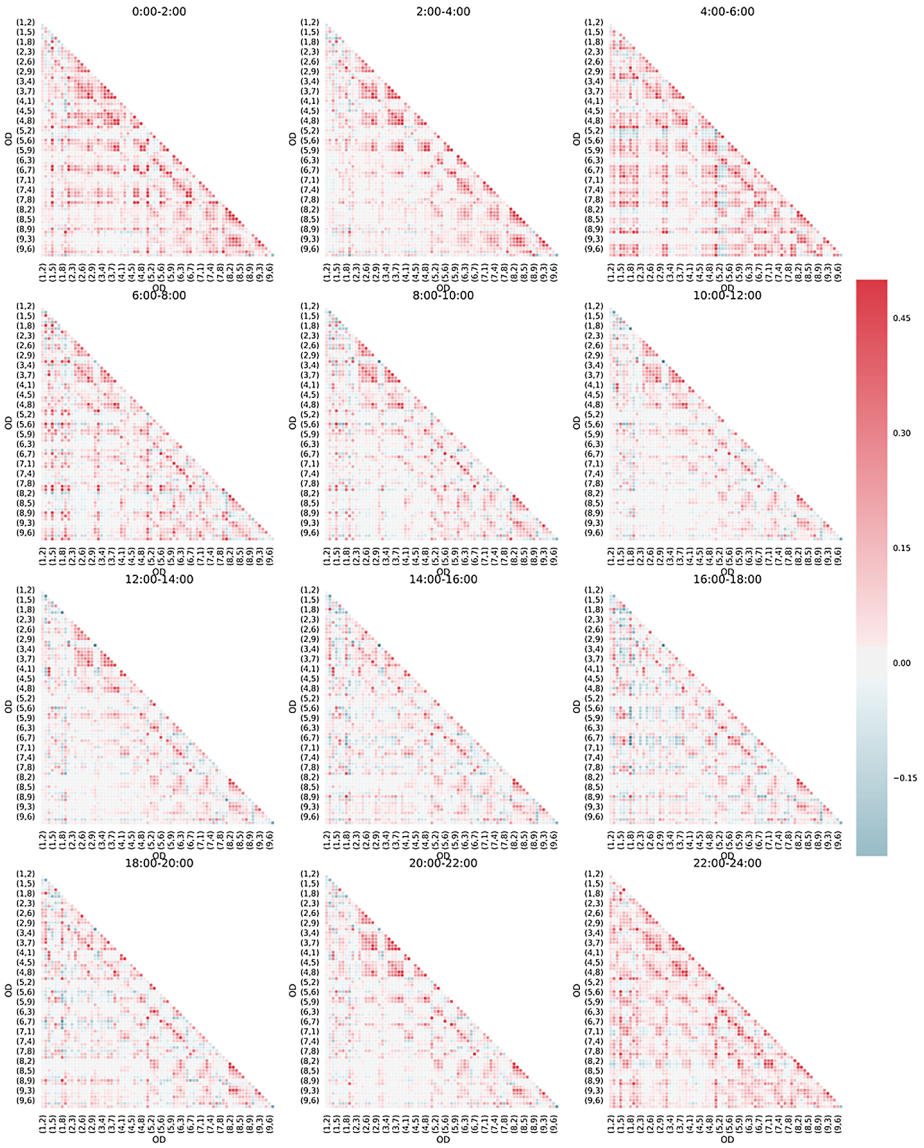


Fig. 21. OD demand correlation for different time intervals.

the OD demand level.

As can be seen from Fig. 20, the mean and variance of each OD pair roughly follow similar patterns, and the variance increases with respect to the increase in mean. Origin zones 1, 5, 6, 7 are the most important origins generating demand for southbound direction. Similarly, origin zones 2, 5, 8, 9 are the important demand origins for northbound direction.

In addition, there exist several OD pairs, such as (4, 1), (1, 6), with low demand mean and relatively high flow variability. The high variability of the demand among these OD pairs may be caused by accidents or events, so in a way, they may be more vulnerable under non-recurrent traffic conditions.

The correlation between OD pairs is useful when making the transportation planning policies. We compute the Pearson correlation factor between all OD pairs by time of day, and present the results in Fig. 21. The demand among majority of OD pairs is positively correlated. Only a small portion of OD pairs are negatively correlated, which may be worth further investigating the reasons. Generally correlations are higher during peak hours and midnight than those from 10:00 to 16:00.

5.5.3. Holidays v.s. weekdays immediately after holidays

We visualize the day-to-day mean and variance of OD demand for each OD pair on holidays and two weekdays immediately after holidays in Fig. 22. The results are consistent with before, generally demand variance increases with respect to the mean for each OD pair. There is no significant morning or afternoon peak hours for holiday travel demand. Though the total OD demand level on holidays is lower than weekdays, the holiday demand variance is much higher. The first weekday after holidays and the second

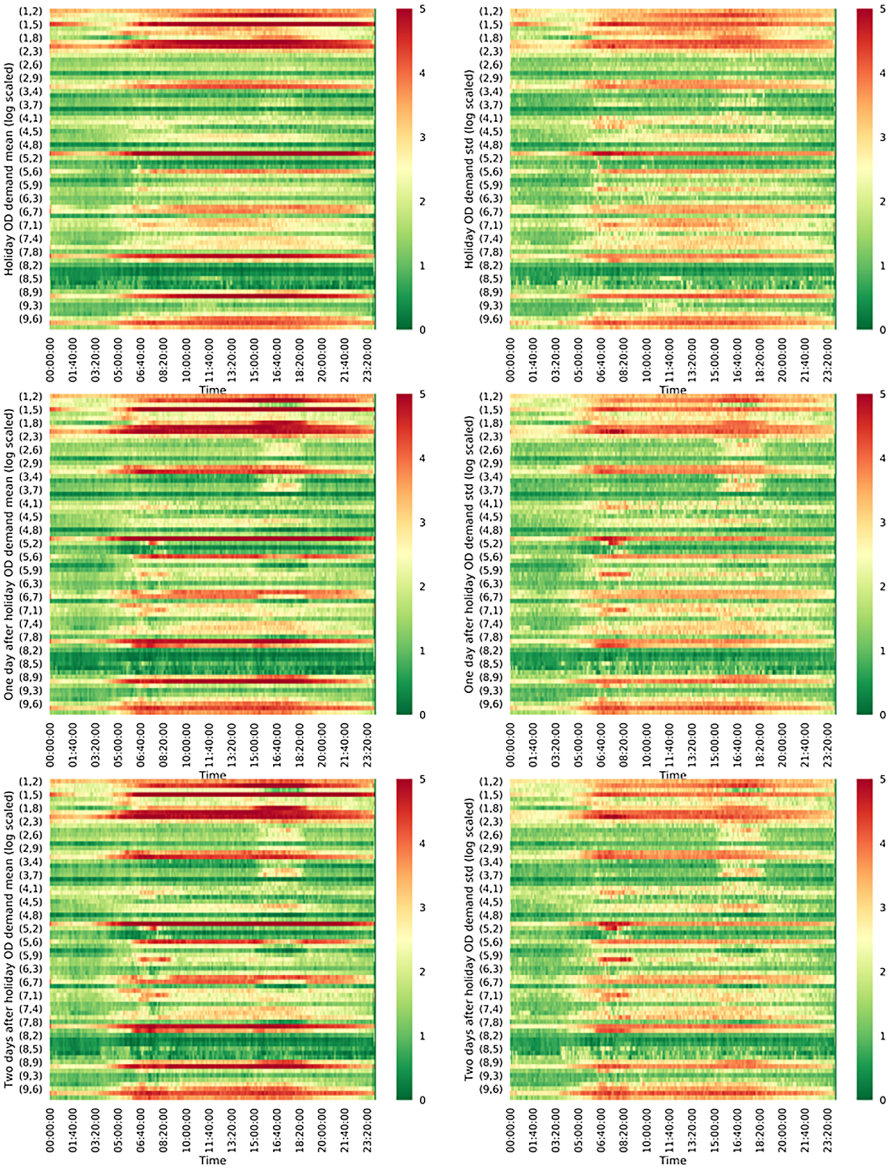


Fig. 22. Day-to-day OD demand mean and variance on holidays and weekdays immediately after holidays (left: mean; right: standard deviation; the first row: holidays; the second row: the first weekday after holidays; the third row: the second weekday after holidays).

weekday after holidays follow a similar pattern, while the latter demand is overall higher than the former demand. This again validates our finding for the aggregated OD demand.

6. Conclusion

This paper proposes a data-driven framework for estimating multi-year 24/7 dynamic OD demand using high-granular traffic counts and speed data. The proposed framework defines a dynamic assignment ratio (DAR) matrix to encapsulate the traffic flow dynamics and congestion spill-over in the large-scale network. The DAR matrix can be calibrated through high-granular speed data (such as probe vehicle speeds), which alleviates the complexity of non-linear large-scale network simulation for DODE.

The purposed framework adopts t-SNE and k-means methods to reduce the dimensionality of multi-source high-granular data, and cluster those data into typical daily traffic patterns. The t-SNE method projects the multi-source data onto a low dimensional feature space that enables examination of the daily, weekly and monthly patterns of traffic data. The k-means method clusters the projected counts and speed data into traffic patterns. The framework works with any general route choice models that considers day-to-day and within-day travel time and cost. In particular, a Logit-based route choice model is demonstrated to compute the route choice portions under each traffic patterns separately.

The DODE framework can be cast into a standard non-negative least square (NNLS) problem with, however, very high dimensions provided with high-granular data. A novel stochastic projected gradient descent (SPGD) method is purposed to solve for NNLS. The SPGD method can be implemented on GPU, which is able to solve the high dimensional NNLS efficiently compared to the traditional active set method for the NNLS problem. The entire solution framework is implemented in Python and open sourced.

Finally, a case study is conducted on a regional Sacramento network consisting with I-5 and SR-99 corridors, interchanges and ramps. High-granular counts and speed data are used to estimate 5-min dynamic OD demands over the three years from 2014 to 2016. The estimation takes around 20 h on an inexpensive GPU-based desktop. The estimated dynamic OD demand can fit the large-scale high-granular data fairly well. We also examine daily, monthly, seasonal and yearly changes in OD demand that vary by time of day, by holidays, weekdays and weekends. The new information regarding travel demand can help city planners and policymakers better understand the characteristics of dynamic OD demands and their evolution/trends in the past few years. The estimated dynamic OD can also be used to compute the variability of day-to-day OD demand, a critical input for network reliability studies (Li et al., 2018).

Acknowledgements

This research is funded in part by National Science Foundation Award CMMI-1751448 and Carnegie Mellon University's Mobility21, a National University Transportation Center for Mobility sponsored by the US Department of Transportation. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. The U.S. Government assumes no liability for the contents or use thereof. We would also like to thank anonymous reviewers for their valuable suggestions.

Appendix A. Supplementary materials

The proposed framework is implemented in Python and open-sourced on Github (https://github.com/Lemma1/DPFE). The Github repository also contains the dimension reduction results by PCA, Latent Dirichlet Allocation (LDA) and kernel PCA with degree 3 polynomial kernel.

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. Transport. Res. C: Emerg. Technol. 59, 129–146.
Antoniou, C., Ben-Akiva, M., Koutsopoulos, H., 2004. Incorporating automated vehicle identification data into origin-destination estimation. Transport. Res. Record: J. Transport. Res. Board (1882), 37–44.
Antoniou, C., Ben-Akiva, M., Koutsopoulos, H.N., 2006. Dynamic traffic demand prediction using conventional and emerging data sources. In: IEE Proce.-Intell. Transp. Syst., vol. 153. IET, pp. 97–104.
Ashok, K., Ben-Akiva, M.E., 2000. Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows. Transport. Sci. 34 (1), 21–36.
Balakrishna, R., Ben-Akiva, M., Koutsopoulos, H., 2008. Time-dependent origin-destination estimation without assignment matrices. In: Second International Symposium of Transport Simulation (ISTS06). Lausanne, Switzerland. 4–6 September 2006. EPFL Press.
Barceló, J., Montero, L., Marqués, L., Carmona, C., 2010. Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. Transport. Res. Record: J. Transport. Res. Board (2175), 19–27.
Ben-Akiva, M.E., Gao, S., Wei, Z., Wen, Y., 2012. A dynamic traffic assignment model for highly congested urban networks. Transport. Res. C: Emerg. Technol. 24, 62–82.
Bierlaire, M., Crittin, F., 2004. An efficient algorithm for real-time estimation and prediction of dynamic od tables. Oper. Res. 52 (1), 116–127.
Booth, J., Roussos, A., Zafeiriou, S., Ponniah, A., Dunaway, D., 2016. A 3d morphable model learnt from 10,000 faces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 5543–5552.
Calabrese, F., Di Lorenzo, G., Liu, L., Ratti, C., 2011. Estimating origin-destination flows using mobile phone location data. IEEE Pervasive Computing 10 (4), 0036–44.
Cascetta, E., Inaudi, D., Marquis, G., 1993. Dynamic estimators of origin-destination matrices using traffic counts. Transport. Sci. 27 (4), 363–373.
Chen, X., He, Z., Wang, J., 2018. Spatial-temporal traffic speed patterns discovery and incomplete data recovery via SVD-combined tensor decomposition. Transport. Res. C: Emerg. Technol. 86, 59–77.
Cipriani, E., Florian, M., Mahut, M., Nigro, M., 2011. A gradient approximation approach for adjusting temporal origin-destination matrices. Transport. Res. C: Emerg. Technol. 19 (2), 270–282.
Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12 (Jul), 2121–2159.
Eppstein, D., 1998. Finding the k shortest paths. SIAM J. Comput. 28 (2), 652–673.
Fisk, C., 1989. Trip matrix estimation from link traffic counts: the congested network case. Transport. Res. B: Methodol. 23 (5), 331–336.
Florian, M., Chen, Y., 1995. A coordinate descent method for the bi-level o-d matrix adjustment problem. Int. Trans. Oper. Res. 2 (2), 165–179.
Flötteröd, G., Bierlaire, M., Nagel, K., 2011. Bayesian demand calibration for dynamic traffic simulations. Transport. Sci. 45 (4), 541–561.
Frederix, R., Viti, F., Corthout, R., Tampère, C., 2011. New gradient approximation method for dynamic origin-destination matrix estimation on congested networks. Transport. Res. Record: J. Transport. Res. Board (2263), 19–25.
García Fernández, F.J., Verleysen, M., Lee, J.A., Díaz Blanco, I., 2013. Stability comparison of dimensionality reduction techniques attending to data and parameter variations, In: Eurographics Conference on Visualization (EuroVis)(2013)', The Eurographics Association.
Ghali, M., Smith, M., 1995. A model for the dynamic system optimum traffic assignment problem. Transport. Res. B: Methodol. 29 (3), 155–170.
Hazelton, M.L., 2008. Statistical inference for time varying origin-destination matrices. Transport. Res. B: Methodol. 42 (6), 542–552.
Huang, S., Sadek, A.W., Guo, L., 2012. Computational-based approach to estimating travel demand in large-scale microscopic traffic simulation models. J. Comput. Civil Eng. 27 (1), 78–86.
Iqbal, M.S., Choudhury, C.F., Wang, P., González, M.C., 2014. Development of origin-destination matrices using mobile phone call data. Transport. Res. C: Emerg. Technol. 40, 63–74.
Jha, M., Gopalan, G., Garms, A., Mahanti, B., Toledo, T., Ben-Akiva, M., 2004. Development and calibration of a large-scale microscopic traffic simulation model. Transport. Res. Record: J. Transport. Res. Board (1876), 121–131.
Jin, W.-L., 2012. A link queue model of network traffic flow. arXiv preprint arXiv:1209.2361.

Josefsson, M., Patriksson, M., 2007. Sensitivity analysis of separable traffic equilibrium equilibria with application to bilevel optimization in network design. Transport. Res. B: Methodol. 41 (1), 4–31.
Kattan, L., Abdulhai, B., 2006. Noniterative approach to dynamic traffic origin-destination estimation with parallel evolutionary algorithms. Transport. Res. Record: J. Transport. Res. Board (1964), 201–210.
Kim, H., Baek, S., Lim, Y., 2001. Origin-destination matrices estimated with a genetic algorithm from link traffic counts. Transport. Res. Record: J. Transport. Res. Board (1771), 156–163.
Lawson, C.L., Hanson, R.J., 1995. *Solving least squares problems*. SIAM.
LeBlanc, L.J., Farhangian, K., 1982. Selection of a trip table which reproduces observed link flows. Transport. Res. B: Methodol. 16 (2), 83–88.
Lee, J.-B., Ozbay, K., 2009. New calibration methodology for microscopic traffic simulation using enhanced simultaneous perturbation stochastic approximation approach. Transport. Res. Record: J. Transport. Res. Board (2124), 233–240.
Li, L., Huang, W., Lo, H.K., 2018. Adaptive coordinated traffic control for stochastic demand. Transport. Res. C: Emerg. Technol. 88, 31–51.
Lu, C.-C., Zhou, X., Zhang, K., 2013. Dynamic origin-destination demand flow estimation under congested traffic conditions. Transport. Res. C: Emerg. Technol. 34, 16–37.
Lu, L., Xu, Y., Antoniou, C., Ben-Akiva, M., 2015. An enhanced SPSA algorithm for the calibration of dynamic traffic assignment models. Transport. Res. C: Emerg. Technol. 51, 149–166.
Lu, X., Han, B., Hori, M., Xiong, C., Xu, Z., 2014. A coarse-grained parallel approach for seismic damage simulations of urban areas based on refined models and GPU/CPU cooperative computing. Adv. Eng. Softw. 70, 90–103.
Ma, W., Qian, Z.S., 2017. On the variance of recurrent traffic flow for statistical traffic assignment. Transport. Res. C: Emerg. Technol. 81, 57–82.
Ma, W., Qian, Z.S., 2018. Statistical inference of probabilistic origin-destination demand using day-to-day traffic data. Transport. Res. C: Emerg. Technol. 88, 227–256.
Nguyen, S., 1977. Estimating and OD Matrix from Network Data: a Network Equilibrium Approach, Montréal: Université de Montréal, Centre de recherche sur les transports.
Nie, Y.M., Zhang, H.M., 2008. A variational inequality formulation for inferring dynamic origin–destination travel demands. Transport. Res. B: Methodol. 42 (7), 635–662.
Nie, Y.M., Zhang, H.M., 2010. A relaxation approach for estimating origin-destination trip tables. Netw. Spatial Econ. 10 (1), 147–172.
Qian, Z.S., Shen, W., Zhang, H., 2012. System-optimal dynamic traffic assignment with and without queue spillback: its path-based formulation and solution via approximate path marginal cost. Transport. Res. B: Methodol. 46 (7), 874–893.
Qian, Z., Zhang, H.M., 2011. Computing individual path marginal cost in networks with queue spillbacks. Transp. Res. Rec. 2263 (1), 9–18.
Rao, W., Wu, Y.-J., Xia, J., Ou, J., Kluger, R., 2018. Origin-destination pattern estimation based on trajectory reconstruction using automatic license plate recognition data. Transport. Res. C: Emerg. Technol. 95, 29–46.
Shen, W., Wynter, L., 2012. A new one-level convex optimization approach for estimating origin–destination demand. Transport. Res. B: Methodol. 46 (10), 1535–1555.
Srivastava, N., Salakhutdinov, R.R., 2012. Multimodal learning with deep boltzmann machines. In: Advances in neural information processing systems, pp. 2222–2230.
Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9.
Tavana, H. 2001. Internally-consistent estimation of dynamic network origin-destination flows from intelligent transportation systems data using bi-level optimization.
Th, M., Sahu, S., Anand, A., 2015. Evaluating distributed word representations for capturing semantics of biomedical concepts. In: Proceedings of BioNLP 15, pp. 158–163.
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. Transport. Res. C: Emerg. Technol. 55, 231–245.
Van Der Zijpp, N., 1997. Dynamic origin-destination matrix estimation from traffic counts and automated vehicle identification data. Transport. Res. Record: J. Transport. Res. Board (1607), 87–94.
Vaze, V., Antoniou, C., Wen, Y., Ben-Akiva, M., 2009. Calibration of dynamic traffic assignment models with point-to-point traffic surveillance. Transport. Res. Record: J. Transport. Res. Board (2090), 1–9.
Verbas, İ., Mahmassani, H., Zhang, K., 2011. Time-dependent origin-destination demand estimation: challenges and methods for large-scale networks with multiple vehicle classes. Transport. Res. Record: J. Transport. Res. Board (2263), 45–56.
Xu, Y., Tan, G., Li, X., Song, X., 2014. Mesoscopic traffic simulation on cpu/gpu. In: Proceedings of the 2nd ACM SIGSIM/PADS conference on Principles of advanced discrete simulation. ACM, pp. 39–50.
Yang, H., 1995. Heuristic algorithms for the bilevel origin-destination matrix estimation problem. Transport. Res. B: Methodol. 29 (4), 231–242.
Yang, H., Meng, Q., Bell, M.G., 2001. Simultaneous estimation of the origin-destination matrices and travel-cost coefficient for congested networks in a stochastic user equilibrium. Transport. Sci. 35 (2), 107–123.
Yang, H., Sasaki, T., Iida, Y., Asakura, Y., 1992. Estimation of origin-destination matrices from link traffic counts on congested networks. Transport. Res. B: Methodol. 26 (6), 417–434.
Yen, J.Y., 1971. Finding the k shortest loopless paths in a network. Manage. Sci. 17 (11), 712–716.
Zhang, H., Nie, Y., Qian, Z., 2008. Estimating time-dependent freeway origin-destination demands with different data coverage: sensitivity analysis. Transport. Res. Record: J. Transport. Res. Board (2047), 91–99.
Zhang, M., Nie, Y., Shen, W., Lee, M.S., Jansuwan, S., Chootinan, P., Pravinvongvuth, S., Chen, A., Recker, W.W., 2008. Development of a path flow estimator for inferring steady-state and time-dependent origin-destination trip matrices. Caltrans Final Rep. TO 5502.
Zhou, X., Mahmassani, H.S., 2006. Dynamic origin-destination demand estimation using automatic vehicle identification data. IEEE Trans. Intell. Transport. Syst. 7 (1), 105–114.
Zhou, X., Mahmassani, H.S., 2007. A structural state space model for real-time traffic origin-destination demand estimation and prediction in a day-to-day learning framework. Transport. Res. B: Methodol. 41 (8), 823–840.
Zhou, X., Qin, X., Mahmassani, H., 2003. Dynamic origin-destination demand estimation with multiday link traffic counts for planning applications. Transport. Res. Record: J. Transport. Res. Board (1831), 30–38.