Transportation Research Part C 96 (2018) 321–346

Contents lists available at ScienceDirect

Transportation Research Part C

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

Hierarchical travel demand estimation using multiple data sources: A forward and backward propagation algorithmic framework on a layered computational graph

Xin Wu[b], Jifu Guo[a,*], Kai Xian[a], Xuesong Zhou[a,b,*]

[a] Beijing Transport Institute, No. 9 LiuLiQiao South Lane, Fengtai District, Beijing, China
[b] School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, AZ 85281, USA

ARTICLE INFO

Keywords:
Travel demand estimation
Multi-source data
Back propagation
Computational graph

ABSTRACT

Aiming to develop a theoretically consistent framework to estimate travel demand using multiple data sources, this paper first proposes a multi-layered Hierarchical Flow Network (HFN) representation to structurally model different levels of travel demand variables including trip generation, origin/destination matrices, path/link flows, and individual behavior parameters. Different data channels from household travel surveys, smartphone type devices, global position systems, and sensors can be mapped to different layers of the proposed network structure. We introduce Big data-driven Transportation Computational Graph (BTCG), alternatively Beijing Transportation Computational Graph, as the underlying mathematical modeling tool to perform automatic differentiation on layers of composition functions. A feedforward passing on the HFN sequentially implements 3 steps of the traditional 4-step process: trip generation, spatial distribution estimation, and path flow-based traffic assignment, respectively. BTCG can aggregate different layers of partial first-order gradients and use the back-propagation of "loss errors" to update estimated demand variables. A comparative analysis indicates that the proposed methods can effectively integrate different data sources and offer a consistent representation of demand. The proposed methodology is also evaluated under a demonstration network in a Beijing subnetwork.

1. Introduction

The overarching rationale of transportation system intelligence is that developments in sensing, cyber-physical infrastructures, and crowdsourcing big data technologies can be integrated to effective use for improving the performance of transportation systems. There are growing interests to adapt and apply a more successful set of computational tools (e.g., TensorFlow and Theano) in travel demand modeling. Along this line, transportation planners and modelers hope to utilize deep learning methodologies to further understand the inherent traveler choice decisions hierarchically using heterogonous data sources.

In the field of transportation network modeling, travel demand is represented sequentially by 4-step traveler choice decisions: trip generation (the origin to travel from), trip distribution (the destination to travel to), mode split (the mode by which to travel) and traffic assignment (the route and link on which to travel) (Small et al., 2007). Faced with multiple data sources from the emerging big

* Corresponding authors at: School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, AZ 85281, USA (X. Zhou).
E-mail addresses: Xinwu-griever@outlook.com (X. Wu), guojf@bjtrc.org.cn (J. Guo), xiank@bjtrc.org.cn (K. Xian), xzhou99@gmail.com (X. Zhou).

https://doi.org/10.1016/j.trc.2018.09.021
Received 14 May 2018; Received in revised form 19 September 2018; Accepted 21 September 2018
Available online 09 October 2018

data environment, planners critically need a theoretically consistent framework to structurally estimate different layers of travel demand. This research specifically considers the Traffic Demand Flow Estimation problem, which aims to infer the number of persons/vehicles traveling between a particular origin and destination via a particular route/link. A simultaneous estimation problem of traffic demand flows and user behavior coefficients can be also denoted as "traffic demand flow estimation" (TDFE) in this paper.

1.1. Existing origin/destination matrix estimation problems

Early TDFE models focus on estimating origin/destination (OD) matrices from traffic counts (Willumsen, 1978; Zuylen and Willumsen, 1980). In the classic 4-step method, the traffic assignment problem and OD matrices estimation (ODME) problem can be viewed as a pair of forward and inverse problems. Specifically, the forward problem allocates OD trips on different routes and calculates traffic link volume (Sheffi, 1985). On the other hand, the inverse problem estimates the OD trip matrix from observed flows. In a more advanced bi-level programming framework (Nguyen, 1977; Tavana, 2001), the lower level can be a dynamic traffic assignment (DTA) model, while the upper-level model uses a generalized least squares estimator to obtain an updated value of the OD matrix using a demand-dependent link proportion matrix. To consider more data sources, Zhou et al. (2003) and Zhou and Mahmassani (2006, 2007) utilized multi-day traffic counts and automated vehicle identification (AVI) data. Another research line of ODME problems is to develop an integrated model to estimate OD matrices and the behavior coefficient that reflects the variation in route choices among users (Liu and Fricker, 1996a,b). In one of the classical papers for OD demand estimation, Yang et al. (2001) considered the simultaneous traffic demand and behavioral coefficient estimation problem as a set of composite (possibly non-convex) functions, and the non-convexity of the simultaneous estimation problem could be computationally challenging, especially for large-scale networks.

A very important computational instrument in general traffic network modeling is the gradient-based calculation. For example, by applying the analytical approach proposed by Ghali and Smith (1995) in an ODME model for congested bottlenecks, Lu et al. (2013) evaluated the partial derivatives of different measurement types with respect to an additional unit of path flow. The marginal analysis, to be further extended in this paper, is also the foundation of other variational inequality approaches and supply chain analytics (Nagurney et al., 2013; Wu et al., 2018). Focusing on obtaining numerical derivatives or gradients through a limited number of simulation runs, many studies such as Balakrishna and Koutsopoulos (2008), Cipriani et al. (2011), Lu et al. (2015), Tympakianaki et al. (2015), and Antoniou et al. (2015) have used various gradient approximation methods within a simultaneous perturbation stochastic approximation (SPSA) framework.

1.2. Background of new big data resources

The typical traffic OD matrix estimation problem can be underdetermined because there are many possible combinations of demand patterns corresponding to the same observed flow values (Dafermos and Nagurney, 1984, Frederix et al., 2011). To address this issue, many studies, e.g., Frederix (2012), discussed how to incorporate knowledge about OD matrices or route choice behaviors, and a number of data-driven traffic estimation approaches (Toole et al., 2015; Shi and Abdel-Aty, 2015; Mudigonda and Ozbay, 2015; Antoniou et al., 2016; Ge and Fukuda, 2016; Carrese et al., 2017; Hu et al., 2017; Yang et al., 2018) aim to use a wide range of emerging data sources as listed below.

(1) Household travel surveys

Travel surveys together with census tracts have been the empirical foundation of many classical models that capture many important demographic characteristics (Sheffi, 1985). Traditional travel surveys are expensive in their own right, and a typical survey cycle is 5–10 years in even the most developed cities (Toole et al., 2015). Household travel surveys can provide trip production and abstraction from zones and sometimes OD trip tables between the zones.

(2) Mobile phone sample data

Mobile phone data records, typically available in an aggregated form to avoid privacy concerns, can capture many essential characteristics of human movement patterns. A number of studies are devoted to finding how to incorporate such data into activity-based models to estimate trip chain behaviors (Hao et al., 2017; Yin et al., 2018). Researchers have made significant progress in utilizing mobile phone data for inferring human mobility patterns (González et al., 2008; Wu et al., 2015; Toole et al., 2015) and identifying the relationship between land use and spatiotemporal distribution of people (Bauer et al., 2012).

However, (i) while mobile devices are available to passively provide large samples, they do not contain demographic information due to privacy reasons, and possible demographic biases (e.g. income or age bias) and sample penetration rates should be carefully examined. (ii) The spatial granularity of mobile phone data depends on the number of cell towers to which call records are mapped, e.g., with a resolution of 200–2000 m, so it is generally difficult to infer choices of mode and further conduct link/corridor analyses.

(3) Floating car data and vehicle location/identification data

Global Positioning Systems (GPS) and other location-based services can record vehicles' locations second by second. The taxis or buses equipped with GPS receivers are called floating cars in many practical applications, especially in Asia. Floating car data have a

Table 1
Comparison of different data sources.

Characteristics	Household travel surveys	Mobile phone data	Floating car data	Sensor data
Time period	5–10 years	Multi-hour periods within the day	individual hours of the day	Per 15 min of the day
Demand types	Aggregated and disaggregated	Aggregated	Aggregated sometime disaggregated traces	Aggregated
Sample penetration	Low	Depend on market share	Low	High
Spatial resolution	Household-based	200–2000 m	1–10 m	Link-based
Spatial coverage	Zone-based, individuals	Origin to destination	OD and path	Link

higher granularity than mobile phone data, and the derived average speed information can be used to estimate the observed travel time of links and establish urban traffic indexes (Guo et al., 2004; Zhao et al., 2010). However, the relatively low sample penetration of floating car data, especially when they mainly come from taxis and buses, might not be able to represent the population route choice behavior. Furthermore, to reduce and mitigate the possible vehicle location/identification errors of floating car data, a series of map matching and estimation algorithms have been proposed in the context of transportation network modeling (Zhou and Mahmassani, 2007; Tang et al., 2016).

(4) Sensor data at fixed locations

Large volumes of observed link counts can be collected from sensors including inductive loops, radars, cameras, etc. (Yang et al., 2006; Qiu et al., 2010; Han et al., 2012). Many existing approaches to estimate an OD demand matrix using link counts are associated with Wardrop equilibrium conditions (Willumsen, 1978). However, it has been widely recognized that Wardrop user equilibrium problems do not always have unique UE path flow patterns (and then OD flow patterns) even though the associated link flow patterns are determined (Tobin and Friesz, 1988). Further, the undetermined patterns might be aggravated in that most of the sensors only installed on freeway links as well as a significant percentage of fixed location sensors might not be fully functioning due to high maintenance costs and other reasons. The link count based OD estimation methods in transportation networks can be improved through fusing multiple data sources. For example, Wu et al. (2015) attempted to fuse mobile phone data and sensor data to estimate route flows.

Table 1 compares features of different data sources.

While big data sources present new opportunities to measure traffic demand patterns from different perspectives, urban planners must fully recognize and understand many error sources due to mismeasurement, poor sampling, miscomputation, and data aggregation (Zhao and Kockelman, 2002). It is necessary to develop a “consistent check tool” to cross-validate and fuse different information sources. For example, Bonnel et al. (2015) reported the high percentage disparity between the mobile phone data and those from the household travel surveys for each OD pair.

1.3. Background information in the field of deep learning

Deep learning technologies have been applied in a number of studies mainly focusing on traffic flow prediction (Dougherty, 1995; Park et al., 1998; Dia, 2001; Yin et al., 2002; Vlahogianni et al., 2005; Zhong et al., 2005; Zheng et al., 2006; Chan et al., 2012; Kumar et al., 2013). Interested readers can refer to Seo et al. (2017) to check recent developments in both model-driven and (stream) data-driven traffic state estimation approaches. Ermagun and Levinson (2018) also provided a comprehensive review of forecasting short-term traffic conditions using spatial information. Lv et al. (2015) demonstrated one of the applications of deep learning networks to uncover and identify hidden patterns from the observed traffic measurements as a time series from multiple days. Dixon et al. (2017) also attempted to develop a traffic-oriented deep learning framework for spatiotemporal modeling, while typically effective training is achieved by stochastic gradient descent and drop-out scheme.

It has been well recognized that the simple application of the artificial neural network (ANN) software package is insufficient in explaining the behavioral relationship between different traffic demand variables. Brathwaite et al. (2017) also indicated that the lack of economic interpretation in machine learning methods is one major concern for discrete choice modelers to adopt such models. Frosst and Hinton (2017) also indicated that deep learning networks traditionally depend on the use of hidden layers which make it hard to understand the functional role of each neuron. In this research, by focusing on traveler behavior and related decision making, our major goal is to develop a theoretically interpretable deep learning approach to estimate different layers of demand variables which have specific transportation meaning and are connected by explainable behavior coefficients.

(1) Deep learning network

With the use of layered structures, deep learning networks (Hinton and Sejnowski, 1986; Goodfellow et al., 2016) are widely applied to approximate the factorizations of complex composite mathematical functions. In a deep learning network, each layer is made of a stack of vertexes, and a vertex multiplies input by a set of weights. Through the vertex’s activation function, the forward passing in the deep learning network can determine to what extent the signal progresses through the network to affect the ultimate outcome. We would like to remark that deep learning networks are in fact closely related to discrete choice models used in traffic demand forecasting. For example, in the early McCulloch-Pitts Neuron model (McCulloch and Pitts, 1943), its linear classifier model

is similar to the conditional logit model used in mode split analysis (Small et al., 2007). In https://www.researchgate.net/publication/325126768, we provide an interesting illustrative example, adapted from Koppelman and Bhat (2006) in the context of transportation demand modeling, to show the relationship between the conditional logit model and a three-layer ANN. The example can be viewed as a short tutorial of deep learning and computational graph frameworks in transportation modeling.

(2) Computational graph-based modeling framework

The back-propagation (BP) algorithm has been reinvented dozens of times in different fields (Trefethen, 2005; Griewank, 2012). Further, the concept of **computational graphs** serves as a lower-level building block and a descriptive language for representing BP. Automated Differentiation (AD) is a set of computer techniques based on the chain rule of calculus. AD is used to numerically calculate the derivative of a function based on the fact that many functions can be broken down into a composition of elementary arithmetic operations involving just one or two arguments at a time. In the book by Wright and Nocedal (1999) about AD (chapter 8), the two-argument operations include addition, multiplication, division, and power with examples of the single-argument operations as exponential and logarithmic. In conjunction with the AD, computational graphs can provide a systematic modeling tool to calculate derivatives between different variables/parameters and further backpropagate the gradients throughout the graphs.

It is widely recognized that the use of the BP combined with any gradient descent algorithm plays a vital role in training deep learning networks (Rumelhart et al., 1986). As discussed in the textbook by Goodfellow et al. (2016), the universal approximation theorem shows that a single hidden layer neural network can approximate continuous function to any degree of precision, but it does not guarantee convergence to a globally optimal solution using the BP combined with gradient descent algorithm. On the other hand, the BP combined with gradient descent algorithm is still a widely used efficient algorithm that minimizes loss functions of ANNs with proper architectures (Montúfar et al., 2014).

For transportation modelers, we think it is important to understand, how the underlying computational graph of a deep learning network, in conjunction with the BP algorithm, can be used to describe the forward propagation and backward feedback processes between different levels of transportation planning and decision makings. Interested readers can find examples of computational graphs in classical books written by Goodfellow et al. (2016). We also refer readers to some tutorials on the relationship of computational graphs and the AD method in references by http://colah.github.io/posts/2015-08-Backprop/. In https://www.researchgate.net/publication/325126768, we also use the illustrative example adapted from Koppelman and Bhat (2006) to show how to extend the ANN of the conditional logit model into a computational graph abstraction.

1.4. Statement of contributions in the proposed method

This paper aims to uniquely combine the insights from deep learning methods with spatial characteristics and domain knowledge in transportation network modeling applications from the following perspectives:

(1) Along the research line of deep learning networks, this research proposes a modeling framework using a multi-layer Hierarchical Flow Networks (HFN) representation. This flow-oriented estimation formulation can structurally formulate the TDFE problem and simultaneously estimate different levels of unobservable or partially observed traffic demand variables and behavior parameters.
(2) By recognizing the multiple sources of information in emerging big data applications, we map different levels of traffic demand variables to various data sources in traffic demand estimation applications including household travel survey, mobile phone, floating car, and sensor data. The systematic linkage between each representation layer in HFN and individual sources enable planners to better conduct cross-validation and data fusion.
(3) To build a theoretically sound modeling framework, this paper hopes to trace back to the fundamental or low-level representation for deep learning networks and construct a transportation-focused computational graph as a structured modeling language. This modeling paradigm enables us to capture the mathematical structure inside the TDFE problem by representing and decomposing complex composite functions through a graph of current states and numerical gradients.
(4) To enable computational efficient solution methods, back-propagation sequentially aggregates the “loss errors” on the HFN to update multiple levels of demand variables and behavior parameters efficiently. It is interesting to comment that many new deep learning toolkits, such as TensorFlow (Abadi et al., 2016), have nicely offered easy-to-use capabilities to formulate and solve transportation data mining applications as computational graphs. By using this new set of tools, we can quickly represent modular structures and compute many compounded partial derivatives at different representation layers of the proposed HFN.

The remainder of the paper is organized in the following manner. Section 2 contains a description of the TDFE in a four-layer deep learning network: HFN. Section 3 formulates the TDFE problem using a nonlinear programming model. Section 4 proposes a novel computational graph modeling framework to extend the HFN and express the detailed mathematical formulations in the proposed nonlinear programming model. Section 5 proposes an implementable BP algorithm based on the computational graph. Section 6 evaluates the performance of the method through several numerical experiments with a discussion of some key features. A real-world case study in Beijing is implemented to demonstrate the applicability of the proposed framework. Fig. 1 shows the precedence relationship between sections and the basic concepts of this paper.

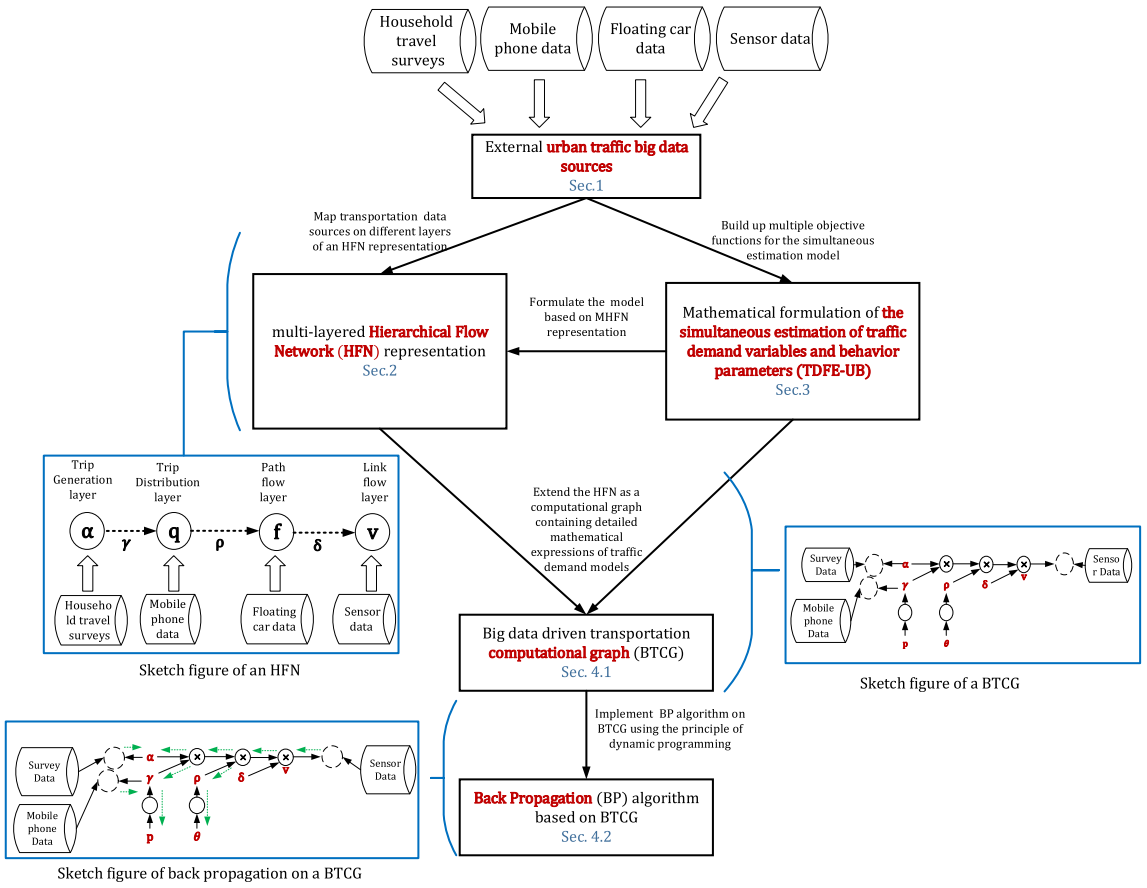


Fig. 1. The relationship between sections and basic concepts of this paper.

2. Problem statement based on multi-layered hierarchical flow network representation

In this section, we introduce the TDFE problem using an HFN representation. In the joint traffic demand flow and behavioral parameter estimation problem, we only consider the travel mode of private cars, while the proposed method can be extended to multimodal cases in future research.

2.1. Decision variables and observed traffic measurements

We formulate the TDFE problem on a directed transportation network (physical network) $\mathscr{G} = (\mathscr{N}, \mathscr{A})$, comprising of a node set $\mathscr{N}$ and a link set $\mathscr{A}$. Each link $a \in \mathscr{A}$ has observed travel time t_a and possible toll c_a. $\mathscr{Z}$ is defined as the set of traffic analysis zones that produce or attract traffic demand. In transportation modeling, traffic zones are represented by centroids of a certain urban area. The centroids are also called activity locations which are viewed as a special type of nodes (Sheffi, 1985). Once set $\mathscr{Z}$ is defined, the traffic flows of each origin-to-destination (OD) pair can be represented in terms of OD trips from the origin centroid to the destination centroid. Table 2 lists the estimation variables, other indices, and sets. The notations with the bar signs are associated with sensor measurements or external survey/mobile data.

The variables listed in Table 2 include **trip generation rate** $\boldsymbol{\alpha}$, **OD flows** $\mathbf{q}$, **path flows** $\mathbf{f}$ and **link flows** $\mathbf{v}$, respectively. **Choice behavior** is expressed by the matrix π and vector θ. The traffic measurements in Table 2 considered as the given input are generated as follows:

(1) Measurements associated with trip generation $\bar{X}_o^m$

We consider reference trip generation rates from a zone $o \in \bar{\mathscr{Z}} \subseteq \mathscr{Z}$, which can be derived using household travel surveys. For each household category with different household characteristics and travel purposes, one can calculate the number of trips produced from $o \in \bar{\mathscr{Z}} \subseteq \mathscr{Z}$ by multiplying the population size and trip rate using private cars. Then, a sample of reference trip generation $\bar{X}_o^m$ can be obtained by summing over the trip generation of different household categories (Hensher and Button, chapter 3, 2007; Patriksson, 2015).

Table 2
Sets, indexes, input data, and variables.

Sets	Definition
$\bar{\mathscr{A}}$	Subset of links with sensors in the transportation network
$\bar{\mathscr{Z}}$	Subset of zones that have reference trip generation generated from household travel survey
$\mathscr{W}_o$	Set of OD pairs originating from $o \in \mathscr{Z}$ in the transportation network
$\bar{\mathscr{W}}$	Subset of OD pairs that have reference OD split generated from mobile phone data
$\bar{\mathscr{W}}_o$	Subset of OD pairs starting from $o \in \mathscr{Z}$ that have reference OD split generated from mobile phone data
$\mathscr{R}_w$	Set of routes for OD pair $w = (o, d)$, where$\cup_{w \in \mathscr{W}} \mathscr{R}_w = \mathscr{R}$
Indexes	**Definitions**
o	Indexes of production zones in$\mathscr{Z}$
d	Indexes of destination zones in$\mathscr{Z}$
r	Indexes of routes in$\mathscr{R}$
w	Indexes of OD pairs in$\mathscr{W}$
a	Indexes of links in$\mathscr{A}$
m	Indexes of samples
Input data	**Definitions**
$\bar{X}_o^m$	The m^{th} sample of reference trip production of zone $o \in \bar{\mathscr{Z}}$ from surveys, where $m = 1, 2, \cdots M_1$
$\bar{P}_{ow}^m$	The m^{th} sample from phone data of reference OD split rates of $w \in \bar{\mathscr{W}}$ that originates from zone $o \in \mathscr{Z}$ where $m = 1, 2, \cdots M_2$
$\bar{X}_a^m$	The m^{th} sample of observed flow counts on link $a \in \bar{\mathscr{A}}$ from sensor data, where $m = 1, 2, \cdots M_3$
δ_{ra}	$\delta_{ra} = 1$if route r uses link a, and 0 otherwise
t_a	Observed link travel time on link $a \in \mathscr{A}$ generated from floating car data
c_a	Toll on link $a \in \mathscr{A}$
Estimation variables	**Definitions**
X_o	Estimated trip generations from zone$o \in \mathscr{Z}$
X_w	Estimated OD volumes of$w \in \mathscr{W}$
X_r	Estimated path flows loaded on path$r \in \mathscr{R}$
X_a	Estimated link flows on links $a \in \mathscr{A}$ (including links with and without sensors)
π_{ow}	Estimated probability of OD w in set $\mathscr{W}_o$, **before normalization**
P_{ow}	**Normalized** estimated probability of users who passes OD $w \in \mathscr{W}$ from $o \in \mathscr{Z}$, $\sum_{w \in \mathscr{W}} P_{ow} = 1$
P_{wr}	**Normalized** estimated probability of users who passes route $r \in \mathscr{R}_w$ between OD pair$w \in \mathscr{W}$
θ_w	Estimated value of time (VOT) to translate time to monetary cost
α	The vector of trip generation rate, $\alpha = (X_o \vert o \in \mathscr{Z})$
$\mathbf{q}$	The vector of OD flows $\mathbf{q}$, $\mathbf{q} = (X_w \vert w \in \mathscr{W})$
$\mathbf{f}$	The vector of path flows $\mathbf{f}$, $\mathbf{f} = (X_r \vert r \in \mathscr{R})$
$\boldsymbol{\gamma}$	The matrix of for **normalized** OD split $\boldsymbol{\gamma} = (P_{ow} \vert o \in \mathscr{Z}, w \in \mathscr{W})$
$\boldsymbol{\rho}$	The matrix of path flow proportion$\boldsymbol{\rho} = (P_{wr} \vert w \in \mathscr{W}, r \in \mathscr{R})$
$\boldsymbol{\pi}$	The matrix of for OD split $\boldsymbol{\pi} = (\pi_{ow} \vert o \in \mathscr{Z}, w \in \mathscr{W})$
$\boldsymbol{\theta}$	The vector of estimated VOT in logit model$\boldsymbol{\theta} = (\theta_w \vert w \in \mathscr{W})$

(2) Measurements associated with OD split rate $\bar{P}_{ow}^m$

Reference OD split rates can be generated from mobile phone data that match to the given zoning system. Firstly, the trips between the start and end zones can be deduced from the records of mobile phone data through a hypothesis of the minimal duration of stationary activities within a cell-tower location area. Secondly, we can make proper assumptions to match the records of a cell-tower location area onto different zones. We can also partition a trip chain into a series of OD pairs based on a proper stationary activity assumption (Bonnel et al., 2015). Because it is technically difficult to identify the mode of transportation using mobile phone data, in this paper, we only use the OD split rate $\bar{P}_{ow}^m$, which can be interpreted as the m^{th} sample of the proportion of the journey records between the OD pair $w \in \bar{\mathscr{W}}_o$. Obviously, for each $o \in \mathscr{Z}$, $\sum_{w \in \bar{\mathscr{W}}} \bar{P}_{ow}^m = 1$, and $0 \le \bar{P}_{ow}^m \le 1$.

(3) Measurements associated with flow counts $\bar{X}_a^m$

Sensor flow counts are collected from a subset of links $\bar{\mathscr{A}} \subseteq \mathscr{A}$ with sensors.

(4) Candidate path set $\mathscr{R}$ and observed travel time t_a

Floating car (GPS) records can be matched onto traffic networks using map matching algorithms (Chen et al., 2016, Tang et al., 2016). Then, we can apply the corrected floating car data to generate the estimated travel time of link $a \in \mathscr{A}$ and candidate path set R in the transportation network. As floating car records cannot cover all important paths in a traffic network, we could generate K-

shortest paths according to a pre-specified rule to prepare a complete candidate path set. Ramming (2002) reviewed some of the common techniques used by transportation practitioners for generating route choice sets of auto users.

2.2. Description of the multi-layered hierarchical flow network

An HFN representation is used as a high-level modeling abstract to formulate the TDFE problem. Let an HFN $G = G(\boldsymbol{V}, \boldsymbol{E})$ be the collection of all elements of traffic demand variables at different layers, where each layer controls a subset of the demand variables and receives network flows from its upper layers. Let $\boldsymbol{V} = \mathscr{Z} \cup \mathscr{W} \cup \mathscr{R} \cup \mathscr{A}$ be the sets of vertexes arranged in different layers.

(1) Definition of vertexes:
- (i) The first layer is trip generation layer $\mathscr{Z}$ containing all zones corresponding to demand variables $\boldsymbol{\alpha}$.
- (ii) The second layer is trip distribution layer $\mathscr{W}$ containing all OD pairs corresponding to demand variables $\mathbf{q}$.
- (iii) The third layer is the path flow layer $\mathscr{R}$ containing all paths in a candidate set corresponding to demand variables $\mathbf{f}$.
- (iv) The fourth layer is the link flow layer $\mathscr{A}$ containing all links corresponding to demand variables $\mathbf{v}$.
- (v) The edges in the graph G is defined as $\boldsymbol{E} = \boldsymbol{E}_{ZW} \cup \boldsymbol{E}_{WR} \cup \boldsymbol{E}_{RA}$ to specify the connections between vertexes

(2) Definition of edges:
- (i) $\boldsymbol{E}_{ZW}$ contains edges connecting vertexes in $\mathscr{Z}$ and $\mathscr{W}$, where each edge corresponds to an **OD split rate** P_{ow} in matrix $\boldsymbol{\gamma} = (P_{ow} | o \in \mathscr{Z}, w \in \mathscr{W})$.
- (ii) $\boldsymbol{E}_{WR}$ contains edges connecting vertexes in $\mathscr{W}$ and $\mathscr{R}$, where each edge corresponds to a **route choice proportion** P_{wr} in matrix $\boldsymbol{\rho} = (P_{wr} | w \in \mathscr{W}, r \in \mathscr{R})$.
- (iii) $\boldsymbol{E}_{RA}$ contains edges connecting vertexes in $\mathscr{R}$ and $\mathscr{A}$, where each edge corresponds to a **link-route incidence parameter** δ_{ra} in **link-route incidence matrix** $\delta = (\delta_{ra} | r \in \mathscr{R}, a \in \mathscr{A})$.

Fig. 2 shows an example of an HFN, in two different styles. In Fig. 2(b), we plot every demand variable as a vertex in the HFN. As shown in Fig. 2(a), we have the following flow conservation equations in traffic demand modeling.

$$\boldsymbol{\alpha} \times \boldsymbol{\gamma} = \mathbf{q} \tag{1}$$

$$\mathbf{q} \times \boldsymbol{\rho} = \mathbf{f} \tag{2}$$

$$\mathbf{f} \times \delta = \mathbf{v} \tag{3}$$

Eq. (1) describes the process of how trip production from a zone is distributed onto different OD pairs. Eq. (2) maps the flow from an OD pair to the candidate routes. Eq. (3) aggregates path flows to link flows. As shown in Fig. 2(a), specifically, each layer of vertexes correspond to different estimation demand variables. In Fig. 2(b), we draw a vertex in a compact form in the HFN for each entire vector, with the name of the parameters on the edges describing the relationship between two layers of vertexes.

From a deep learning network perspective, each vertex of the HFN becomes a neuron with a rectified linear unit (ReLU) activation function: $f(x) = \max(0, x)$, which expresses both non-negativity and flow conservation of traffic demand variables. The ReLU function is nearly linear, so it preserves many of the properties that enable a straightforward optimization of linear models by gradient-based methods (Goodfellow et al., 2016). Each edge corresponds to a connection between neurons to describe the mapping from the output of a neuron to the input of a neuron.

3. Nonlinear optimization model for the simultaneous model

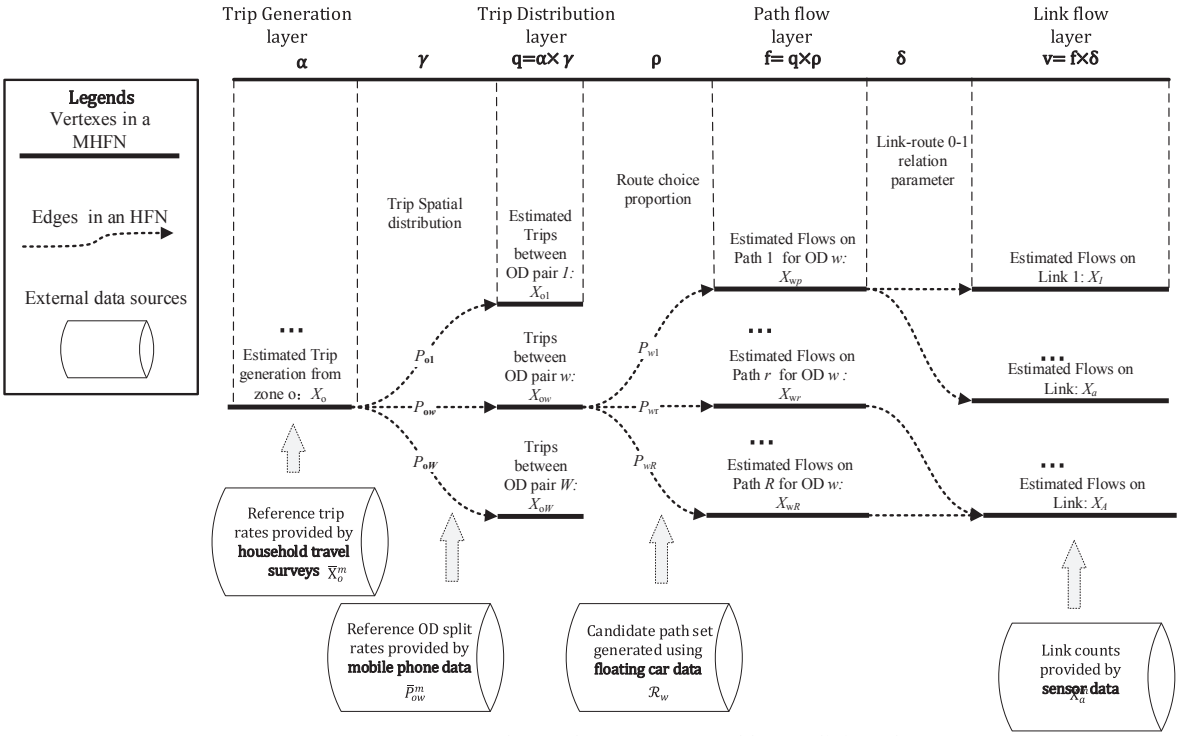
Based on the constructed HFN and given data sources, this section aims to formulate the estimation model as an optimization program, which can be further transformed as a computational graph for lower-level numerical calculations. By using the HFN representation, we can capture the relationship between different data sources and the estimation variables/parameters. Specifically, household travel survey, mobile phone, and sensor data are mapped to different layers of the HFN using mean square error (MSE) objective functions $\mathrm{F}_1(\boldsymbol{\alpha})$, $\mathrm{F}_2(\boldsymbol{\gamma})$, and $\mathrm{F}_3(\mathbf{v})$, respectively.

$$\mathrm{F}_1(\boldsymbol{\alpha}) = \frac{1}{2M_1} \sum_{m=1}^{M_1} \sum_{o \in \mathscr{Z}} (\mathrm{X}_o - \bar{\mathrm{X}}_o^m)^2 \tag{4}$$

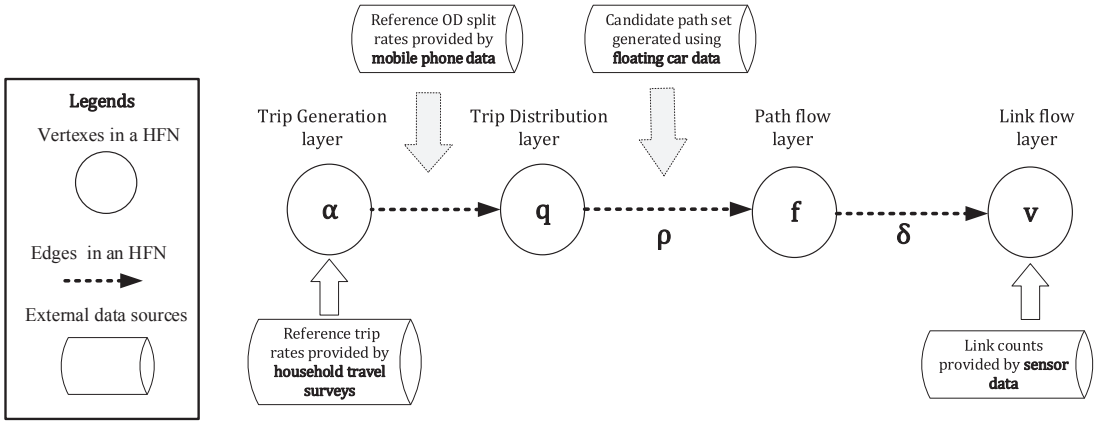
$$\mathrm{F}_2(\boldsymbol{\gamma}) = \frac{1}{2M_2} \sum_{m=1}^{M_2} \sum_{o \in \mathscr{Z}} \sum_{w \in \mathscr{W}_o} (P_{ow} - \bar{P}_{ow}^m)^2 \tag{5}$$

$$\mathrm{F}_3(\mathbf{v}) = \frac{1}{2M_3} \sum_{m=1}^{M_3} \sum_{a \in \mathscr{A}} (\mathrm{X}_a - \bar{\mathrm{X}}_a^m)^2 \tag{6}$$

$\mathrm{F}_1(\boldsymbol{\alpha})$ implies the deviation between reference trip generation from household travel surveys and estimated trip generation. $\mathrm{F}_2(\boldsymbol{\gamma})$ indicates the deviation between the reference OD split rate from mobile phone data and the estimated trip spatial distribution rate. $\mathrm{F}_3(\mathbf{v})$ corresponds to the deviation between sensor counts and link flows and is to be estimated for a set of links with measurements. Thus, we can minimize the sum of three MSE loss functions:



(a) Four layered HFN expressed by explicit style



(b) Four layered MHFN expressed by vectorized style

Fig. 2. The four-layered HFN representation drawn in two different styles.

$$\min F(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{v}) = \lambda_1 F_1(\boldsymbol{\alpha}) + \lambda_2 F_2(\boldsymbol{\gamma}) + \lambda_3 F_3(\mathbf{v}) \tag{7}$$

The above **objective function** can be viewed as a **cross-validation** strategy which aims to reduce the generalization error of the learning model. The objective function attempts to learn three tasks at the same time using jointly weights λ_1, λ_2, and λ_3. The values of the weights can be used to reflect the prior preferences for different data sources. Zhou et al. (2003) discussed a number of weight selection strategies from a multi-criteria decision perspective. Fig. 3 illustrates how different types of data sources are mapped to different layers of the HFN representation.

Then, we can formulate the simultaneous estimation model based on the HFN representation.

TDFE model:

$$\min F(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{v}) = \lambda_1 F_1(\boldsymbol{\alpha}) + \lambda_2 F_2(\boldsymbol{\gamma}) + \lambda_3 F_3(\mathbf{v})$$

(1) Flow conservation constraints

$$X_o P_{ow} = X_w \ \forall\, w \in \mathscr{W} o \in \mathscr{Z} \tag{8}$$

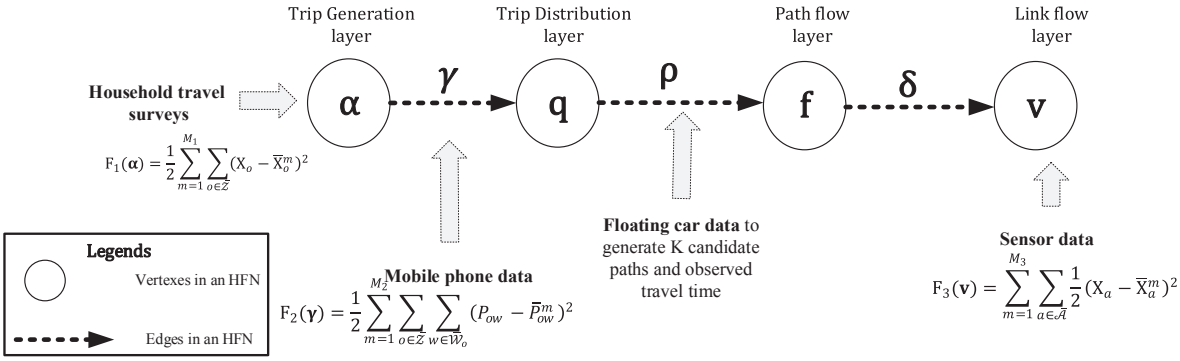


Fig. 3. Map different data sources to the HFN representation.

$$X_w P_{wr} = X_r \ \forall\, r \in \mathscr{R} w \in \mathscr{W} \tag{9}$$

$$\sum_{r\in\mathscr{R}} \delta_{ra} X_r = X_a \ \forall\, a \in \mathscr{A} \tag{10}$$

The above constraints are equal to Eqs. (1), (2) and (3) embedded in HFNs. It should be noted that the above flow conservation constraints using the product form can also be reformulated as a series of **linear** multi-commodity flow conservation constraints. This can be done using a path flow form of $x(o, d, r)$ as path flow volume from origin o to destination d using the kth path, as the sum of path flow is the OD flow and the sum of OD flow over different destinations is the origin flow. One can refer to the study by Zhou and Mahmassani (2007), where the OD split factor in $F_2(\gamma)$ is handled as a highly nonlinear (and possible computationally involving) fractional form of $x(o, d, r)$.

(2) Constraints to describe choice probability and behaviors of users

$$P_{ow} = \frac{\pi_{ow}}{\sum_{w\in\mathscr{W}_o}\pi_{ow}}, \quad \forall\, w \in \mathscr{W}_o, \quad o \in \mathscr{Z} \tag{11}$$

$$P_{wr} = \frac{\exp(-\theta_w \sum_{a\in\mathscr{A}}\delta_{ra} t_a - \sum_{a\in\mathscr{A}}\delta_{ra} c_a)}{\sum_{r\in\mathscr{R}_w}\exp(-\theta_w \sum_{a\in\mathscr{A}}\delta_{ra} t_a - \sum_{a\in\mathscr{A}}\delta_{ra} c_a)}, \quad \forall\, r \in \mathscr{R}_w, \quad w \in \mathscr{W} \tag{12}$$

where Eq. (12) implements a conditional logit based stochastic network loading that only considers parameter θ_w in the conditional logit model. The parameter $\theta_w \geq 0$ measures the sensitivity of route choices to travel time. As $\theta_w \to +\infty$, route choices are concentrated on the shortest time route in each set $\mathscr{R}_w$. As $\theta_w \to 0$, the probabilities are determined by toll differences.

We can bring back all the constraints Eqs. (8)–(12) to the MSE loss function $F(\alpha, \gamma, \mathbf{v})$, because $\mathbf{v}$ is the dependent variable of $\mathbf{f}$; $\mathbf{f}$ is a function of $\mathbf{q}$ and ρ; $\mathbf{q}$ is a function of α and γ; γ is a function of π; and ρ is the dependent variable of θ. Then

$$\mathbf{v} = \mathbf{v}(\mathbf{f}) = \mathbf{v}(\mathbf{f}(\mathbf{q}, \rho)) = \mathbf{v}(\mathbf{f}(\mathbf{q}(\alpha, \gamma(\pi)), \rho(\theta))) \tag{13}$$

The model is transformed into a non-convex optimization problem with only non-negativity constraints:

$$\min F(\alpha, \pi, \mathbf{v}) = \min F_1(\alpha) + F_2(\gamma(\pi)) + F_3(\mathbf{v}(\mathbf{f}(\mathbf{q}(\alpha, \gamma(\pi)), \rho(\theta)))) \tag{14}$$

In Appendix A, we examine the composite and non-convexity properties of the simultaneous estimation model.

4. Big data-driven transportation computational graph framework

As the mathematical formulations of Eqs. (11) and (12) are not expressed in the proposed HFN representation, in this section, we extend the HFN as a computational graph to express the choice probability and behaviors of users. As the computation graph "inherits" the relationship between the HFN and different transportation big data sources (as shown in Fig. 3), we call the graph "Big data-driven Transportation Computational Graph" (BTCG). In BTCG, we implement forward passing and BP to update the estimation variables to approximate the complex functional relationship expressed by Eq. (14). The details about the procedure will be discussed in Section 5. As the BP is the essential part of the procedure, we use the term BP algorithm to represent the overall procedure throughout this paper.

4.1. Computational graph framework to enable automated differentiation and back-propagation on the HFN

Now we construct a computational graph $G(\boldsymbol{V}_c, \boldsymbol{A}_c)$ that depends on the proposed HFN $G(\boldsymbol{V}, \boldsymbol{A})$ by adding an extra set of vertexes to express the mathematical structure of the TDFE model. Let $\boldsymbol{V}_c$ denote the set of vertexes derived from the set $\boldsymbol{V}$ in HFN $G(\boldsymbol{V}, \boldsymbol{A})$. Aside from the HFN representation, computational graphs create vertexes for all variables, parameters, and intermediary variables.

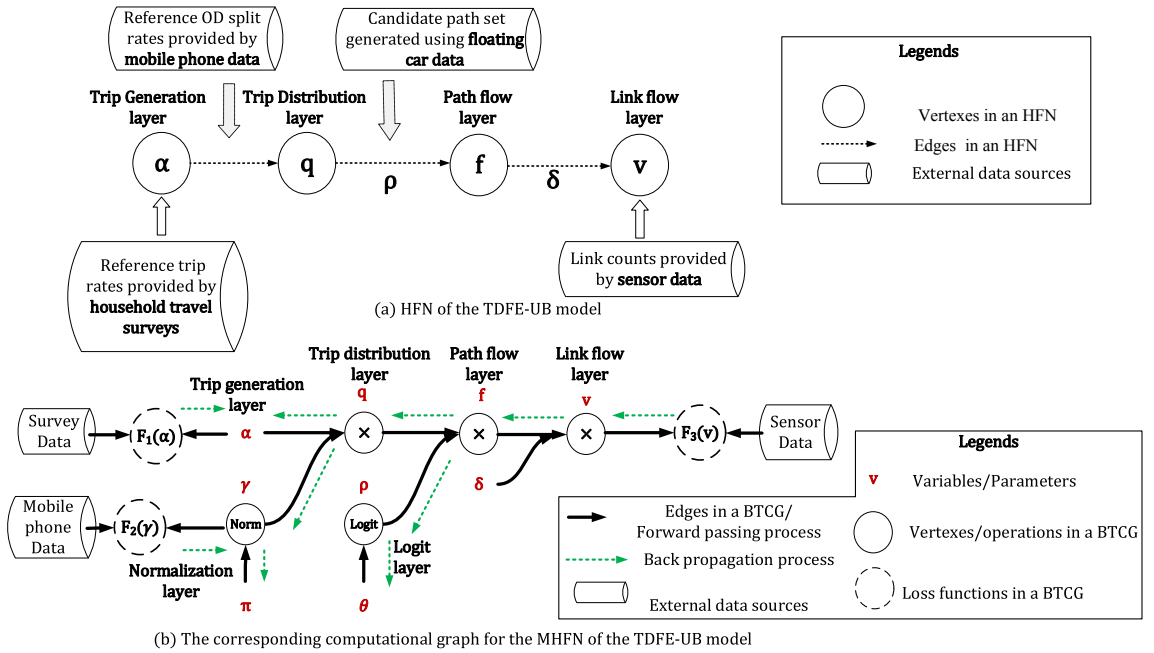


Fig. 4. The HFN representation for TDFE model and its corresponding BTCG.

The vertexes can indicate a scalar, vector, matrix, or tensor. $\boldsymbol{A}_c$ indicates the set of directed edges that describe different mathematical operations in the TDFE model. If a variable y is computed by applying an operation to a variable x, we draw a directed edge from the vertex of x to the vertex of y and annotate the vertex of y with the name of the operation. In a recursive fashion, more complicated functions can be described by composing many elementary operations together (Goodfellow et al., chapter 6, 2016).

Fig. 4 displays how an HFN can be extended as a BTCG using a vectorized style. As shown in Fig. 4, the computational graph consists of six types of layers. Each layer is comprised of a more detailed computational graph, which can be constructed easily using single-argument or two-argument operations. The link flow and path flow layer are used to relate link flows to path flows. The trip generation and trip distribution layers are used to describe trip production and trip spatial estimation. The multinomial logit layer describes the logit-based route choice proportion. A normalized layer is used to describe the normalization process of OD split rates.

As shown in BTCG, when traffic demand variable values are conveyed into the multi-layer hierarchical architecture (i.e., HFN), they are propagated forward through the BTCG, layer by layer, until they reach the output layers that are connected with external data sources. The outputs of the BTCG are compared with the given reference measurements provided by external data sources using MSE loss functions. Then, "MSE loss errors" are propagated backward, starting from the outputs until each vertex has an associated error value which represents its contribution to the deviations from the output layers. Illustrated by Fig. 5, the "loss errors" from sensor data should be back-propagated from the link flow layer to the path flow and other layers. This fashion is very similar to the OD flow or path flow adjustment process through the link proportions used in the common OD estimation methods. To compute the gradients of different variables with respect to loss functions, it is necessary to compute the partial derivative on each edge of the BTCG using reverse mode. There are four types of partial derivatives shown in Fig. 5. We list the mathematical expressions of the partial derivatives in Table 3.

Using "atoms" listed in Table 3, we can calculate many complex marginal values using the chain rule in calculus, for example,

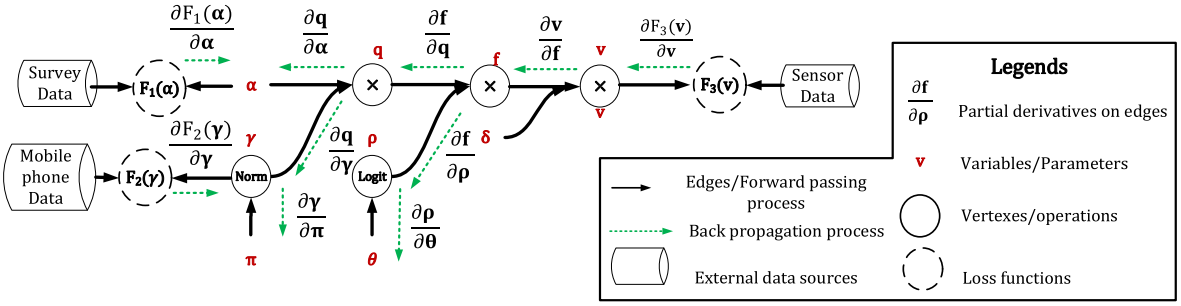


Fig. 5. Back-propagation of partial derivatives on edges of the BTCG.

Table 3
Partial derivatives in BTCG.

Types	Partial derivatives	Mathematical expressions	Dimensions
Type I Loss functions wrt. estimated variables	$\frac{\partial F_1(\alpha)}{\partial \alpha}$	$\frac{\partial F_1(\alpha)}{\partial X_o} = \frac{1}{M_1}\sum_{m=1}^{M_1}(X_o - \bar{X}_o^m)\ \forall o \in \mathscr{Z}$	$\lvert\mathscr{Z}\rvert$ vector
	$\frac{\partial F_2(\gamma)}{\partial \gamma}$	$\frac{\partial F_2(\gamma)}{\partial P_{ow}} = \frac{1}{M_2}\sum_{m=1}^{M_2}(P_{ow} - \bar{P}_{ow}^m)\ \forall w \in \mathscr{W}_o o \in \mathscr{Z}$	$\lvert\mathscr{Z}\rvert \times \lvert\mathscr{W}\rceil$ matrix
	$\frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}}$	$\frac{\partial F_3(\mathbf{v})}{\partial X_a} = \frac{1}{M_3}\sum_{m=1}^{M_3}(X_a - \bar{X}_a^m)\ \forall a \in \mathscr{A}$	$\lvert\mathscr{A}\rvert$ vector
Type II Lower layer demand variables wrt. upper layer demand variables	$\frac{\partial \mathbf{v}}{\partial \mathbf{f}}$	$\frac{\partial X_a}{\partial X_r} = \delta_{ra}\ \forall a \in \mathscr{A}, r \in \mathscr{R}$	$\lvert\mathscr{A}\rvert \times \lvert\mathscr{R}\rvert$ Jacobian matrix
	$\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$	$\frac{\partial X_r}{\partial X_w} = P_{wr}\ \forall r \in \mathscr{R}, w \in \mathscr{W}$	$\lvert\mathscr{R}\rvert \times \lvert\mathscr{W}\rceil$ Jacobian matrix
	$\frac{\partial \mathbf{q}}{\partial \alpha}$	$\frac{\partial X_w}{\partial X_o} = P_{ow}\ \forall w \in \mathscr{W}, o \in \mathscr{Z}$	$\lvert\mathscr{W}\rceil \times \lvert\mathscr{A}\rvert$ Jacobian matrix
Type III Demand variables wrt. choice probability	$\frac{\partial \mathbf{f}}{\partial \rho}$	$\frac{\partial X_r}{\partial P_{wr'}} = X_w\ \forall r = r' \in \mathscr{R}_w, w \in \mathscr{W}$	$\lvert\mathscr{R}\rvert \times \lvert\mathscr{W}\rceil \times \lvert\mathscr{R}\rvert$ tensor
	$\frac{\partial \mathbf{q}}{\partial \gamma}$	$\frac{\partial X_w}{\partial P_{ow'}} = X_o\ \forall w = w' \in \mathscr{W}_o, o \in \mathscr{Z}$	$\lvert\mathscr{W}\rceil \times \lvert\mathscr{Z}\rceil \times \lvert\mathscr{W}\rceil$ tensor
Type IV Choice probability wrt. behavioral parameters	$\frac{\partial \rho}{\partial \theta}$	$\frac{\partial P_{wr}}{\partial \theta_w} = P_{wr}\left[\sum_{r' \in \mathscr{R}_w}(\sum_{a \in \mathscr{A}} \delta_{r'a} t_a)P_{wr'} - (\sum_{a \in \mathscr{A}} \delta_{ra} t_a)\right] \forall r \in \mathscr{R}_w, \forall w = w' \in \mathscr{W}$	$\lvert\mathscr{R}\rvert \times \lvert\mathscr{W}\rceil$ Jacobian matrix
	$\frac{\partial \gamma}{\partial \pi}$	$\frac{\partial P_{ow}}{\partial \pi_{ow'}} = \begin{cases} \frac{1 - P_{ow'}}{\sum_{w'' \in \mathscr{W}_o} \pi_{ow''}}, \forall w = w' \in \mathscr{W}_o\ \forall o \in Z \\ -\frac{P_{ow}}{\sum_{w'' \in \mathscr{W}_o} \pi_{ow''}}, \forall w \neq w' \in \mathscr{W}_o\ \forall o \in Z \end{cases}$	$\lvert\mathscr{W}\rceil \times \lvert\mathscr{W}\rceil$ matrix

$$\frac{\partial F_3(\mathbf{v})}{\partial \alpha} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \times \frac{\partial \mathbf{q}}{\partial \alpha} \tag{15}$$

where $\frac{\partial F_3(\mathbf{v})}{\partial \alpha}$ is a $\lvert\mathscr{A}\rvert$ dimension vector of partial derivatives. We see that the marginal values consist of performing a Jacobian-gradient product for each operation in the computational graph.

When calculating $\frac{\partial F_3(\mathbf{v})}{\partial \Theta}$, we can apply the BP algorithm using a tensor form:

$$\frac{\partial F_3(\mathbf{v})}{\partial \Theta} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \rho} \times \frac{\partial \rho}{\partial \Theta} \tag{16}$$

which is a $\lvert\mathscr{W}\rceil \times \lvert\mathscr{W}\rceil$ dimension matrix of partial derivatives with values only on the diagonal line. Similarly

$$\frac{\partial F_3(\mathbf{v})}{\partial \pi} = \frac{\partial F_3(\mathbf{v})}{\partial \mathbf{v}} \times \frac{\partial \mathbf{v}}{\partial \mathbf{f}} \times \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \times \frac{\partial \mathbf{q}}{\partial \gamma} \times \frac{\partial \gamma}{\partial \pi} \tag{17}$$

where $\frac{\partial F_3(\mathbf{v})}{\partial \pi}$ is a $\lvert\mathscr{Z}\rvert \times \lvert\mathscr{W}\rceil$ dimension matrix of partial derivatives

$$\frac{\partial F_2(\gamma)}{\partial \pi} = \frac{\partial F_3(\gamma)}{\partial \gamma} \times \frac{\partial \gamma}{\partial \pi} \tag{18}$$

The computational graph offers the flexibility of adding new variables and parameters for possible joint estimation of a wide range of behavior and network performance models. For simplicity, the travel times are assumed to be observed in our study. It should be remarked that when link travel times are considered as variables and internally determined by a Bureau of Public Road (BPR) function or other time impedance functions based on link volume variables, an additional link travel time layer should be constructed in the BTCG. An iterative computation structure is needed to calculate travel time based on link volume and recalculate link volume based on updated path flow assignment results. This recursive relationship might be represented by acyclic computational graphs directly, for example, Recurrent Neural Network (RNN) (Goodfellow et al., chapter 10, 2016), and we will consider this extension in our future study. Furthermore, it is also important to develop a dynamic HFN/BTCG to estimate time-dependent demand in the future. The dynamic HFN/BTCG can be constructed based on time-space networks, where a set of trajectories in the networks can keep track of the movement of multiple vehicles in packets.

4.2. Illustrative example for HFN and BTCG

This section now uses an example to illustrate how to construct an HFN and its corresponding BTCG. Let us consider a traffic network with four nodes and three zones (Node 1, 2 and 4) presented as an HFN in Fig. 6(a). In Fig. 6(b), we express demand variables as vertexes in the HFN representation including

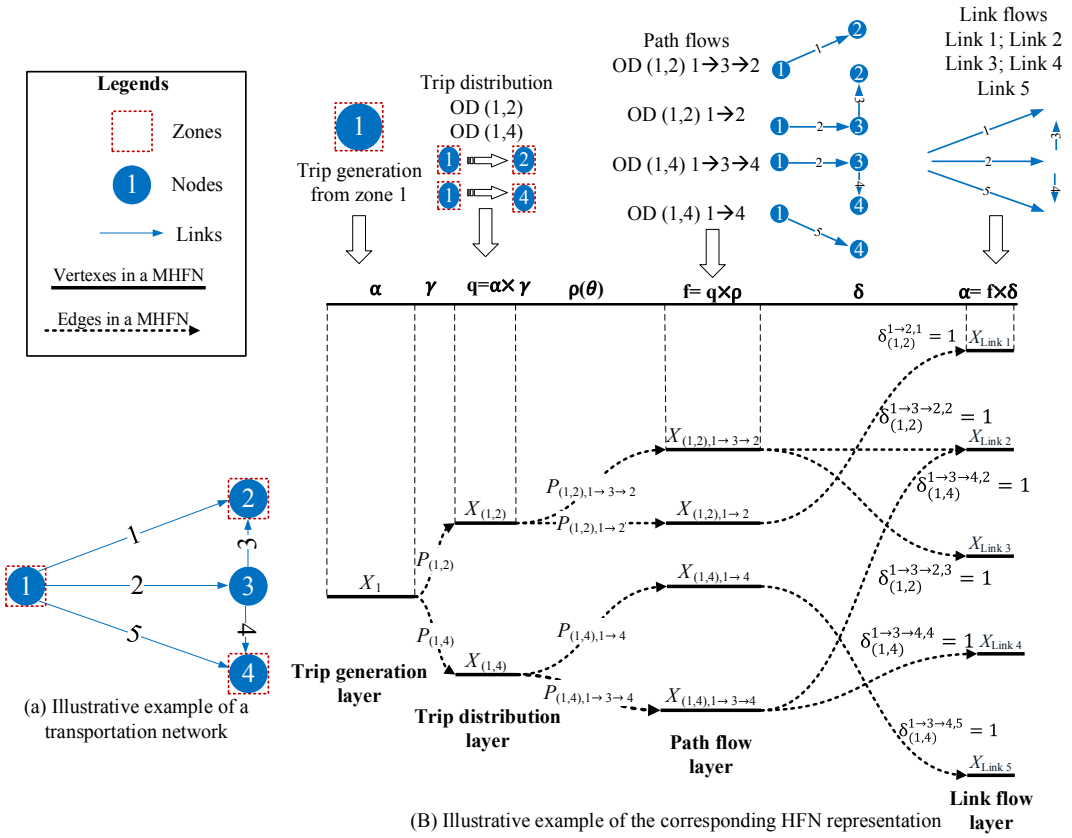


Fig. 6. An Illustrative example of the corresponding HFN representation.

(1) Trip production X_1
(2) OD flows $X_{(1,2)}$, $X_{(1,4)}$
(3) Path flows $X_{(1,2),1\to3\to2}$, $X_{(1,2),1\to2}$, $X_{(1,4),1\to3\to4}$, $X_{(1,2),1\to4}$
(4) Link flows X_{Link1}, X_{Link2}, X_{Link3}, X_{Link4}, X_{Link5}
(5) Estimated OD split rate $P_{1,(1,2)}$, $P_{1,(1,4)}$ (For simplicity, we use $P_{(1,2)}$ and $P_{(1,4)}$ instead of $P_{1,(1,2)}$ and $P_{1,(1,4)}$)
(6) Estimated route choice proportion $P_{(1,2),1\to3\to2}$, $P_{(1,2),1\to2}$, $P_{(1,4),1\to3\to4}$, $P_{(1,2),1\to4}$

Fig. 6(b) details a set of flow reservation constraints (1), (2) and (3):

$$X_1 \times (P_{(1,2)}, P_{(1,4)}) = (X_{(1,2)}, X_{(1,4)}) \tag{19}$$

$$(X_{(1,2)}, X_{(1,4)}) \times \begin{pmatrix} P_{(1,2),1\to3\to2} & P_{(1,2),1\to2} & 0 & 0 \\ 0 & 0 & P_{(1,4),1\to4} & P_{(1,4),1\to3\to4} \end{pmatrix} = (X_{(1,2),1\to3\to2}, X_{(1,2),1\to2}, X_{(1,4),1\to4}, X_{(1,4),1\to3\to4}) \tag{20}$$

$$(X_{(1,2),1\to3\to2}, X_{(1,2),1\to2}, X_{(1,4),1\to4}, X_{(1,4),1\to3\to4}) \times \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} = (X_{Link1}, X_{Link2}, X_{Link3}, X_{Link4}, X_{Link5}) \tag{21}$$

Fig. 7 shows the BTCG for the HFN representation shown in Fig. 6(b). Here, we explicitly draw every operation as a vertex in the computational graph. Figs. 8 and 9 illustrate the detailed computational graph of the normalization and logit layer for the OD pair (1, 4), and a similar method can be used to complete the BTCG in Fig. 7.

Figs. 7 and 8 display the three paths from vertex X_{Link2} to vertex $\pi_{(1,4)}$ on the computational graph. We could multiply partial derivatives of the edges of each path from vertex X_{Link2} to vertex $\pi_{(1,4)}$. That is

(1) Derivatives of Path ① $= \frac{\partial X_{Link2}}{\partial X_{(1,2),1\to3\to2}} \times \frac{\partial X_{(1,2),1\to3\to2}}{\partial X_{(1,2)}} \times \frac{\partial X_{(1,2)}}{\partial P_{(1,2)}} \times \frac{\partial P_{(1,2)}}{\pi_{(1,4)}} = P_{(1,2),1\to3\to2} \times X_1 \times \frac{-P_{(1,2)}}{\pi_{(1,2)}+\pi_{(1,4)}}$
(2) Derivatives of Path ② $= \frac{\partial X_{Link2}}{\partial X_{(1,4),1\to3\to4}} \times \frac{\partial X_{(1,4),1\to3\to4}}{\partial X_{(1,4)}} \times \frac{\partial X_{(1,4)}}{\partial P_{(1,4)}} \times \frac{-\pi_{(1,4)}}{[\pi_{(1,2)}+\pi_{(1,4)}]^2} = P_{(1,4),1\to3\to4} \times X_1 \times \frac{-P_{(1,2)}}{\pi_{(1,2)}+\pi_{(1,4)}}$
(3) Derivatives of Path ③ $= \frac{\partial X_{Link2}}{\partial X_{(1,4),1\to3\to4}} \times \frac{\partial X_{(1,4),1\to3\to4}}{\partial X_{(1,4)}} \times \frac{\partial X_{(1,4)}}{\partial P_{(1,4)}} \times \frac{1}{\pi_{(1,2)}+\pi_{(1,4)}} = P_{(1,4),1\to3\to4} \times X_1 \times \frac{1}{\pi_{(1,2)}+\pi_{(1,4)}}$

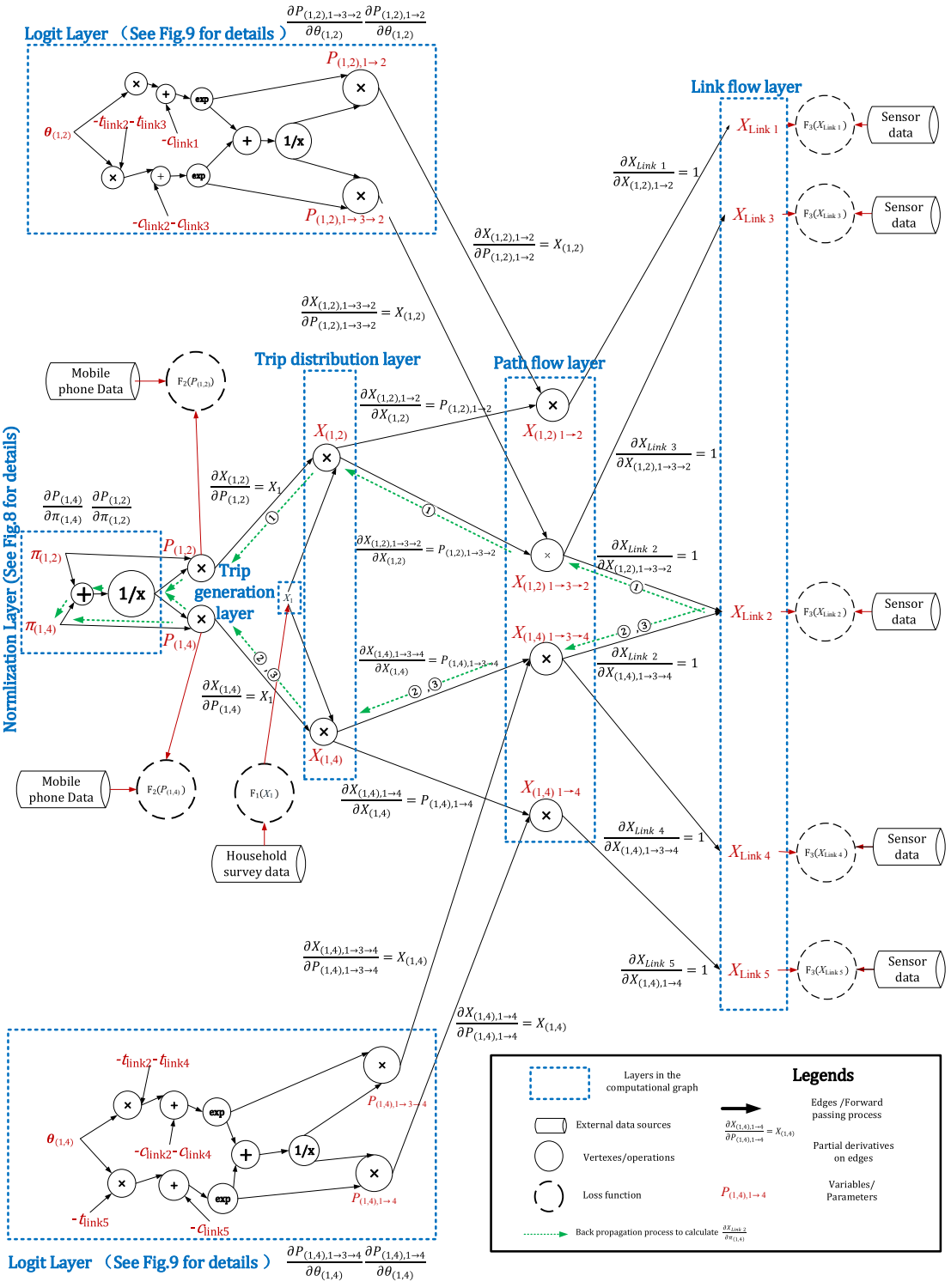


Fig. 7. BTCG corresponding to the HFN example in Fig. 6.

and then sum over the derivatives to calculate $\frac{\partial X_{Link2}}{\partial \pi_{(1,4)}}$.

$$\frac{\partial X_{Link2}}{\partial \pi_{(1,4)}} = \text{Derivatives of Path } ① + \text{Derivatives of Path } ② + \text{Derivatives of Path } ③$$

Fig. 9 shows three paths from vertex $P_{(1,4),1\to3\to4}$ to vertex $\theta_{(1,4)}$ in the logit layer of the computational graph. We could multiply the

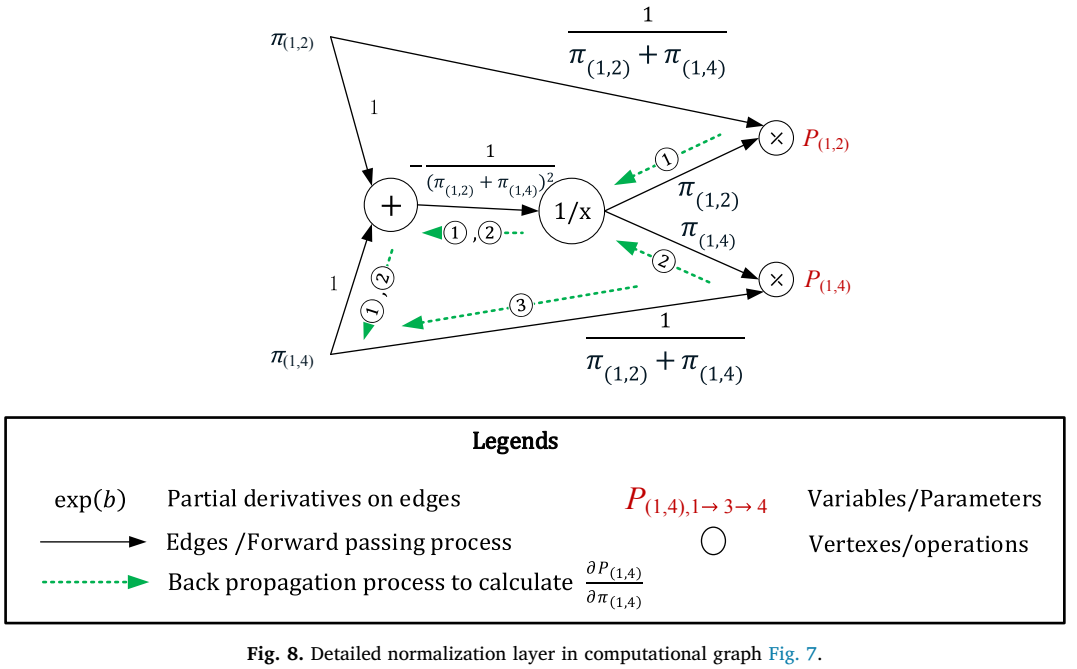


Fig. 8. Detailed normalization layer in computational graph Fig. 7.

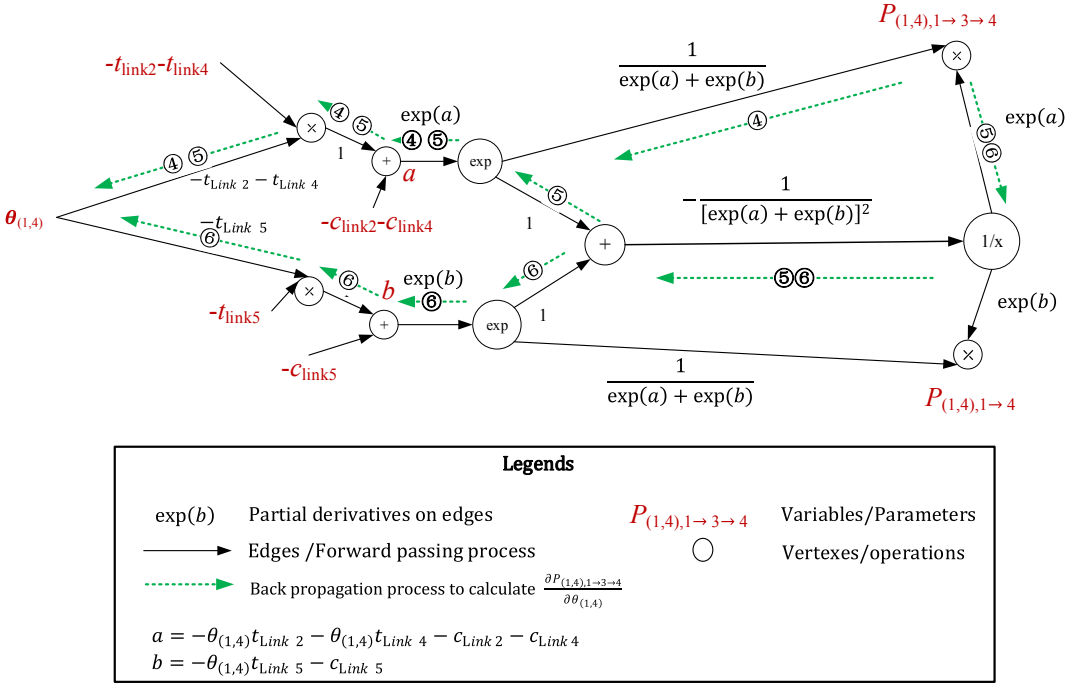


Fig. 9. Detailed logit layer of OD pair (1, 4) in computational graph Fig. 7.

partial derivatives of edges of all paths from vertex $P_{(1,4),1\to3\to4}$ to vertex $\Theta_{(1,4)}$ to calculate $\frac{\partial P_{(1,4),1\to3\to4}}{\partial \Theta_{(1,4)}}$.

That is

(1) Derivatives of Path ④$=\frac{1}{\exp(a)+\exp(b)}\times\exp(a)\times1\times(-t_{link2}-t_{link4})$

(2) Derivatives of Path ⑤$=\exp(a)\times\frac{-1}{[\exp(a)+\exp(b)]^2}\times1\times\exp(a)\times1\times(-t_{link2}-t_{link4})$

(3) Derivatives of Path ⑥$=\exp(a)\times\frac{-1}{[\exp(a)+\exp(b)]^2}\times1\times\exp(b)\times1\times(-t_{link5})$

$a=-\Theta_{(1,4)}t_{Link2}-\Theta_{(1,4)}t_{Link4}-c_{Link2}-c_{Link4}$

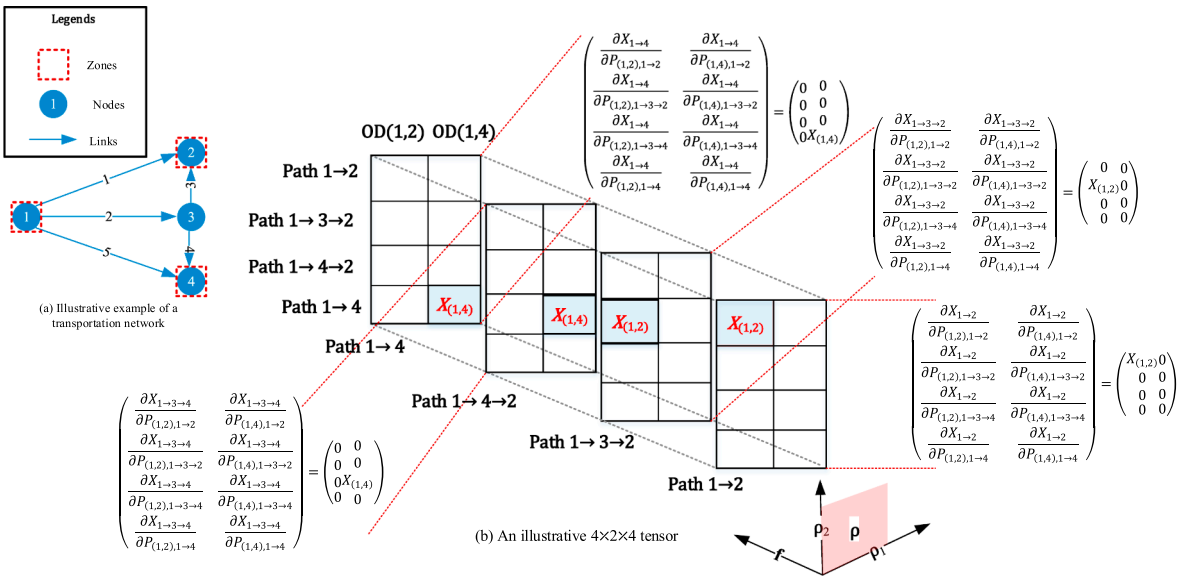


Fig. 10. Tensor expressing $\frac{\partial \mathbf{f}}{\partial \boldsymbol{\rho}}$ corresponding to the HFN representation in Fig. 4.

$$b = -\theta_{(1,4)} t_{Link5} - c_{Link5}$$

Then we sum over the derivatives of path ④, ⑤, and ⑥.

$$\frac{\partial P_{(1,4),1\to3\to4}}{\partial \theta_{(1,4)}} = \text{Derivatives of Path } ① + \text{Derivatives of Path } ② + \text{Derivatives of Path } ③$$

$$= P_{(1,4),1\to3\to4}\left[(t_{link2} + t_{link4})P_{(1,4),1\to3\to4} + t_{link5}P_{(1,4),1\to4} - t_{link2} - t_{link4}\right]$$

The result is consistent with the formula provided by Koppelman and Bhat (2006) in mode choice modeling (see Table 3).

When we calculate Eqs. (16) and (17), we should calculate tensors of high dimensions. Fig. 10 shows how to arrange numbers in a grid to form a $4 \times 2 \times 4$ dimension tensor to express $\frac{\partial \mathbf{f}}{\partial \boldsymbol{\rho}}$, corresponding to the transportation network in Fig. 6(a) and the HFN representation in Fig. 6(b), which have two OD pairs and four routes.

5. Solution procedure for implementing back-propagation

The solution algorithm for finding estimation results through BTCG includes the following three main steps iteratively:

(1) Forward passing

The forward passing step sequentially implements trip generation, trip distribution estimation, and a route-based traffic assignment which can be viewed as a process of the 4-step approach in the area of traffic planning.

(2) Backward propagation:

The back propagation step inversely implements a feedback control on the forward passing process. Different layers of first-order partial derivatives or "loss errors" are aggregated to calculate marginal gradients.

(3) Updating values of variables using stochastic gradient descent (SGD):

The estimation variables are updated using SGD algorithms using the marginal gradients.

It is widely known in deep learning that the BP algorithm, at its core, consists of repeatedly applying the chain rule of calculus through all possible paths in the computational graph. In our specific case of BTCG, we have established a layered network of composite functions, and the derivatives of those functions can be obtained using the chain rule. It also should be noted that to compute the gradients and perform the related learning tasks in a numerically reliable manner, we need to incorporate both the BP solution procedure and SGD algorithm effectively.

In a computational graph, any partial derivatives of the proposed loss functions with respect to any variables (including behavior parameters) can be calculated by finding all "computational paths" between the output computational vertex and the vertex of a parameter of interest. Obviously, a potential major difficulty is that there is an exponential number of directed computational paths.

To address this issue, a solution framework with the form of dynamic programming can be adapted to reuse the intermediate computational results in calculating the gradients across different layers. One can refer to chapter 6 in Goodfellow et al. (2016) to further understand how the BP algorithm is used within a DP framework to avoid the exponential explosion in repeated sub-expressions.

More precisely, we have a directed acyclic BTCG $G_c(\boldsymbol{V}_c, \boldsymbol{A}_c)$ corresponding to HFN $G(\boldsymbol{V}, \boldsymbol{A})$. $\boldsymbol{V}_c$ represents a set of all vertexes in the BTCG or the common term "states" in dynamic programming. Initial states are set on the vertexes viewed as an interface between **external data sources** and the **internal mathematical models**.

Consider $\boldsymbol{V}_c^B \subset \boldsymbol{V}_c$ as the set of vertexes expressing the initial states:

$$\frac{\partial \mathrm{F}}{\partial v} = c, \forall\, v \in \boldsymbol{V}_c^B \tag{22}$$

where F is the generalized loss function and c is a given constant expressing marginal changes. In the TDFE problem, X_o, P_{ow}, and X_a are all vertexes in $\boldsymbol{V}_c^B$. The initial states of gradients (as the boundary conditions) are shown below for different data sources:

$$\frac{\partial \mathrm{F}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{v})}{\partial X_o} = \lambda_1 \frac{\partial \mathrm{F}_1(\boldsymbol{\alpha})}{\partial X_o} \;\forall\, o \in \bar{\mathscr{D}}$$

$$\frac{\partial \mathrm{F}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{v})}{\partial P_{ow}} = \lambda_2 \frac{\partial \mathrm{F}_2(\boldsymbol{\gamma})}{\partial P_{ow}} \;\forall\, w \in \bar{\mathscr{W}}_o o \in \mathscr{D}$$

$$\frac{\partial \mathrm{F}(\boldsymbol{\alpha}, \boldsymbol{\gamma}, \mathbf{v})}{\partial X_a} = \lambda_3 \frac{\partial \mathrm{F}_3(\mathbf{v})}{\partial X_a} \;\forall\, a \in \bar{\mathscr{A}}$$

In this directed **acyclic** computational graph, we can use vertex reaching rules to compute their derivatives, that is, compute the gradient of a parent vertex only after all of the child vertexes have been "visited". Accordingly, the state transition function in a backward fashion is shown as follows:

$$\frac{\partial \mathrm{F}}{\partial v} = \sum_{v' \in \Delta^+(v)} \frac{\partial \mathrm{F}}{\partial v'} \frac{\partial v'}{\partial v} \;\forall\, v \in \boldsymbol{V}_c \tag{23}$$

Table 4
BP algorithm based on BTCG.

Step 1. Initialization

Step 2. **For** epoch n

Step 2.1. **For** all possible combinations of $m_1 = 1, 2, \cdots, M_1$, $m_2 = 1, 2, \cdots, M_2$ and $m_3 = 1, 2, \cdots, M_3$

Step 2.1.1 Perform forward passing step

Step 2.1.2. Backward propagate the deviations and partial gradients

For all o in $\mathscr{D}$

For all w in $\mathscr{W}_o$

For all r in $\mathscr{R}_w$

$\frac{\partial F_3(\mathbf{v})}{\partial X_r} = \sum_{a \in \mathscr{A}} \frac{\partial F_3(\mathbf{v})}{\partial X_a} \frac{\partial X_a}{\partial X_r} = \sum_{a \in \mathscr{A}} \delta_{ra} \times \frac{\partial F_3(\mathbf{v})}{\partial X_a}$

$\frac{\partial F_3(\mathbf{v})}{\partial P_{wr}} = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \frac{\partial X_r}{\partial P_{wr}}$

End (for all *r*)

$\frac{\partial F_3(\mathbf{v})}{\partial X_w} = \sum_{r \in \mathscr{R}_w} \frac{\partial F_3(\mathbf{v})}{\partial X_r} \frac{\partial X_r}{\partial X_w}$

$\frac{\partial F_3(\mathbf{v})}{\partial \theta_w} = \frac{\partial F_3(\mathbf{v})}{\partial X_r} \frac{\partial X_r}{\partial P_{wr}} \sum_{r \in \mathscr{R}_w} \frac{\partial P_{wr}}{\partial \theta_w}$

$\frac{\partial F_3(\mathbf{v})}{\partial P_{ow}} = \frac{\partial F_3(\mathbf{v})}{\partial X_w} \frac{\partial X_w}{\partial P_{ow}}$

$\frac{\partial F_2(\boldsymbol{\gamma})}{\partial \pi_{ow}} = \frac{\partial F_2(\boldsymbol{\gamma})}{\partial P_{ow}} \frac{\partial P_{ow}}{\partial \pi_{ow}}$

End (for all *w*)

For all w in $\mathscr{W}_o$

$\frac{\partial F_3(\mathbf{v})}{\partial \pi_{ow}} = \sum_{w' \in \mathscr{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_{w'}} \frac{\partial X_{w'}}{\partial P_{ow'}} \frac{\partial P_{ow'}}{\partial \pi_{ow}}$

End (for all *w*)

$\frac{\partial F_3(\mathbf{v})}{\partial X_o} = \sum_{w \in \mathscr{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times \frac{\partial X_w}{\partial X_o} = \sum_{w \in \mathscr{W}_o} \frac{\partial F_3(\mathbf{v})}{\partial X_w} \times P_{ow}$

End (for all *o*)

Step 2.1.3. Variable value updating

$X_o \leftarrow \max(0, X_o - \varphi_1 \lambda_3 \frac{\partial F_3(\mathbf{v})}{\partial X_o} - \varphi_1 \lambda_1 \frac{\partial F_1(\alpha)}{\partial X_o})\ \forall\, o \in \mathscr{D}$.

$\pi_{ow} \leftarrow \max(0, P_{ow} - \varphi_2 \lambda_3 \frac{\partial F_3(\mathbf{v})}{\partial \pi_{ow}} - \varphi_2 \lambda_2 \frac{\partial F_2(\boldsymbol{\gamma})}{\partial \pi_{ow}})$. Use Eq.(11) to update P_{ow}, $\forall\, w \in \mathscr{W}_o o \in \mathscr{D}$

$\theta_w \leftarrow \max(0, \theta_w - \varphi_3 \lambda_3 \frac{\partial F_3(\mathbf{v})}{\partial \theta_w})$. Use Eq.(12) to update P_{wr} $\forall\, r \in \mathscr{R}_w$, $w \in \mathscr{W}$

End (for all samples)

End until $n > N_{max}$. **(for all epochs)**

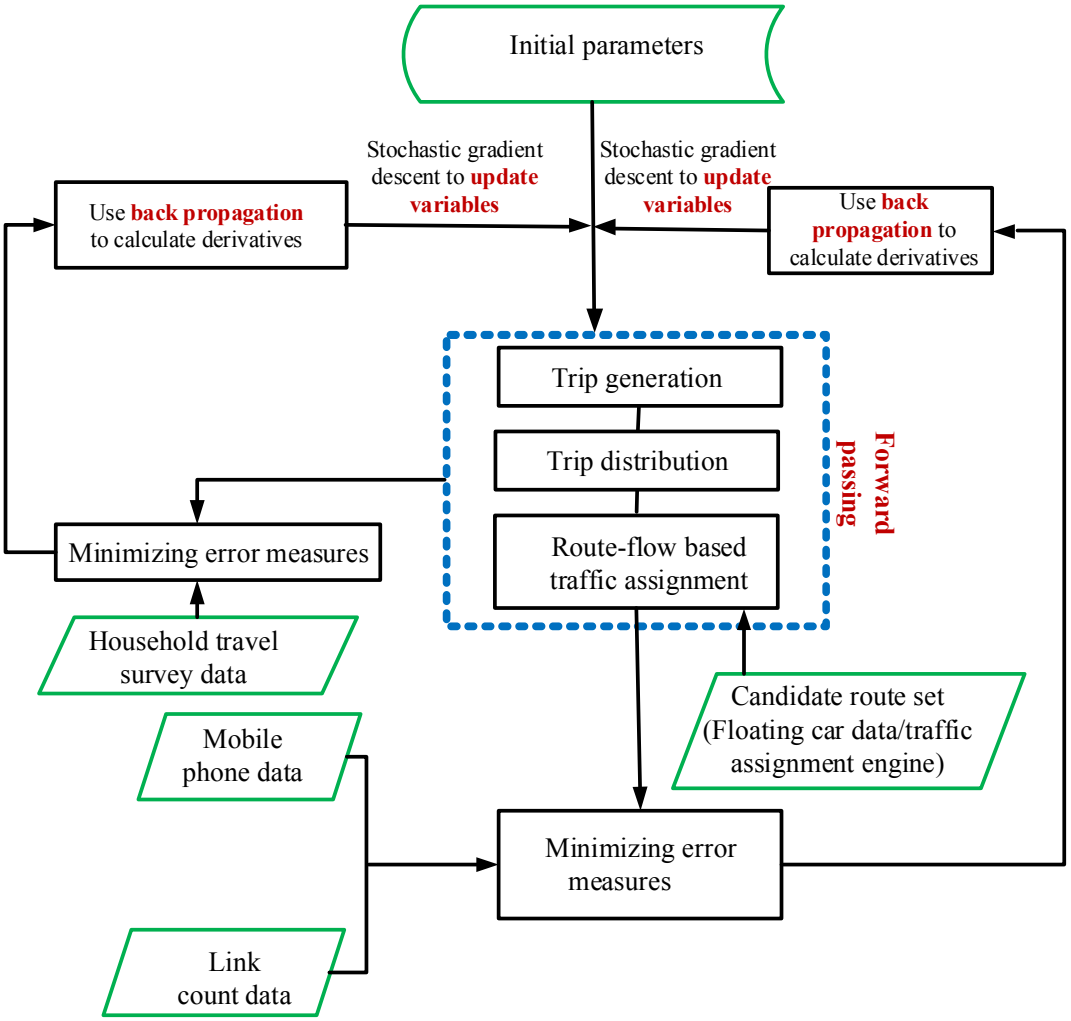


Fig. 11. Overall solution procedure proposed in this paper.

where v' is a successor of the vertexes v in the set $\Delta^+(v)$. The BP algorithm for BTCG is detailed in Table 4.

Fig. 11 summarizes the overall solution procedure of the proposed BP algorithm.

We would like to discuss two potential issues in using the above BP algorithm based on BTCG.

(1) Vanishing gradient issue in BTCG

The gradients of $\frac{\partial \mathbf{f}}{\partial \mathbf{q}}$ and $\frac{\partial \mathbf{q}}{\partial \alpha}$ are usually small in scale within a small range of [0, 1]. Regarding Eqs. (15) and (17), $\frac{\partial F_3(\mathbf{v})}{\partial \alpha}$ could be too insignificant to induce the change in the values of α, as illustrated in Fig. 12.

(2) Exploding gradient issue in BTCG

On the other hand, the gradients of $\frac{\partial \mathbf{f}}{\partial \rho}$ and $\frac{\partial \mathbf{q}}{\partial \gamma}$ could be very large approximations of the number of trips between an OD pair or from a zone. As shown in the example of Fig. 12, related to Eqs. (16) and (17), derivatives taking on larger values might lead to an unstable gradient descent process on parameters $\mathbf{p}$ and θ.

While we need to conduct future research to examine a more thorough way to address the above modeling issues, our implementation has used the following rules:

(i) As the magnitudes of the above three estimators are different in their own right, we should normalize the reference measures as the following:

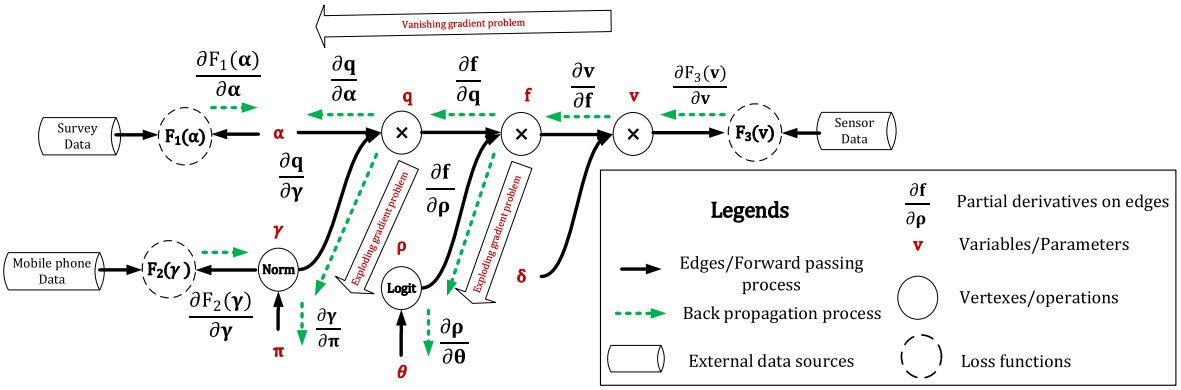


Fig. 12. Vanishing and exploding gradient problem in our proposed BTCG.

$$\mathrm{F}_1(\boldsymbol{\alpha}) = \frac{1}{2M_1}\sum_{m=1}^{M_1}\sum_{o\in\mathscr{I}}\left(\frac{\mathrm{X}_o}{\bar{\mathrm{X}}_o^m}-1\right)^2;$$

$$\mathrm{F}_2(\boldsymbol{\gamma}) = \frac{1}{2M_2}\sum_{m=1}^{M_2}\sum_{o\in\mathscr{I}}\sum_{w\in\mathscr{W}_o}\left(\frac{P_{ow}}{\bar{P}_{ow}^m}-1\right)^2;$$

$$\mathrm{F}_3(\mathbf{v}) = \frac{1}{2M_3}\sum_{m=1}^{M_3}\sum_{a\in\mathscr{A}}\left(\frac{\mathrm{X}_a}{\bar{\mathrm{X}}_a^m}-1\right)^2$$

As a result, $\frac{\partial \mathrm{F}(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})}{\partial X_o}$, $\frac{\partial \mathrm{F}(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})}{\partial P_{ow}}$ and $\frac{\partial \mathrm{F}(\boldsymbol{\alpha},\boldsymbol{\gamma},\mathbf{v})}{\partial X_a}$ have a similar order of magnitude.

(ii) Empirically, we could ensure step size $\varphi_3 \gg \varphi_2$ and $\varphi_3 \gg \varphi_1$ in **Step 2.1.3 Variable value updating** in the proposed BTCG BP algorithm (Table 4) to stabilize the solution search process.

6. Numerical experiments

In this section, we use two small-scale examples (case studies 1 and 2) to demonstrate the effectiveness of our proposed approaches. The BTCG models are implemented in MATLAB 2015. In case study 3, we implemented the BTCG model in a TensorFlow framework (Abadi et al., 2016) using Python 3.5, and a Beijing subnetwork was selected to examine the computational efficiency. The computational environment is a DELL PowerEdge T630 tower server with two Intel Xeon Quad CPUs, eight 16 GB of RAMs and 512 GB of SSD. In addition to TensorFlow, one can use other off-the-shelf software tools to construct a computation graph-based model, such as Theano. For educational and research purposes, one can find the Matlab and Python source code for small networks at https://github.com/xzhou99/BTCG.

6.1. Case study 1: multiple data sources vs. single data source

We would like to examine how different data sources within our proposed modeling framework can be used for both cross-validation and better approximating the "ground truth" trip demand. Consider a network with three zones, one freeway, and two arterials along two OD pairs: (A, B) and (A, C), as shown in Fig. 13. Now, we use some hypothetic "ground truth" values to evaluate generalization errors of our calculation results. That is, there are 1500 vehicle trips produced from zone A. In particular, 350 cars/hour (23%) are on the freeway, 550 cars/hour (37%) are on arterial 1, and 600 cars/hour (40%) are on arterial 2. The toll and travel time of the freeway are set to 2 dollars and 15 min, with the travel times of arterials 1 and 2 as 30 min and 60 min, respectively.

Further, we have the following data sources. The reference traffic measurements, $\bar{\mathrm{X}}_o$, $\bar{P}_{ow}$ and $\bar{\mathrm{X}}_a$ in Eqs. (4),(5),(6), are particularly assumed to have significant deviations from the above hypothetic "ground truth" values, to simulate the possible biases in real-world data sets. For example, the total production values from the survey data are 1400 trips, which are different from the ground truth of 1500 trips. As it is possible for cell phone users to take other modes of transportation such as ride-sharing, the bus, or subway, the total trips detected from cell phone records could be significantly larger than the ground truth value in terms of vehicles.

Household travel survey:

- Zone A has 4000 households, producing 0.35 vehicle trips per household.

Mobile phone data:

- There are 2000 cell phone records from A, with 60% of the trips going to B and the remaining 40% going to C.

Sensor Data:

- The total traffic counts on the freeway are 400 cars.

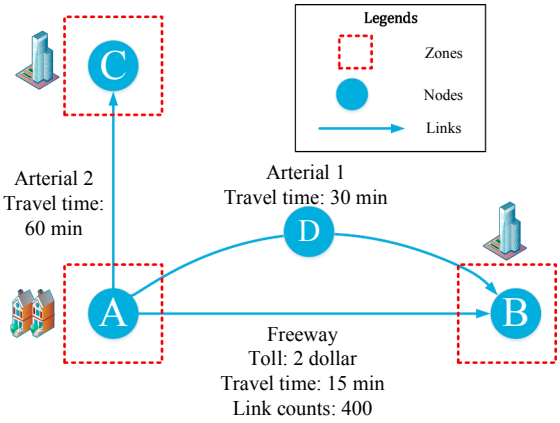


Fig. 13. An Illustrative example of a road network for the necessity of using multiple data sources.

Table 5 shows the estimated results based on the following six scenarios, each with different weights in their loss functions: (i) Mobile phone data ($\lambda_1 = 0,\ \lambda_2 = 1,\ \lambda_3 = 0$); (ii) Sensor data ($\lambda_1 = 0,\ \lambda_2 = 0,\ \lambda_3 = 1$); (iii) Household survey data ($\lambda_1 = 1,\ \lambda_2 = 0,\ \lambda_3 = 0$); (iv) Mobile phone data + household survey data ($\lambda_1 = 0.5,\ \lambda_2 = 0.5,\ \lambda_3 = 0$); (v) Sensor data + household survey data ($\lambda_1 = 0.5,\ \lambda_2 = 0,\ \lambda_3 = 0.5$); (vi) Mobile phone data + sensor data + household survey data ($\lambda_1 = 0.33,\ \lambda_2 = 0.33,\ \lambda_3 = 0.33$). The average GAP values decrease when more data sources are integrated in our model, from single-source scenarios (i) to (iii) with an average between 18% and 48%, to the multi-source scenarios (iv) to (vi) with an average between 1.5% and 22%. One can also examine the maximum gap performance with the best case of 4.9% in scenario (vi), to verify that our proposed approach can effectively fuse multiple data sources and serve as an excellent modeling framework to cross validate different layers of estimates, including production flow, OD flow, and link volume.

6.2. Case study 2: learning using different numbers of samples

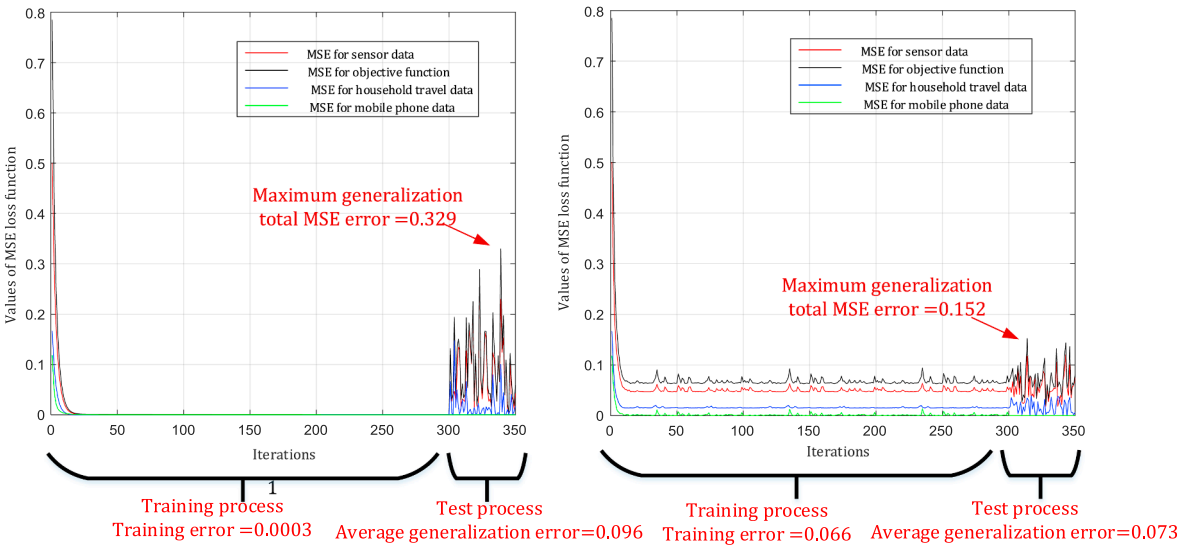
One challenge in machine learning is that we must perform well on new testing sets—not just on training sets. The representational capacity measures how well the algorithm performs on training sets, while the term of generalization measures the

Table 5
Comparison analysis between single data sources and multiple data sources.

	Ground truth	Ref. Values	(I) Phone data	GAP	(II) Sensor data	GAP	(III) Survey data	GAP
Production from A	1500	1400	2000	33%	2337	56%	1400	6.7%
OD A from B	900	60%	1200	33%	966	7%	700	22%
OD A from C	600	40%	800	33%	1371	128%	700	17%
Link flows A to B	350	400	267	24%	400	14%	156	55%
Link flows A to C	600		800	33%	1371	128%	700	22%
Link flows A to D	550		933	70%	566	3%	544	1%
Link flows D to B	550		933	70%	566	3%	544	1%
Avg. GAP				43%		48%		18%
Max GAP				70%		128%		55%
Min GAP				24%		3%		6.7%

	Ground truth	Ref. Values	(IV) Phone data + Survey data	GAP	(V) Phone data + Sensor data	GAP	GAP	(VI) Phone data + Sensor data + Survey data
Production from A	1500	1400	1400	6.7%	1415	5.7%	1502	0.1%
OD A from B	900	60%	840	6.7%	1060	18%	908	0.8%
OD A from C	600	40%	560	6.7%	355	41%	595	0.8%
Link flows A to B	350	400	187	47%	396	13%	367	4.9%
Link flows A to C	600		560	6.7%	355	41%	595	0.8%
Link flows A to D	550		653	18%	644	17%	541	1.6%
Link flows D to B	550		653	18%	644	17%	541	1.6%
Avg. GAP				15.7%		21.8%		1.5%
Max GAP				47%		41%		4.9%
Min GAP				6.7%		5.7%		0.1%

Note: GAP = $\%|estimatedvalue - Groundtruthvalue|/Groundtruthvalue$.



(a) Evolution of values of MSE loss functions during 3 epochs, when we use 1 samples for training and 50 samples for testing

(b) Evolution of values of MSE loss functions during 3 epochs, when we use 100 samples for training and 50 samples for testing

Fig. 14. Training errors and generalization errors using 1 sample and 100 samples, respectively.

ability of the algorithm to perform on new testing sets. Similar to other ANNs, the representational capacity of BTCG increases with more computational vertexes. When the capacity is higher than the requirement of the task, that is, there are insufficient data samples to learn, the training process may have an issue of poor generalization due to overfitting.

We implement two tests based on the network shown in Fig. 6 to demonstrate the need for larger training sets. Within a stochastic gradient descent (SGD) framework, 1 epoch means the sample is used once in the training process, and 300 epochs indicate the sample is used repeatedly, 300 times, for training. In both tests, we further use 50 randomly generated samples to test the generalization of the training process.

Test 1: Use a single sample with 300 epochs
Test 2: Use 100 randomly generated samples with 3 epochs.

As shown in Fig. 14, in general, when only a single sample is used repeatedly for training, the training curve can be pushed to a very low level, while the training errors might become larger under more samples (0.066). However, larger sample sets typically lead to tight generalization in new data. As shown in Fig. 14(a), the average generalization MSE error is about 0.096, and the maximum generalization MSE error is 0.329. In Fig. 14(b) with 100 random samples, the capacity of generalization increases, leading to an average generalization error of 0.073 and a significantly lower level of lower maximum generalization error.

6.3. Case study 3: Assessing computational efficiency in real-world test case study based on Beijing subnetwork

We now consider an experiment with real-world sample data based on a subnetwork in Beijing city to demonstrate the applicability of our proposed framework. Extracted from the regional planning data set with rich survey data, the subnetwork has 2502 nodes, 5397 links (4127 links with sensor data of 16 days), and 236 zones (see, Fig. 15). Furthermore, we aggregate floating car data to zones to produce the reference trip generation rates and OD split rates. In this experiment, we let the maximum iterations = 1000 and set the initial learning rate = 0.00001. Fig. 16 shows the convergence curve of the total MSE loss function with the visualization tool TensorBoard at the 487th training step. The total training CPU time till this step is 1 h 58 min.

Fig. 17 shows a snapshot of TensorFlow's visualization interface. The computational graph mainly consists of four modules: Node layer (i.e. Trip generation layer), OD layer, Path layer, and Link layer. In addition, the darker the module box, the greater the total resource consumption of the layer. The results illustrate that the calculation time and memory consumption of trip productions, OD pairs, paths, and links dramatically increases layer by layer This is consistent with the HFN representation which gradually allocates traffic flow from trip generation to the link flows.

7. Conclusion and future research

By recognizing theoretic insights in the field of deep learning and the multiple sources of information in emerging big data applications, this paper proposes the following major key modeling elements (as shown in Fig. 1):

(1) This research proposed a multi-layered **Hierarchical Flow Network** (HFN) to formulate the simultaneous estimation problem of

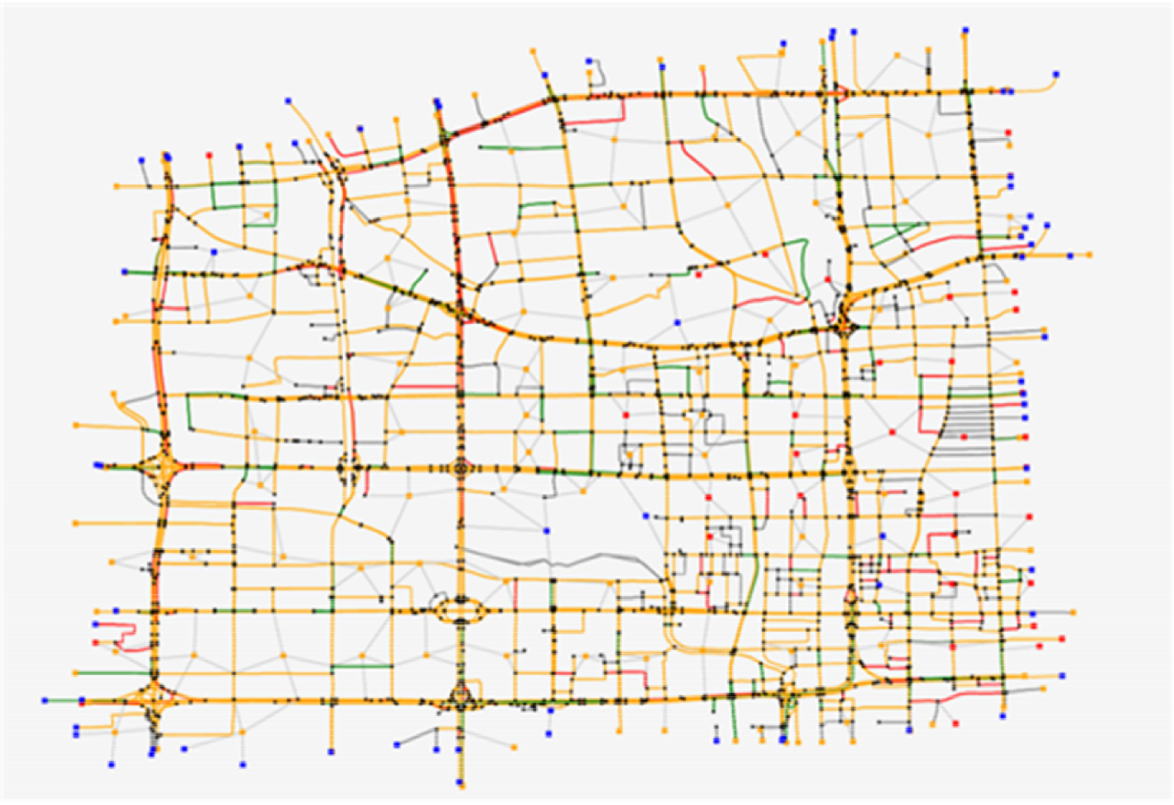


Fig. 15. The real-world subnetwork of Beijing city used in the case study.

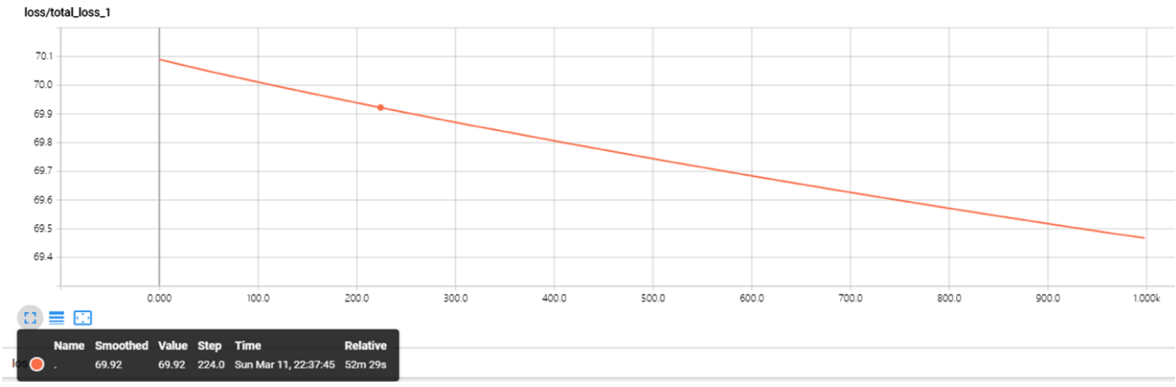


Fig. 16. Evolution of error value of MSE loss function $F(\alpha, \boldsymbol{\gamma}, \mathbf{v})$ in the 1000 epochs.

different levels of traffic demand and behavioral parameters. The proposed HFN is essentially a deep learning network with a special architecture. Each of the layers represents a different level of demand variables. Being different from traditional ANNs, the vertexes and connections of HFN integrate multiple modeling components and steps in traffic planning to ensure a more inherently consistent representation.

(2) We map different layers of HFN to different **big data sources**, including household travel surveys, mobile phone data, floating car data, and sensor data. This systematic linkage from the HFN to data sources enables planners to better conduct cross-validation and data fusion in emerging urban computing applications.

(3) The HFN is further utilized to reformulate the **TDFE model**, as a non-linear optimization program with composite functions of different demand variable layers. A training process of the HFN aims to optimize those layered variables to capture the different features of traffic demand and users' behaviors.

(4) By extending the HFN representation as a **Big data-driven Transportation Computational Graph** (BTCG), we propose a systematic method to evaluate the marginal effects between different variables and parameters. BTCG is inspired by the automated differentiation methods for numerical analysis which decomposes complex composite functions into elementary operations.

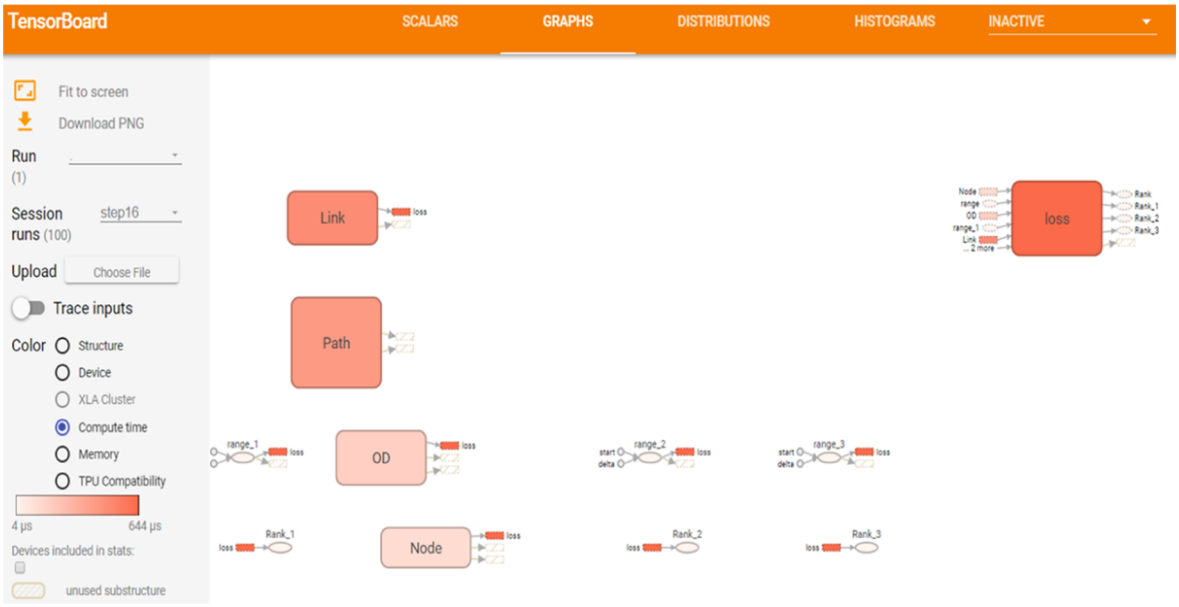


Fig. 17. Snapshot of a TensorBoard for the BTCG model and the training resources consumption.

(5) The first-order partial derivatives on the BTCG can be calculated efficiently in a sequence. The TDFE model can be solved using the **back propagation** (BP) algorithm.

In traffic planning applications, the demand-based models should be causal so a behavioral link can be established between the attributes of the transportation system and the decisions of the individual (Domencich and McFadden, 1975). The BTCG framework provides a systematic approach to express the interactions between different variables. Moreover, Boyce et al. (1994), Boyce (2007) and Pendyala et al. (2017) reviewed some integrated modeling efforts to view the 4-step method as a forward propagation process and back propagate the feedback from the assignment step to the trip distribution or activity-based models. In the proposed BTCG, we propagate the gradients to achieve a better approximation for the underlying traffic decision process. Some issues are extremely valuable to explore for future research.

(1) It has been well recognized in the traffic demand modeling community that the IIA assumption of the logit model could not capture the underlying correlation structure due to overlapping routes. More sophisticated models can be embedded in the HFN, e.g. link-nested logit models (Vovsha and Bekhor, 1998), and the network generalized extreme value (GEV) model (Bierlaire, 2002). Overall, the network GEV model provides an intuitive network representation of different discrete choice models including the multinomial logit, nested Logit, and link-nested logit models.
(2) Our data-driven multi-layer model can be related to a class of combined travel demand models (Yang and Chen, 2009), which have been developed to describe the hierarchical structure of the four-step travel decision processes. Starting from early studies by Lam and Huang (1992), it is developed based on the combined distribution-assignment problem and formulated as a constrained convex optimization program. It is worthwhile to compare and find further insightful linkage between the model-driven models with our data-driven model.
(3) It is interesting to examine the relationship between the proposed BTCG framework and the simulation-based calibration problem (Zhang et al., 2017), which used an analytical and differentiable meta-model to approximate the simulation-based objective function (Zhang and Osorio, 2017) and then apply gradient-based solution algorithms. In comparison, our proposed model in Sec.3 focuses more on how to iteratively use the BP solution algorithm to solve the underlying multi-layer nonlinear models represented through a computational graph.
(4) The HFN and BTCG can also extend to the field of public transit service and supply chain management as layered network representation.

Acknowledgments

This research project, especially the large-scale Beijing test network and various traffic data set, has been supported through Beijing Key Laboratory of Urban Traffic Operation Simulation and Decision Support and Beijing International Science and Technology Cooperation Base of Urban Transport. This research project is also supported by National Natural Science Foundation of China project no.71734004, tilted “Research on advanced theories for urban transportation governance”. The last author is partially funded by National Science Foundation–United States under NSF Grant No. CMMI 1538105 “Collaborative Research: Improving

Spatial Observability of Dynamic Traffic Systems through Active Mobile Sensor Networks and Crowdsourced Data" and NSF Grant No. CMMI 1663657. "Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks". The last author thanks Mr. Brian Gardner from Federal Highway Administration (FHWA) for his constructive comments. We also thank some kind comments from the collaborating team from Beihang University (China). The work presented in this paper remains the sole responsibility of the authors.

Appendix A. Mathematical properties of the simultaneous estimation problem

Here, we examine the non-convexity properties of the TDFE model with a simple network as shown in Fig. a1. The toll and travel time of the freeway is 5 dollars and 1 h, respectively. The toll and travel time of arterial 1 are 1 dollar and 5 h, respectively. The travel time of arterial 1 is 12 h. Sensor traffic count on the freeway is 400 cars, 380 on arterial 1, and 380 on arterial 2. Here we only consider the MSE loss function $F_3(\mathbf{v})$.

Then we have

$$F_3(\mathbf{v}) = \frac{1}{2}(X_{freeway}-400)^2 + \frac{1}{2}(X_{arterial1}-380)^2 + \frac{1}{2}(X_{arterial2}-350)^2$$

subject to

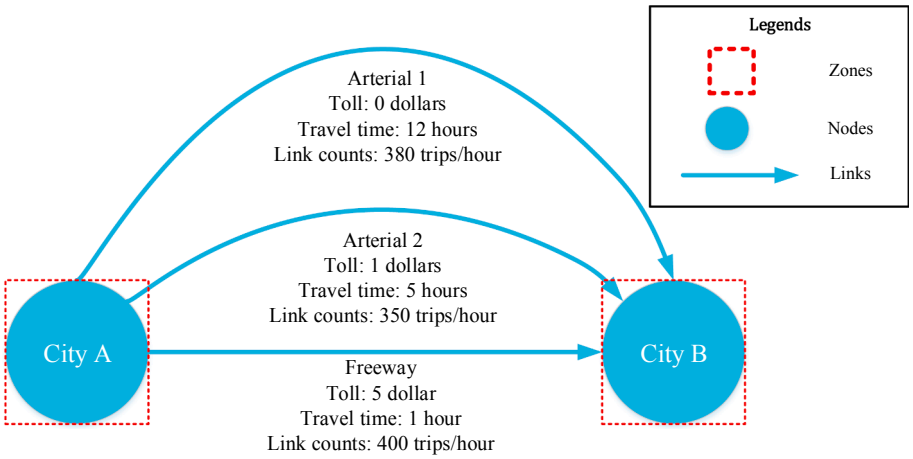
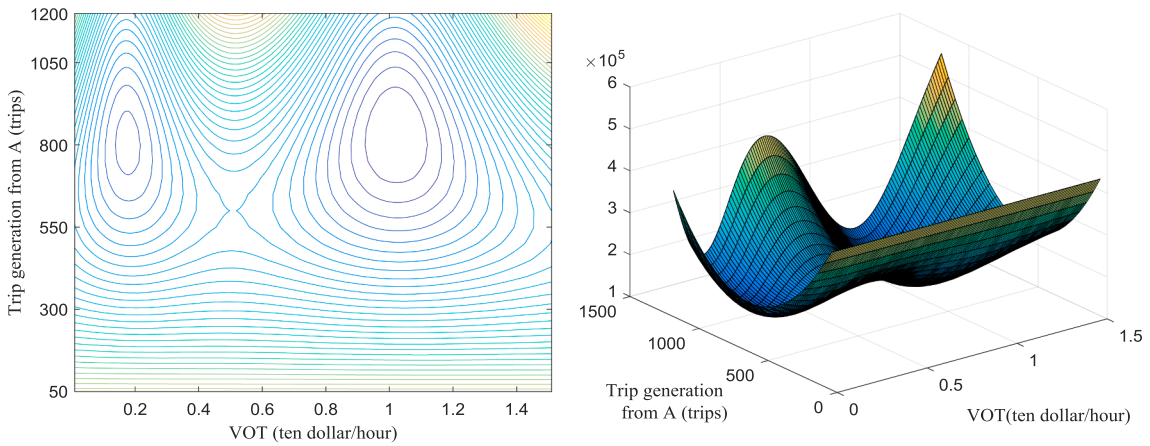


Fig. a1. An illustrative example of a road network for non-convexity of the TDFE problem.



(A) Contour line of the objective function (B) Values of the objective function versus trip generation and VOT

Fig. a2. Contour line and values of the illustrative MSE loss function $F_3(\mathbf{v})$ versus trip generation and VOT.

$$X_o \times \frac{\exp(-1 \times \theta - 5)}{\exp(-1 \times \theta - 5) + \exp(-5 \times \theta - 1) + \exp(-12 \times \theta)} = X_{freeway}$$

$$X_o \times \frac{\exp(-5 \times \theta - 1)}{\exp(-1 \times \theta - 5) + \exp(-5 \times \theta - 1) + \exp(-12 \times \theta)} = X_{arterial1}$$

$$X_o \times \frac{\exp(-12 \times \theta)}{\exp(-1 \times \theta - 5) + \exp(-5 \times \theta - 1) + \exp(-12 \times \theta)} = X_{arterial2}$$

It is easy to show that the Hessian matrix of $F_3(\mathbf{v})$ is not positive semidefinite and the MSE loss function is non-convex, which leads difficult to find the global optimal solution (Yang et al, 2001). Fig. a2(a) shows the contour lines of $F_3(\mathbf{v})$ and Fig. a2(b) shows the MSE loss function value $F_3(\mathbf{v})$ versus X_o and θ. Two local minimums are found in Fig. a2(b). On the other hand, by taking VOT θ as a constant, we then have a convex optimization problem.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., et al., 2016. TensorFlow: a system for large-scale machine learning. In: The Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), pp. 265–283.

Antoniou, C., Barceló, J., Breen, M., Bullejos, M., Casas, J., Cipriani, E., Ciuffo, B., Djukic, T., Hoogendoorn, S., Marzano, V., Montero, L., 2016. Towards a generic benchmarking platform for origin–destination flows estimation/updating algorithms: Design, demonstration and validation. Transport. Res. Part C: Emerg. Technol. 66, 79–98.

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. Part C: Emerg. Technol. 59, 129–146.

Boyce, D., 2007. Forecasting travel on congested urban transportation networks: review and prospects for network equilibrium models. Networks & Spatial Econ. 7 (2), 99–128. https://doi.org/10.1007/s11067-006-9009-0.

Boyce, D.E., Zhang, Y.F., Lupa, M.R., 1994. Introducing" feedback" into four-step travel forecasting procedure versus equilibrium solution of combined model. Transp. Res. Rec. 1443, 65.

Balakrishna, R., Koutsopoulos, H., 2008. Incorporating within-day transitions in simultaneous offline estimation of dynamic origin-destination flows without assignment matrices. Transport. Res. Record: J. Transport. Res. Board 2085, 31–38.

Bonnel, P., Hombourger, E., Olteanu-Raimond, A.M., Smoreda, Z., 2015. Passive mobile phone dataset to construct origin-destination matrix: potentials and limitations. Transp. Res. Procedia 11, 381–398.

Bauer, D., González, M.C., Toole, J.L., Ulm, M., 2012. Inferring land use from mobile phone activity. In: Proceedings of the Acm Sigkdd International Workshop on Urban Computing, pp. 1–8.

Bierlaire, M., 2002. The network GEV model. In: Swiss Transport Research Conference (No. TRANSP-OR-CONF-2006-050).

Brathwaite, T., Vij, A., Walker, J. L. 2017. Machine learning meets microeconomics: the case of decision trees and discrete choice. arXiv preprint arXiv:1711.04826.

Carrese, S., Cipriani, E., Mannini, L., Nigro, M., 2017. Dynamic demand estimation and prediction for traffic urban networks adopting new data sources. Transport. Res. Part C: Emerg. Technol. 81, 83–98.

Chen, C., Ma, J., Susilo, Y., Liu, Y., Wang, M., 2016. The promises of big data and small data for travel behavior (aka human mobility) analysis. Transport. Res. Part C: Emerg. Technol. 68, 285–299.

Chan, K.Y., Dillon, T.S., Singh, J., Chang, E., 2012. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg–Marquardt algorithm. IEEE Trans. Intell. Transp. Syst. 13 (2), 644–654. https://doi.org/10.1109/TITS.2011.2174051.

Cipriani, E., Florian, M., Mahut, M., Nigro, M., 2011. A gradient approximation approach for adjusting temporal origin–destination matrices. Transport. Res. Part C: Emerg. Technol. 19 (2), 270–282.

Dafermos, S., Nagurney, A., 1984. On some traffic equilibrium theory paradoxes. Transp. Res. Part B 18 (2), 101–110. https://doi.org/10.1016/0191-2615(84)90023-7.

Dia, H., 2001. An object-oriented neural network approach to short-term traffic forecasting. Eur. J. Oper. Res. 131 (2), 253–261. https://doi.org/10.1016/S0377-2217(00)00125-9.

Dougherty, M., 1995. A review of neural networks applied to transport. Transport. Res. Part C: Emerg. Technol. 3 (4), 247–260. https://doi.org/10.1016/0968-090X(95)00009-8.

Dixon, M.F., Polson, N.G., Sokolov, V.O., 2017. Deep learning for spatio-temporal modeling: dynamic traffic flows and high frequency trading. arXiv preprint arXiv:1705.09851.

Ermagun, A., Levinson, D., 2018. Spatiotemporal traffic forecasting: review and proposed directions. Trans. Rev. 1–29.

Domencich, T.A., Mcfadden, D.L., 1975. Urban travel demand: a behavioral analysis. Can. J. Econ./revue Canadienne D'economique 10 (4).

Frosst, N., Hinton, G.E., 2017. Distilling a neural network into a soft decision tree. CoRR, abs/1711.09784.

Frederix, R., 2012. Dynamic OD estimation in large-scale congested networks, PhD thesis, KU Leuven.

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 (-1), 19–25. https://doi.org/10.3141/2263-03.

Gonza'lez, M.C., Hidalgo, C.A., Baraba'si, A.L., 2008. Understanding individual human mobility patterns. Nature 453 (7196), 779–782.

Guo, J., Wen, H., Zhang, K., Wang, G., 2004. Research and construction of the demonstration project of Beijing comprehensive traffic information platform. J. Transport. Syst. Eng. Inform. Technol. 7–20 (3).

Ge, Q., Fukuda, D., 2016. Updating origin–destination matrices with aggregated data of GPS traces. Transport. Res. Part C: Emerg. Technol. 69, 291–312.

Ghali, M.O., Smith, M.J., 1995. A model for the dynamic system optimum traffic assignment problem. Transport. Res. Part B: Methodol. 29 (3), 155–170.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press, Cambridge.

Han, K., Yao, T., Friesz, T.L., 2012. Lagrangian-based hydrodynamic model: freeway traffic estimation. Transport. Res. Board Ann. Meet.

Griewank, A., 2012. Who Invented the Reverse Mode of Differentiation? Documenta Mathematica, Extra Volume ISMP, pp. 389–400.

Hao, P., Boriboonsomsin, K., Wu, G., et al., 2017. Modal activity-based stochastic model for estimating vehicle trajectories from sparse mobile sensor data. IEEE Trans. Intell. Transp. Syst. 18 (3), 701–711. https://doi.org/10.1109/TITS.2016.2584388.

Hu, X., Chiu, Y.C., Villalobos, J.A., Nava, E., 2017. A sequential decomposition framework and method for calibrating dynamic origin—destination demand in a congested network. IEEE Trans. Intell. Transp. Syst. 18 (10), 2790–2797.

Hinton, G.E., Sejnowski, T.J., 1986. Learning and relearning in Boltzmann machines. In: Rumelhart, D.E., McClelland, J.L. (Eds.), Parallel Distributed Processing. MIT Press, pp. 282–317. https://doi.org/10.1234/12345678.

Hensher, D.A., Button, K.J., 2007. Handbook of Transport Modelling. Emerald Group Publishing Limited.

Zhao, H., Yu, L., Guo, J., 2010. Estimation of time-varying OD demands incorporating FCD and RTMS data. J. Transport. Syst. Eng. Inform. Technol. 10 (1), 72–80.

Kumar, K., Parida, M., Katiyar, V.K., 2013. Short term traffic flow prediction for a non urban highway using artificial neural network. Procedia – Soc. Behav. Sci. 104,

755–764. https://doi.org/10.1016/j.sbspro.2013.11.170.
Koppelman, F.S., Bhat, C., 2006. A self instructing course in mode choice modeling: multinomial and nested logit models. US Department of Transportation, Federal Transit Administration, 31. https://doi.org/10.1002/stem.294.
Lam, W.H., Huang, H.J., 1992. A combined trip distribution and assignment model for multiple user classes. Transp. Res. 26 (4), 275–287.
Liu, S., Fricker, J.D., 1996a. Estimation of a trip table and the h parameter in a stochastic network. Transp. Res. Part A 30 (4), 287–305. https://doi.org/10.1016/0965-8564(95)00031-3.
Liu, S., Fricker, J.D., 1996b. Estimation of a trip table and the theta parameter in a stochastic network. Transp. Res. Part A 30 (4), 287–305. https://doi.org/10.1016/0965-8564(95)00031-3.
Lu, C.C., Zhou, X., Zhang, K., 2013. Dynamic origin–destination demand flow estimation under congested traffic conditions. Transport. Res. Part C 34 (34), 16–37. https://doi.org/10.1016/j.trc.2013.05.006.
Lu, L., Xu, Y., Antoniou, C., Ben-Akiva, M., 2015. An enhanced SPSA algorithm for the calibration of Dynamic Traffic Assignment models. Transport. Res. Part C: Emerg. Technol. 51, 149–166.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. 16 (2), 865–873. https://doi.org/10.1109/TITS.2014.2345663.
McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115–133. https://doi.org/10.1007/BF02478259.
Mudigonda, S., Ozbay, K., 2015. Using big data and efficient methods to capture stochasticity for calibration of macroscopic traffic simulation models. Symposium Celebrating 50 years of Traffic Flow Theory.
Montúfar, G.F., Pascanu, R., Cho, K., Bengio, Y., 2014. On the number of linear regions of deep neural networks. Adv. Neural Inform. Process. Syst. 2924–2932.
Nagurney, A., Yu, M., Masoumi, A.H., Nagurney, L.S., 2013. Networks Against Time: Supply Chain Analytics for Perishable Products. Springer Science & Business Media.
Nguyen, S., 1977. Estimation of an OD matrix from network data: A network equilibrium approach. In: Publication no. 60, Centŕe de recherćhe sur les transports, Universíe de Montŕeal, Montŕeal, Qúebec, Canada.
Nick Trefethen. Who invented the greatest numerical algorithms, 2005. www.comlab.ox.ac.uk/nick.trefethen.
Park, B., Messer, C., Urbanik, T., 1998. Short-term freeway traffic volume forecasting using radial basis function neural network. Transport. Res. Record: J. Transport. Res. Board 1651 (1), 39–47. https://doi.org/10.3141/1651-06.
Patriksson, M., 2015. The Traffic Assignment Problems: Models and Methods. Courier Dover Publications.
Pendyala, R., You, D., Garikapati, V., Konduri, K., Zhou, X., 2017. Paradigms for integrated modeling of activity-travel demand and network dynamics in an era of dynamic mobility management. In: The 96th Annual Meeting of the Transportation Research Board.
Qiu, T.Z., Lu, X.Y., Chow, A.H.F., Shladover, S.E., 2010. Estimation of freeway traffic density with loop detector and probe vehicle data. Transport. Res. Record J. Transport. Res. Board 2178 (-1), 21–29. https://doi.org/10.3141/2178-03.
Ramming, M.S., 2002. Network Knowledge and Route Choice. Institute of Technology, Massachusetts.
Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323 (6088), 533.
Sheffi, Y., 1985. Urban Transportation Networks - Equilibrium Analysis with Mathematical Programming Methods. Prentice-Hall.
Shi, Q., Abdel-Aty, M., 2015. Big Data applications in real-time traffic operation and safety monitoring and improvement on urban expressways. Transp. Res. Part C 58, 380–394. https://doi.org/10.1016/j.trc.2015.02.022.
Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: a comprehensive survey. Ann. Rev. Control 43, 128–151.
Small, K.A., Verhoef, E.T., Lindsey, R., 2007. The Economics of Urban Transportation. Routledge.
Tang, J., Song, Y., Miller, H.J., Zhou, X., 2016. Estimating the most likely space–time paths, dwell times and path uncertainties from vehicle trajectory data: A time geographic method. Transp. Res. Part C 66, 176–194. https://doi.org/10.1016/j.trc.2015.08.014.
Tavana, H., 2001. Internally-Consistent Estimation of Dynamic Network Origin–Destination Flows from Intelligent Transportation Systems Data Using Bi-level Optimization. The University of Texas at Austin, pp. 815–830.
Tobin, R.L., Friesz, T.L., 1988. Sensitivity analysis for equilibrium network flow. Transport. Sci. 22 (4), 242–250. https://doi.org/10.1287/trsc.22.4.242.
Toole, J.L., Colak, S., Sturt, B., Alexander, L.P., Evsukoff, A., González, M.C., 2015. The path most traveled: Travel demand estimation using big data resources. Transp. Res. Part C 58, 162–177. https://doi.org/10.1016/j.trc.2015.04.022.
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. Part C: Emerg. Technol. 55, 231–245.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and metaoptimized neural networks for short-term traffic flow prediction: A genetic approach. Transp. Res. Part C 13 (3), 211–234. https://doi.org/10.1016/j.trc.2005.04.007.
Vovsha, P., Bekhor, S., 1998. Link-nested logit model of route choice: overcoming route overlapping problem. Transport. Res. Record: J. Transport. Res. Board 1645, 133–142.
Willumsen, L.G., 1978, Estimation of an O-D Matrix from Traffic Counts – A Review. Working Paper. Institute of Transport Studies, University of Leeds, Leeds, UK.
Wright, S., Nocedal, J., 1999. Numerical optimization. Springer Sci. 35, 67–68.
Wu, X., Nie, L., Xu, M., Yan, F., 2018. A perishable food supply chain problem considering demand uncertainty and time deadline constraints: Modeling and application to a high-speed railway catering service. Transport. Res. Part E: Logist. Transport. Rev. 111, 186–209. https://doi.org/10.1016/j.tre.2018.01.002.
Wu, C., Thai, J., Yadlowsky, S., Pozdnoukhov, A., Bayen, A., 2015. Cellpath: Fusion of cellular and traffic sensor data for route flow estimation via convex optimization. Transport. Res. Part C: Emerg. Technol. 59, 111–128. https://doi.org/10.1016/j.trc.2015.05.004.
Yang, H., Meng, Q., Bell, M.G.H., 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. https://doi.org/10.1287/trsc.35.2.107.10133.
Yang, H., Yang, C., Gan, L., 2006. Models and algorithms for the screen line-based traffic-counting location problems. Comput. Oper. Res. 33 (3), 836–858. https://doi.org/10.1016/j.cor.2004.08.011.
Yang, Y., Fan, Y., Wets, R.J., 2018. Stochastic travel demand estimation: Improving network identifiability using multi-day observation sets. Transport. Res. Part B: Methodol. 107, 192–211.
Yang, C., Chen, A., 2009. Sensitivity analysis of the combined travel demand model with applications. Eur. J. Oper. Res. 198 (3), 909–921.
Yin, H., Wong, S.C., Xu, J., Wong, C.K., 2002. Urban traffic flow prediction using a fuzzy-neural, approach. Transport. Res. Part C Emerg. Technol. 10 (2), 85–98. https://doi.org/10.1016/S0968-090X(01)00004-3.
Yin, M., Sheehan, M., Feygin, S., Paiement, J.F., Pozdnoukhov, A., 2018. A generative model of urban activities from cellular data. IEEE Trans. Intell. Transp. Syst. 19 (6), 1682–1696. https://doi.org/10.1109/TITS.2017.2695438.
Zhao, Y., Kockelman, K.M., 2002. The propagation of uncertainty through travel demand models: an exploratory analysis. Ann. Reg. Sci. 36 (1), 145–163.
Zheng, W., Lee, D.H., Shi, Q., 2006. Short-term freeway traffic flow prediction: Bayesian combined neural network approach. J. Transp. Eng. 132 (2), 114–121. https://doi.org/10.1061/(ASCE)0733-947X(2006) 132:2(114).
Zhong, M., Sharma, S., Lingras, P., 2005. Short-term traffic prediction on different types of roads with genetically designed regression and time delay neural network models. J. Comput. Civil Eng. 19 (1), 94–103. https://doi.org/10.1061/(ASCE)0887-3801(2005) 19:1(94).
Zhou, X., Mahmassani, H.S., 2006. Dynamic origin–destination demand estimation using automatic vehicle identification data. IEEE Trans. Intell. Transp. Syst. 7 (1), 105–114. https://doi.org/10.1109/TITS.2006.869629.

Zhou, X., Qin, X., Mahmassani, H.S., 2003. Dynamic origin–destination demand estimation with multi-day link traffic counts for planning applications. Transport. Res. Record J. Transport. Res. Board 1831 (1), 30–38. https://doi.org/10.3141/1831-04.

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. Part B: Methodol. 41 (8), 823–840.

Zuylen, H.J.V., Willumsen, L.G., 1980. The most likely trip matrix estimated from traffic counts. Trans. Res. Part B 14 (3), 281–293. https://doi.org/10.1016/0191-2615(80)90008-9.

Zhang, C., Osorio, C., Flötteröd, G., 2017. Efficient calibration techniques for large-scale traffic simulators. Transport. Res. Part B: Methodol. 97, 214–239.

Zhang, C., Osorio, C., 2017. Efficient offline calibration of origin-destination (demand) for large-scale stochastic traffic models. Technical report, Massachusetts Institute of Technology. Under review. Available at: http://web.mit.edu/osorioc/www/papers/zhaOsoODcalib.pdf.