



Link-based traffic state estimation and prediction for arterial networks using license-plate recognition data



Xianyuan Zhan^{a,*}, Ruimin Li^b, Satish V. Ukkusuri^c

^a JD Intelligent City Research, JD Digits, Beijing 101111, China

^b Department of Civil Engineering, Tsinghua University, Beijing 100084, China

^c Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Keywords:

License plate recognition data
Arterial network
Traffic state estimation
Signal timing inference
Queue length approximation
Gaussian process
Hybrid dynamic Bayesian network

ABSTRACT

License-plate recognition (LPR) data are emerging data sources in urban transportation systems which contain rich information. Large-scale LPR systems have seen rapid development in many parts of the world. However, limited by privacy considerations, LPR data are seldom available to the research community, which lead to huge research gap in data-driven applications. In this study, we propose a complete solution using LPR data for link-based traffic state estimation and prediction for arterial networks. The proposed integrative data-driven framework provides the inference of both cycle maximum queue length states and average travel times of links using LPR data from a subset of intersections in an arterial network. The framework contains three novel data-driven sub-components that are highly customized based on the characteristics of LPR data, including: a traffic signal timing inference model to find signal timing information from the LPR timestamp sequences; a light-weighted queue length approximation model to estimate lane-based cycle maximum queue lengths and a network-wide traffic state inference model to perform network-level estimation and prediction using partially observed data. This study exploits and utilizes the unique features of LPR data and other similar vehicle re-identification data for urban network-wide link-based traffic state estimation and prediction. A six days' LPR dataset from a small road network in the city of Langfang in China and a more comprehensive link-level field experiment dataset are used to validate the model. Numerical results show that the framework provides good estimation and prediction accuracy. The proposed framework is efficient and calibration-free, which can be easily implemented in urban networks for various real-time traffic monitoring and control applications.

1. Introduction

License-plate recognition (LPR) data from high-definition (HD) video cameras are emerging data sources in urban transportation systems. Because of a wide range of applications, including automatic toll collection, law enforcement, emergency operation and traffic monitoring, the past few years has seen rapid deployment of large-scale LPR systems in many regions of the world (e.g. China, Malaysia, Thailand and the Middle East). For example, in 2010, Beijing has an LPR system with 374 HD LPR cameras; in 2017, this number has increased to 1,958 and covers the whole metropolitan area (Beijing Traffic Management Bureau, 2017). In typical LPR systems, HD video cameras installed at an intersection detect and take an image of each passing vehicle. Once the image is obtained from the camera, an image processing algorithm is used to enhance the image and further extract the alphanumerics of the license

* Corresponding author.

E-mail addresses: zhanxianyuan@jd.com (X. Zhan), lrmin@tsinghua.edu.cn (R. Li), sukkusur@purdue.edu (S.V. Ukkusuri).

plate. Finally, the recognized license-plate number will be saved into the database of the LPR system. Such LPR systems provide a rich source of information that enable recording the complete vehicle departing timestamp sequences at intersection level and allow for tracking the same vehicle at multiple intersections using recognized license-plate, from which the path travel time of the vehicle can be easily obtained. This provides a new opportunity for estimating and forecasting network-wide arterial traffic conditions. However, as LPR data contain license-plate information, local transportation agencies typically do not share the data to research communities due to privacy concerns. This has resulted in very limited research efforts in developing advanced real-time traffic estimation and forecasting models using LPR data. On the other hand, although large amounts of LPR data are collected every day in many cities, they are heavily underutilized and only considered for law enforcement purposes by local transportation authorities. Hence there is significant value in developing advanced data analytics that utilize the valuable information in LPR data.

In literature, there are few studies that used LPR data for traffic applications. [Bertini et al. \(2005\)](#) and [Yasin et al. \(2010\)](#) are among the first researchers to explore using LPR data for link-level travel time estimation. These works are relatively straightforward, as link travel times are simply the timestamp differences from vehicles with matched license-plate numbers between upstream and downstream intersections. Several recent studies developed more advanced models for various traffic estimation problems using LPR data, such as estimate real-time lane-based queue lengths ([Zhan et al., 2015](#)) and estimate the speed profile of vehicles passing a road segment ([Mo et al., 2017](#)). These works are again restricted to link-level and need to collect field data for model calibration. [Chen et al. \(2017\)](#) analyzed city-level travel behavior by clustering a large-scale LPR dataset from Shenzhen, China. However, as the data were collected from highway and parking lots, it was only used for conducting traffic pattern analysis.

Traffic state estimation and prediction for arterial networks are central to many applications in traffic operations and management. Two widely used urban link-based traffic state measures are short-term link travel times and queue lengths. Traditional approaches for short-term urban link travel time estimation and prediction have largely rely on data from various road-based sensors, such as loop detectors ([Coifman and Cassidy, 2002](#); [Zhang and Rice, 2003](#); [Wu et al., 2004](#)), automated vehicle identification (AVI) sensors ([Park and Rilett, 1998](#); [Li and Rose, 2011](#); [Sherali et al., 2006](#)) and Remote Traffic Microwave Sensors (RTMS) ([Yeon et al., 2008](#)). Similar data sources were also used in estimating queue lengths at signalized intersections, such as real-time queue length estimation using loop detector data ([Sharma et al., 2007](#); [Vigos et al., 2008](#); [Skabardonis and Geroliminis, 2008](#)) and event-based signal and vehicle detection data ([Liu et al., 2009](#)). These approaches require installing corresponding fixed sensors, which limit their use only to major road segments or small arterial networks. With the recent advances of pervasive computing technologies, data from mobile sensors such as GPS probe vehicles and mobile phones have become another promising data source for urban traffic monitoring. For example, many recent studies focused on estimating link travel times using detailed GPS trajectory information from probe vehicle data ([Herrera et al., 2010](#); [Herring et al., 2010](#); [Hofleitner et al., 2012](#); [Xumei et al., 2012](#); [Zheng and VanZuylen, 2013](#); [Wang et al., 2014](#); [Tseng et al., 2018](#)). A few studies also explored using sparser trajectories or path-level GPS information to estimate link travel times ([Hunter et al., 2009](#); [Zhan et al., 2013](#); [Zhan et al., 2016](#)). Moreover, [Ban et al. \(2011\)](#) examined the possibility of using GPS probe vehicle data to estimate real-time queue lengths at signalized intersections. The greatest strength of data from mobile sensors over traditional road-based sensor data is that it enables continuous monitoring urban traffic network with large coverage areas without the need of installing any fixed sensors.

Compared with the aforementioned conventional and emerging traffic data sources, LPR data offer several advantages in network-wide link-based traffic state estimation and prediction. First, LPR data do not suffer from the issue of low penetration rate that often occurs in applications using GPS probe vehicle data ([Ban et al., 2011](#); [Hofleitner et al., 2012](#); [Zhan et al., 2017](#)). LPR data records almost all the passing vehicles at an intersection, whereas the GPS probe vehicle data only covers a small fraction of vehicles in the traffic. Despite the previous advantage, the greatest strength of LPR data lies in the rich and unique information provided. From LPR data, we can simultaneously obtain three levels of information on traffic condition. At the intersection level, the LPR data record the departing timestamps from each lane at the intersection. In this case, LPR cameras work similar to road-based sensors (e.g. loop detector). Similar to studies that use road-based sensor data, the traffic dynamics at the end of the link (e.g. queuing) can be estimated by performing shockwave analysis ([Skabardonis and Geroliminis, 2008](#); [Liu et al., 2009](#)). At the link level, if both the upstream and downstream intersections are equipped with LPR cameras, the actual link travel times can be easily obtained through matching the identical license-plate numbers. Although there exist other vehicle re-identification techniques for link travel times estimation, such as using dual loop detectors ([Coifman and Cassidy, 2002](#); [Coifman and Krishnamurthy, 2007](#)) and vehicle magnetic signatures ([Oh et al., 2007](#); [Kwong et al., 2009](#); [Jeng et al., 2010](#)), their vehicle re-identification accuracy is much lower compared with LPR data-based approaches, thus often lead to inferior travel time estimation results. This is because that the re-identification of a vehicle using LPR data is based on unique license plate information, while the other techniques rely on less precise vehicle identifiers, such as vehicle lengths (dual loop detectors approach) and magnetic signals (vehicle signature approach). Finally, at the path level, the LPR data can be used to track the same vehicle at multiple intersections in a network and reveal path travel time information. In that sense, LPR data can also be treated as a path-based data, which is particularly useful in performing network-wide traffic condition estimation when the LPR cameras are only sparsely located in the road network ([Zhan et al., 2013](#); [Zhan et al., 2016](#)). Due to the high installation and maintenance cost, most cities usually only install LPR cameras at intersections connecting important roads. This often leads to a partially monitored network, which poses great challenges for network-level applications using the LPR data. To perform accurate network-wide traffic state estimation and prediction, all the three levels of information in the LPR data need to be fully utilized.

This study proposes an integrative statistical modeling framework to estimate the real-time traffic states of cycle maximum queue length (maximum number of queuing vehicles in a signal cycle) as well as the average travel time for each link in the urban arterial

networks using LPR data from a subset of intersections. The framework contains three components: a signal timing inference model that finds the signal timing information from the LPR timestamp sequences; a light-weighted queue length approximation model that efficiently estimate the lane-based cycle maximum queue lengths; and finally, a network-wide traffic state inference component modeled as a dynamic linear-Gaussian (DLG) model to perform inferences on average cycle-maximum queue lengths and average link travel times for unobserved links. This paper is the first study in the literature that exploits and utilizes the unique features of LPR data for urban network-wide traffic state estimation and prediction, which enables full utilization of the existing LPR systems in urban areas. The proposed statistical modeling framework combines both well-established traffic flow theory and highly customized statistical machine learning techniques to provide robust inferences. We obtain two datasets to fully test and validate the proposed framework. The first dataset is a six-day network-level LPR datasets from 11 intersections from the city of Langfang, China. We also used a smaller but more comprehensive link-level dataset obtained in a field experiment to further validate the signal timing inference model and the queue length approximation model. The numerical results show that the framework achieves good estimation and prediction accuracies.

It should be noted that although the proposed integrative framework focuses on the LPR data, it is also widely applicable to other types of vehicle re-identification data that share similar characteristics with LPR data. One typical example is the data from urban vehicle tracking systems using radio frequency identification (RFID) technology. Similar to the LPR systems, the RFID vehicle tracking systems place RFID tags in vehicles and install sensors on road segments to record the “virtual license-plate” (RFID tags) as well as timestamps of each passing vehicle at monitored locations. The RFID data contain the same types of information as LPR data and demonstrate huge potential for city-wide vehicle monitoring. Currently, a metropolitan-scale RFID vehicle tracking system is already under deployment throughout the city of Shanghai, China (Zhu et al., 2009). It is expected that in the near future, RFID tags will be installed in every vehicle in China (He and Das, 2015). The proposed framework is efficient and calibration-free, which can be easily implemented in urban arterial networks with reasonable LPR camera or RFID sensor coverage for real-time traffic monitoring and control applications.

The paper is organized as follows: the next section presents the overall framework; Section 3–5 describe the detailed methodologies for the three core components of the framework, namely, traffic signal timing inference, queue length approximation and network-wide traffic state inference; Section 6 evaluates the performance of our methods; the final section concludes the paper.

2. Overview

2.1. Problem definition

Given an urban arterial network and LPR data $D = \{(I, ID^n, t_h^d, dir^n, ln^n), n = 1, \dots, N\}$ observed from a subset of intersections equipped with LPR cameras, with the information of vehicle n with license-plate number ID^n departing from intersection I at time t_h^d using lane ln^n and taking the movement direction dir^n , we want to infer the average cycle maximum queue lengths across different lanes of the link ($\mathbf{q}^T = (q_1^T, q_2^T, \dots, q_m^T)^T$, m is the total number of links), as well as the average travel time ($\mathbf{y}^T = (y_1^T, y_2^T, \dots, y_m^T)^T$) for each link in the network during time slice T . A specific direction of an intersection is called *observed* if it is monitored by an LPR camera and the data is obtainable; otherwise, it is referred to as *unobserved*. Due to the existence of unobserved intersections, the input data of the network-wide link-based traffic state estimation and prediction problem are the vehicle departing timestamp sequences of each lane from the observed intersections ($\mathbf{x}^T = (x_1^T, x_2^T, \dots)^T$), average link travel times for links with both observed upstream and downstream intersections and average path travel times ($\mathbf{z}^T = \{z_p^T\}$) between two observed intersections from license-plate matched vehicles, where z_p^T denotes the average path travel time for path p in time slice T . In this problem, we only consider the path travel times involving at most two links, which makes the path of a vehicle easily resolvable based on the network topology. This avoids extra modeling complexity caused by solving the underlying path inference problem.

2.2. Overall framework

Before introducing the details of the proposed framework, we first classify all the links in the network into three types:

Type 1: both upstream and downstream intersections are observed. For this type of links, the actual link travel times can be directly obtained from the differences of departing timestamps between upstream and downstream intersections of license-plate matched vehicles. Furthermore, based on the departure information from the downstream intersection, the cycle maximum queue length can be approximated using shockwave analysis and a Gaussian process model, which will be described in the following section.

Type 2: only downstream intersections are observed. For this type of links, the average link travel times are not observed. However, similar to Type 1 links, we can still approximate the cycle maximum queue lengths using data solely from the downstream intersection.

Type 3: downstream intersections are not observed. For this type of links, both the queue lengths and average link travel times are not observed and need to be inferred. To incorporate more information, the travel times from a set of observed paths (the start and the end intersections of the path are observed) that contain the link are used as input variables.

Fig. 1 presents the overall framework of this paper, which contains three main components: traffic signal timing inference, cycle maximum queue length approximation and a network-wide traffic state inference model based on the dynamic linear-Gaussian

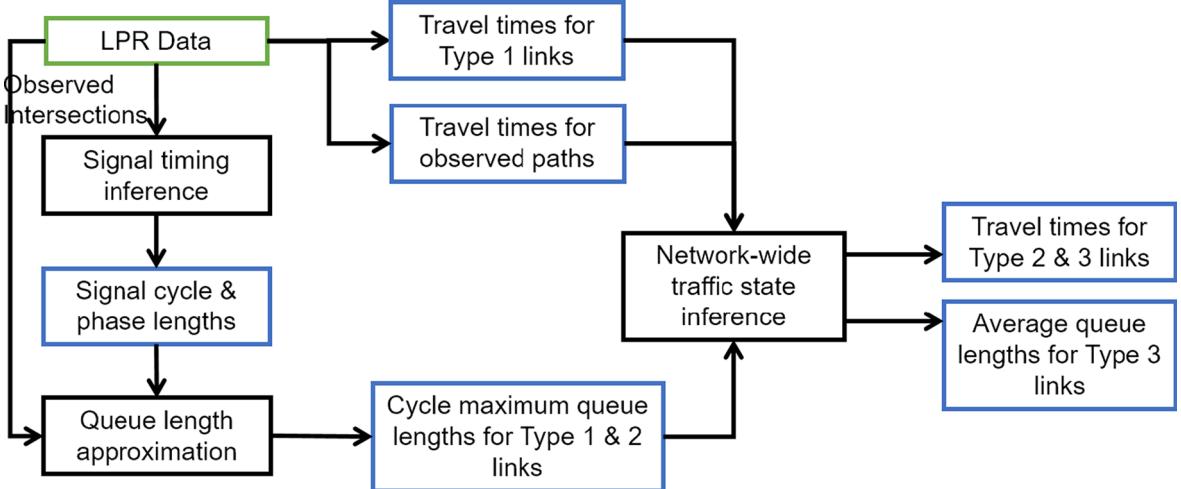


Fig. 1. Overall framework.

model. The network-wide average cycle maximum queue lengths and average link travel times of each time slice are inferred in following steps. The actual travel times for Type 1 links as well as a set of predefined paths are extracted from LPR data by matching the license plate numbers at the upstream and downstream intersections. For other unknown traffic state measures, we first use a signal timing inference model constructed as a weighted soft margin maximization problem to obtain signal cycle and phase lengths for observed intersections. Such information is required for the queue length approximation model. We propose a highly efficient Gaussian process model to approximate the lane-based cycle maximum queue lengths at the downstream intersections of both Type 1 and 2 links. The approximated cycle maximum queue lengths, average link travel times of Type 1 links and average travel times of observed paths are all used as observed data for the final network-wide traffic state inference model. In the last step, the average cycle maximum queue lengths for Type 3 links as well as the average travel times for Type 2 and Type 3 links are inferred using a partially observed DLG model.

3. Traffic signal timing inference

We first introduce the signal timing inference component of the framework, which infers the lane-level signal timing information (length for signal cycle, green and red phases) when real-time signal timing information is obtainable. Many real-world LPR systems are not integrated with the traffic signal control systems. Hence extracting real-time signal timing information for LPR data-based applications sometimes can be problematic, especially for signalized road networks implementing adaptive signal control strategies.

Signal timing inference has been investigated in several studies. Early works estimate signal timing using sample travel time data (Hao et al., 2012). More recent studies (Fayazi et al., 2014; Axer and Friedrich, 2017; Du et al., 2019) used probe vehicle trajectories for signal timing estimation. The general methodology in these studies can be summarized as three steps: 1) perform map-matching on trajectory points; 2) infer intersection stop line passing time; and 3) estimating signal phase lengths using data filters or perform

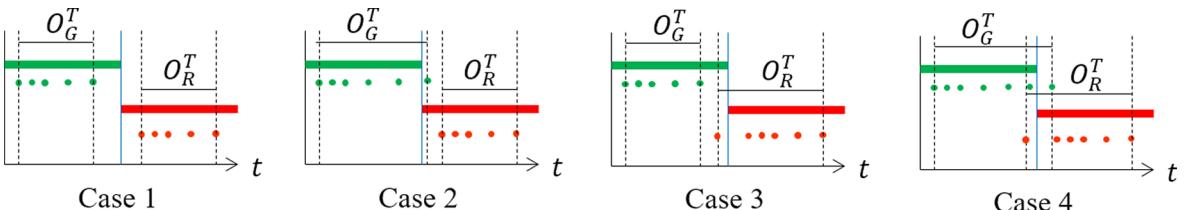


Fig. 2. Real-world cases in signal timing inference. The solid green and red lines represent the actual lengths of green and red phases. The green dots are the departing timestamps of vehicles leaving the lane (within the green phase), and the red dots are the departing timestamps from the orthogonal direction (red phase of the lane). O_G^T and O_R^T are empirically observed longest green and red phase intervals in LPR data, which are very different from the actual green and red phase. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

statistical analysis on collected time intervals. As LPR data directly record the passing timestamps of vehicles at the intersection, this removes the need to perform map-matching or stop line passing time inference that are necessary in trajectory-based signal timing estimation methods. Moreover, LPR system records data of every passing vehicle at a intersection, which provides much richer information compared with sparse trajectories of probe vehicles. Based on these special features of LPR data, we develop a new approach that extends the well-known support vector machine (SVM) to infer signal timing information for each lane. SVM is a widely used machine learning method, that has been widely used transportation domain, such as transportation mode recognition (Jahangiri and Rakha, 2015; Shafique and Hato, 2015), crash severity analysis (Ni et al., 2016; Ahmadi et al., 2018), and traffic volume or travel time forecasting (Zhang and Xie, 2007; Tseng et al., 2018; Xumei et al., 2012). Unlike using SVM for classification or regression tasks in previous works, we focus on solving the soft margin maximization problem (the optimization problem solved in SVM) and extract the learned decision boundary parameter to separate different signal phases.

We start by presenting four typical cases exist in LPR data (Fig. 2). For simplicity, we combine the yellow phase into the green phase. In normal situations, we observe that all the departing timestamps of vehicles lie strictly within the green and red phases (Case 1). However, it is also observed that the departing timestamps may appear in the wrong phase (e.g. green dots in red phase and red dots in green phase, illustrated in Case 2, 3 and 4). There are two reasons for these violations. First, such violations may correspond to vehicles that running the red light. Second, it can result from inaccurate identification of departure vehicle. In typical LPR systems, a virtual detection zone is set before the stop line of each lane, which has the size equal or smaller to the area occupied by a vehicle. The LPR camera will recognize a vehicle only when a vehicle passes the virtual detection zone. In certain conditions, when a vehicle is still waiting for the green light but moves a small distance inside the virtual detection zone, an LPR record with problematic departing timestamps will be generated. These violations result in overlapping data around the actual phase change time and forbid the use of naïve data partitioning methods for phase length estimation.

To accurately infer the signal phase lengths, we cast the signal timing inference problem as a weighted soft margin maximization (WSMM) problem. Denote x_1, x_2, \dots as the observed vehicle departing timestamps in a signal cycle (x_i are normalized such that the earliest observation has $x = 0$); t_1, t_2, \dots are the associated labels, which is -1 if it belongs to the first phase or $+1$ for second phase; and b is the phase change time, which serves as the decision boundary to partition the data. Let $g(x_i) = v(t_i)w(x_i - b)$ be a decision function that satisfy following conditions:

- $g(b) = 0$.
- If $g(x_i) \leq -1$, then the observation x_i has label $t_i = -1$.
- If $g(x_i) \geq 1$, then the observation x_i has label $t_i = 1$.
- If $-1 \leq g(x_i) \leq 1$, then the observation x_i can have label of either -1 or 1 .

The value $|v(t_i)(x_i - b)|$ measures the weighted distance between each observation and the decision boundary b . w in the expression of $g(x_i)$ is a parameter to properly scale the weighted distance. The weighted distance $|v(t)(x - b)|$ at some location x scaled by w such that $g(x) = -1$ or $g(x) = 1$ is called weighted soft margins (Wu and Srihari, 2004) (illustrated in Fig. 3). $v(t_i)$ is a weight mapping from the observation label ($t_i = -1/1$) to a specific set of positive weight values ($v(-1) = 0.5, v(1) = 1$ in actual implementation), which controls the relative size of the margin. As the scaled weighted soft margin sizes are the same ($|v(t)w(x - b)| = 1$) for both green and red phase observations, smaller weight will lead to a larger distance ($|x - b|$) to the phase change time b . The introduction of the weight function $v(t)$ is due to the fact that compared with the last observation in a phase, the first observation in a phase is more likely to be a vehicle immediately discharged from the queue as the signal light turns green, hence is often closer to the actual phase change time. As a result, it is desired to have smaller margin size for observations after the decision boundary.

Our task is to find the optimal decision boundary b such that the weighted soft margin is maximized (best separation of the two classes of observations). We extend the equivalent formulation for WSMM proposed by Wu and Srihari (2004) and develop a customized formulation (referred to as CWSMM) by introducing an additional temporal smoothing term ($\rho(b - b^{c-1})^2$) as follows:

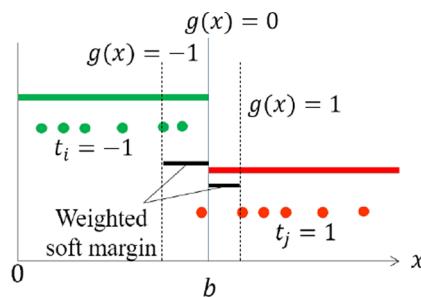


Fig. 3. Illustration of the weighted soft margin maximization approach for signal timing inference.

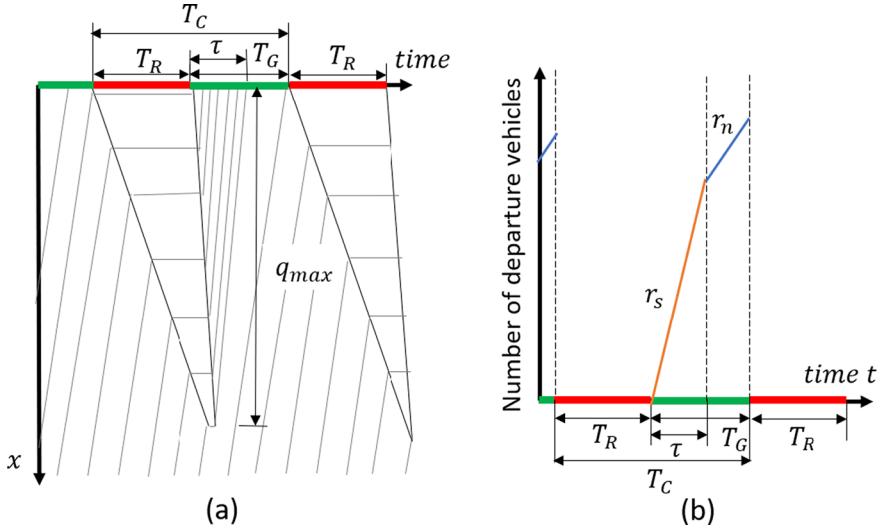


Fig. 4. Illustration of cycle maximum queue length approximation. (a) Space–time diagram of vehicle trajectories with uniform arrivals; and (b) mean cumulative departure process $\mu_D(t|\theta)$ at downstream intersection.

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^2 + M \sum_i \xi_i + \rho(b - b^{c-1})^2 \\ \text{s. t.} \quad & t_i v(t_i) w(x_i - b) + \xi_i \geq 1 \\ & \xi_i \geq 0 \end{aligned} \quad (1)$$

where ξ_i ($\xi_i > 0$) are slack variables for each observation, M is a positive penalty term that controls the trade-off between the penalties of the slack variables and the margin size, ρ is a positive regularization term and b^{c-1} is the decision boundary of the corresponding phase length inferred from the previous signal cycle. For each signal cycle, we run CWSMM twice (each time we move forward one signal phase) to obtain both the green and red phase lengths, and use these values for the inference in the next signal cycle. The original WSMM problem do not have the temporal smoothing term ($\rho(b - b^{c-1})^2$), which can be viewed as a special 1-D reduced form of the weighted support vector machine (WSVM) method (Suykens and Vandewalle, 1999). As most real-world signalized intersections adopt fixed timing plans (the timing plan changes every few hours), the consecutive signal cycles often have identical or similar timing plans. To incorporate this feature, the temporal smoothing term ($\rho(b - b^{c-1})^2$) is added to penalize the difference between the estimated phase length in the current and previous signal cycle. Both CWSMM and WSMM (without $\rho(b - b^{c-1})^2$) are simple 1-dimensional nonlinear optimization problems, which can be efficiently solved by a wide-range of optimization algorithms. Numerical results show that the CWSMM is much more robust than WSMM and SVM (without $\rho(b - b^{c-1})^2$ and weighted margin), especially when the observational data are sparse in a signal cycle. Detailed results please refer to the Section 6.

4. Cycle maximum queue length approximation

4.1. A light-weighted Gaussian process model

To fully utilize the vehicle departing timestamp information in LPR data, we develop a light-weight lane-based Gaussian process cycle maximum queue length approximation model. Suppose $(\zeta_1, x_1), (\zeta_2, x_2), \dots, (\zeta_{n_c}, x_{n_c})$ are the cumulative indices and the timestamps for all vehicles departing at the downstream intersection of a lane within signal cycle c (total number of vehicles is n_c), which form the cycle cumulative departure curve. Under constant vehicle arrival rate, the cumulative departure curve can be considered as the result of the combination of two departure processes with constant mean departure rates: (1) saturation discharging flow with departure rate r_s , which occurs when signal light turns green ($t = T_R$) until the queue is fully dissipated ($t = T_R + \tau$, τ is the queue clearance time); (2) normal departure flow with departure rate r_n ($r_n < r_s$), which occurs after $t = T_R + \tau$ until the signal cycle ends ($t = T_C$). Fig. 4 provides an illustration of this departure process decomposition. From simple shockwave analysis, it can be observed that traffic under undersaturated regime ($\tau < T_G$), the cycle maximum queue length q_{max} can be approximated as the total number of vehicles discharged between $[T_R, T_R + \tau]$; while under oversaturated regime ($\tau \geq T_G$), $q_{max} \geq n_c$.

Consequently, the key to approximate the lane-based cycle maximum queue lengths is to find a set of departure process parameters $\theta = \{r_s, r_n, \tau\}$, which characterizes the most realistic decomposition of the vehicle departure process at the downstream

intersection. To achieve this goal, we develop a non-parametric approach to find the most likely cumulative departure curve given the timestamp information provided in the LPR data using a Gaussian process model. Gaussian process is a powerful method in Bayesian statistical modeling and machine learning. In general, a Gaussian process is defined as a probability distribution over functions $y(\mathbf{x})$ such that the set of values of $y(\mathbf{x})$ evaluated at an arbitrary set of points $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ jointly have a Gaussian distribution. It defines a prior probability distribution over function directly and works with a distribution over the uncountable infinite space of functions (Bishop, 2006; Roberts et al., 2013). Gaussian process serves as an ideal tool in our problem, since we are predicting the most likely function (cumulative departure curve) rather than specific values. For more background information on Gaussian process, please refer to (Bishop, 2006; Williams and Rasmussen, 2006; Roberts et al., 2013).

Denote $\mu_D(t|\theta)$ as the mean cumulative departure process for a lane of a link at the downstream intersection during a signal cycle, then it can be characterized by a piecewise linear function with parameter $\theta = \{r_s, r_n, \tau\}$ (see Fig. 4(b)):

$$\mu_D(t|\theta) = \begin{cases} 0 & 0 \leq t \leq T_R \\ r_s(t - T_R) & T_R < t < T_R + \tau \\ r_s\tau + r_n(t - T_R - \tau) & T_R + \tau < t < T_c \end{cases} \quad (2)$$

The cumulative departure curve given the observed departing timestamps at an intersection can hence be modeled as

$$\zeta(\mathbf{x}|\theta) = \mu_D(\mathbf{x}|\theta) + \eta \epsilon \quad (3)$$

where $\epsilon \sim N(\mathbf{0}, diag(1, \dots, 1))$ is a Gaussian distributed disturbance term and η is a scale parameter. The function $\zeta(\mathbf{x}|\theta)$ can thus be viewed as the cumulative departure vehicle indices ζ 's drawn at timestamps \mathbf{x} from a multivariate Gaussian distribution controlled by parameter θ and η ,

$$p(\mathbf{x}|\theta) = N(\mu_D(\mathbf{x}|\theta), \mathbf{K}(\mathbf{x}, \mathbf{x}|\eta)) \quad (4)$$

which is a Gaussian process with the covariance matrix defined as:

$$\mathbf{K}(\mathbf{x}, \mathbf{x}) = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_{n_c}) \\ \vdots & \ddots & \vdots \\ k(x_{n_c}, x_1) & \cdots & k(x_{n_c}, x_{n_c}) \end{bmatrix} \quad (5)$$

where $k(\cdot)$ is the kernel function to measure the covariance between any pair of departure timestamps on the cumulative departure curve. In this study, we adopt the most widely used squared exponential function with the regulation term as the kernel function:

$$k(x_n, x_m) = h_0 \exp\left(-\left(\frac{x_n - x_m}{\lambda}\right)^2\right) + \eta^2 \delta(x_n, x_m) \quad (6)$$

$$\delta(x_n, x_m) = \begin{cases} 1 & \text{if } x_n = x_m \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where λ is the lengthscale, which controls the smoothness of the kernel function; h_0 is the amplitude, which is a scaling factor determines variation of kernel function value from their means.

The departure process parameters $\theta = \{r_s, r_n, \tau\}$ that parameterizes the most likely cumulative departure curve given the information from the vehicle departing timestamps can be learned by maximizing the joint probability distribution $p(\mathbf{x}|\theta)$. Once the queue clearing time τ is inferred, the cycle maximum queue length can be approximated as follows:

$$q_{max} = \begin{cases} i: x_i \leq T_R + \tau, x_{i+1} > T_R + \tau & \text{ift } < T_G \text{ and } n_c/T_G < r_{threshold} \\ n_c & \text{ift } = T_G \text{ or } n_c/T_G < r_{threshold} \end{cases} \quad (8)$$

where n_c/T_G represents the average departure rate observed from the data, and $r_{threshold}$ is a departure rate threshold introduced to identify overly large average departure rate. This case often occurs under over-saturation condition (queue cannot fully discharge within a signal cycle) and the departure process only contains saturation discharging flow ($\tau = T_G$). Under the undersaturated traffic condition, the maximum queue length can be simply estimated as the number of vehicles discharged during the queue clearing time τ . However, it should be noted that the proposed queue length approximation model can only estimate a lower bound of the maximum queue length under over-saturation traffic condition (second condition of Eq. (8)). This is because in over-saturation condition, the actual maximum queue length is greater than the number of departing vehicles, and the departure process cannot be decomposed into saturation discharging flow and normal departure flow.

4.2. Parameter estimation via MCMC

Inferring θ requires to perform maximum likelihood estimation on the joint probability distribution $p(\mathbf{x}|\theta)$ (Eq. (4)). However, direct maximization of $p(\mathbf{x}|\theta)$ is analytically complex given the non-differentiable piecewise linear mean cumulative departure process $\mu_D(t|\theta)$. In this study, we developed a Metropolis–Hastings (M-H) algorithm to efficiently infer θ using Markov Chain Monte

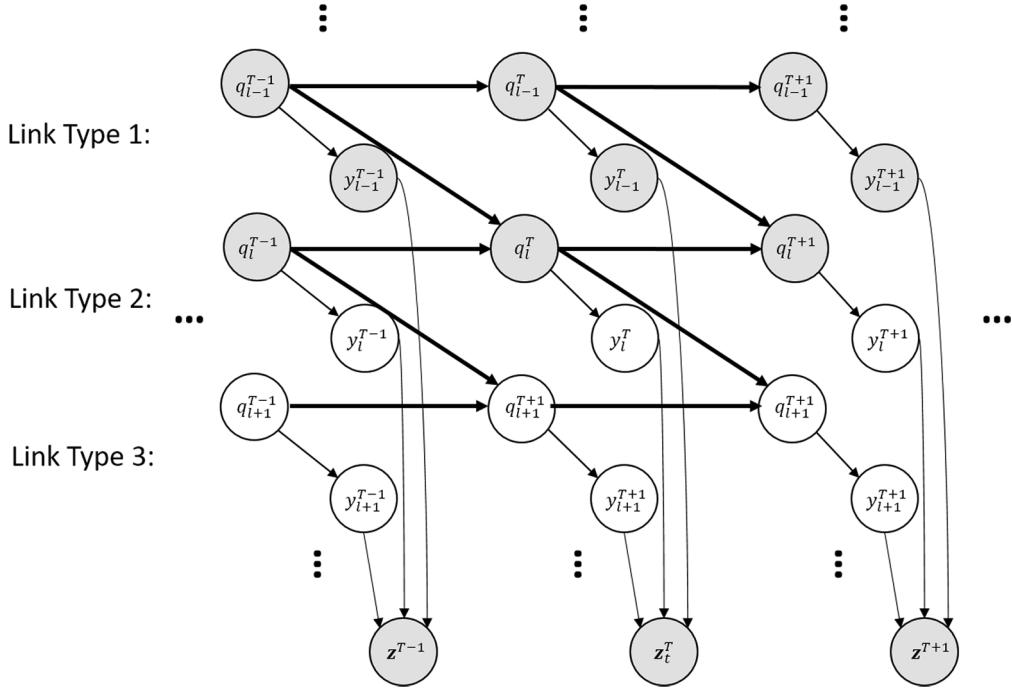


Fig. 5. Graphical illustration of the dynamic linear-Gaussian model for network-wide traffic state inference.

Carlo (MCMC) technique. M-H algorithm is a widely used sampling method, which is useful in estimating parameters from complex probability distributions. More detailed information about M-H algorithm and MCMC technique, please refer to (Bishop, 2006; Berger, 2013). The developed M-H algorithm for learning parameters θ is given in [Algorithm 1](#).

Algorithm 1. H-M algorithm for sampling θ parameters.

- Step 1:** Initialize $\theta^{(0)}$
- Step 2:** Sample $\hat{\theta}$ from following distributions: $\tau \sim U[0, T_G]$, $r_s \sim U[0, n_c/\tau]$, $r_n \sim U[0, r_s]$, where $U[a, b]$ is uniform distribution on interval $[a, b]$.
- Step 3:** Compute and accept $\hat{\theta}$ as $\theta^{(k)}$ according to following ratio:

$$r = \min(p(\mathbf{x}|\hat{\theta})/p(\mathbf{x}|\theta^{(k-1)}), 1)$$
- Step 4:** Repeat step 2-3 for a given number of iterations until $\theta^{(k)}$ are stable.
- Step 5:** Discard the first 75% of accepted samples as burn-in, and use the mean of the remaining sampled θ values as the estimated parameter for θ .

5. Network-wide traffic state inference

5.1. Model description

The core component of our framework is the network-wide link traffic state inference model, which is modeled using a dynamic linear-Gaussian (DLG) model. DLG model is a special class of Dynamic Bayesian Network (DBN) which models the variables using Gaussian distribution and assumes linearity of interaction between variables. In a linear-Gaussian model, the conditional probability distribution between a node y and its parents x is modeled as linear Gaussians ($P(y|x) = N(\beta_0 + \beta^T x, \sigma^2)$, x are the parents of node y) (Roweis and Ghahramani, 1999). The greatest advantage of the linear-Gaussian model over other Bayesian network models lies in its nice analytical property due to the use of Gaussian distribution and fewer model parameters needed, which enables efficient inference in the network. This is very important in modeling large system involving many variables, as using large Bayesian networks involving discrete variables often requires prohibitive amount of computation time to run probabilistic inference, which are impractical to be used in real-world applications. Like other Bayesian network models, linear-Gaussian model is capable of modeling unobserved variables, combined with the DBN framework to capture the temporal dependency, DLG model is an ideal approach for the network-wide link traffic state inference problem proposed in this study. For more background of DBN and linear-Gaussian model, please refer to (Ghahramani, 1998; Roweis and Ghahramani, 1999; Murphy, 2002; Koller and Friedman, 2009).

The representation of the DLG model for this problem is illustrated in Fig. 5. The shaded nodes represent the observed variables and blank nodes represent unobserved (hidden) variables to be inferred. Moreover, for Type 1 links, both the average cycle maximum queue lengths (q_l^T) and the average link travel times (y_l^T) are observed; for Type 2 links, only average cycle maximum queue lengths are observed and the average link travel times are hidden; for Type 3 links, both queue lengths and average link travel times are hidden, which need to be inferred. The forward arrows in Fig. 5 represent the conditional dependencies between the parent node and the child node. The average cycle maximum queue lengths q_l^T of link l in time slice T is assumed to be temporally and spatially conditional dependent on itself q_l^{T-1} and upstream links $q_j^{T-1}, j \in \Gamma(l)$ ($\Gamma(l)$ denote the set of upstream links for link l) in time slice $T - 1$. The average link travel time y_l^T of link l is assumed to be only conditionally dependent on average cycle maximum queue length in the same time slice T . Finally, the path travel times z^T are assumed to be jointly dependent on average travel times of all links y^T .

The average link-level cycle maximum queue length q_l^T for Type 1 and Type 2 links used in the DLG model are obtained by averaging the lane-level queue length estimates (obtained using the queue length approximation model). The lane-level queue length estimate $q_{c,ln}^{\max}$ for signal cycle c and lane ln are averaged based on the number of vehicles recorded on the lanes ($N_{c,ln}$) of the road L_{lane} and the signal cycles within the time slice T , computed as:

$$q_l^T = \frac{\sum_{ln \in L_{lane}} \sum_{c \in T} q_{c,ln}^{\max} \times N_{c,ln}}{\sum_{ln \in L_{lane}} \sum_{c \in T} N_{c,ln}} \quad (9)$$

To fully specify the proposed DLG model, we define the transition model $P(\mathbf{q}^T | \mathbf{q}^{(T-1)})$ and the observation model $P(\mathbf{y}^T | \mathbf{q}^T)$ and $P(\mathbf{z}^T | \mathbf{y}^T)$ as follows.

The transition probability for the average cycle maximum queue length q_l^T of link l transitioning from time slice $T - 1$ to T is modeled to be temporally and spatially conditional dependent on itself q_l^{T-1} and upstream links $q_j^{T-1}, j \in \Gamma(l)$ in time slice $T - 1$, which is

$$P(q_l^T | \mathbf{q}^{T-1}) = P(q_l^T | q_l^{T-1}, q_j^{T-1}, \forall j \in \Gamma(l)) = N(\mu_l^q + \beta_{ll}^q q_l^{T-1} + \sum_{j \in \Gamma(l)} \beta_{lj}^q q_j^{T-1}, \sigma_l^{q2}) \quad (10)$$

where $\mu_l^q, \beta_{ll}^q, \beta_{lj}^q, \sigma_l^{q2}$ are parameter to be learned.

For observation model, given the average cycle maximum queue lengths $q_l^T = q$ for link l , its average link travel time is modeled as,

$$P(y_l^T | q_l^T = q) = N\left(y_l^T | \frac{L_l}{v_l^f} + \mu_l^y + \beta_l^y q, (\sigma_l^y)^2\right) \quad (11)$$

where L_l/v_l^f represents the free-flow travel time for link l , with L_l denotes the link length and v_l^f is the free-flow speed. $\mu_l^y + \beta_l^y q$ and σ_l^y are the mean and standard deviation of delay time for link l caused by average queue length q , and $\mu_l^y, \beta_l^y, \sigma_l^y$ are the learnable parameters. This treatment follows the idea of the well-known point-queue model (Vickrey, 1969) in transportation network modeling literature, that the link travel times can be modeled as the sum of the free-flow travel time and the delay time caused by queuing time proportional to the link queue length.

The path travel times \mathbf{z}^T are modeled as the summation of average link travel times along the observed paths, hence can be conveniently modeled as a linear-Gaussian model, which is

$$P(\mathbf{z}^T | \mathbf{y}^T) = N(\mathbf{z}^T | \mathbf{A}\mathbf{y}^T, \Sigma_p) \quad (12)$$

$$A_{ij} = \begin{cases} 1 & \text{if } l \text{ is on the path } i \text{ for observation } j \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

To reduce the number of parameters to be estimated and enable faster inference, the covariance matrix Σ_p is modeled as a diagonal matrix, which is $\Sigma_p = diag(\sigma_{p_1}^2, \sigma_{p_2}^2, \dots, \sigma_{p_{n_p}}^2)$, where n_p is the number of paths modeled.

Finally, the joint probability for the sequence of states and observations is given as:

$$P(\mathbf{q}^T, \mathbf{y}^T, \mathbf{z}^T) = P(\mathbf{q}^1)P(\mathbf{y}^1 | \mathbf{q}^1)P(\mathbf{z}^1 | \mathbf{y}^1) \prod_{k=2}^T P(\mathbf{q}^k | \mathbf{q}^{k-1})P(\mathbf{y}^k | \mathbf{q}^k)P(\mathbf{z}^k | \mathbf{y}^k) \quad (14)$$

5.2. Exact inference

The central task we wish to solve using the DLG model is the probabilistic inference. For example, given the DLG model defined by parameters $(\mu_l^q, \mu_l^y, \beta_{l,k}^q, \beta_l^y, \sigma_l^q, \sigma_l^y, \Sigma_p, \forall l, k)$ and the observed data D as evidence, we want to obtain the most likely posterior probability explanation of the random variables based on the evidence information. Efficient inference plays a central role when using the DLG model for the traffic state estimation (estimate certain unknown links' state at current time slice) and prediction

(predict all unknown links' states at future time slice) tasks. Both of these two tasks can be achieved by obtaining the posterior marginal distributions of the variables given the data sample D :

$$\begin{aligned} P(q_l^T | D) &= \sum_{k=1}^{NT} \left[\int_{q_l^k \neq q_l^T, \forall l} \cdots \int_{y_l^k, \forall l} \cdots \int_{z_p^k, \forall p} P(D | \mathbf{q}^k, \mathbf{y}^k, \mathbf{z}^k) P(\mathbf{q}^k, \mathbf{y}^k, \mathbf{z}^k) \right] / P(D) \\ P(y_l^T | D) &= \sum_{k=1}^{NT} \left[\int_{q_l^k, \forall l} \cdots \int_{y_l^k \neq y_l^T, \forall l} \cdots \int_{z_p^k, \forall p} P(D | \mathbf{q}^k, \mathbf{y}^k, \mathbf{z}^k) P(\mathbf{q}^k, \mathbf{y}^k, \mathbf{z}^k) \right] / P(D) \end{aligned} \quad (15)$$

To perform the exact inference on the proposed DLG model, we unrolling the DLG model for NT slices (where NT is the length of the modeled time slices) and apply an exact inference algorithm for the static linear-Gaussian model (LGM). LGM is known efficiently solvable using junction tree algorithm (Huang and Darwiche, 1994; Cowell et al., 2006). For details about the junction tree algorithm, the readers are referred to (Huang and Darwiche, 1994; Cowell et al., 2006; Koller and Friedman, 2009). To enforce the behavior of the DBN, the parameters of $\mu_l^q, \mu_l^y, \beta_{l,k}^q, \beta_l^y, \sigma_l^q, \sigma_l^y, \Sigma_p, \forall l, k$ for each link l and path p are shared across different time slices. Moreover, the parameters for the average cycle maximum queue lengths of the first time slice $\mu_l^{q,T=1}, \beta_{l,k}^{q,T=1}, \sigma_l^{q,T=1}, \forall l$ are allowed to take different values compared with $\mu_l^{q,T=2}, \beta_{l,k}^{q,T=2}, \sigma_l^{q,T=2}, \dots, \mu_l^{q,T=NT}, \beta_{l,k}^{q,T=NT}, \sigma_l^{q,T=NT}$ from other time slices. This is because $\mu_l^{q,T=1}, \beta_{l,k}^{q,T=1}, \sigma_l^{q,T=1}, \forall l$ does not depend on any parent nodes and should capture prior information of the queue length states. We implement the junction tree algorithm on the proposed DLG model using the Bayes Net Toolbox developed by Murphy et al. (2001).

5.3. Parameter learning

The proposed DLG model is a large-scale graphical model involving many hidden variables. We use the Expectation–Maximization (EM) algorithm (Dempster et al., 1977) to learn the parameter values of the DLG model ($\mu_l^q, \mu_l^y, \beta_{l,k}^q, \beta_l^y, \sigma_l^q, \sigma_l^y, \Sigma_p, \forall l, k$). EM algorithm is a powerful tool for finding maximum likelihood solution for models involving hidden variables. It is proved to converge to a local maximum of the observed data likelihood function (Dempster et al., 1977). The core idea of EM algorithm is that at each step, we first “fill in” the missing variable values with their expected values (E-step), then use the resulting complete data likelihood function to compute their expected sufficient statistics (ESS) and perform maximum likelihood estimation (MLE) (M-step). An outline of the EM algorithm used in this paper is presented in Algorithm 2.

Algorithm 2. EM algorithm for learning parameters of the DLG model

Step 1: Randomly initialize parameter values of the DLG: $\mu_l^q, \mu_l^y, \beta_{l,k}^q, \beta_l^y, \sigma_l^q, \sigma_l^y, \Sigma_p, \forall l, k$.

Step 2: Repeat

1. Reset the ESS for each variable and the log-likelihood $L = 0$.

2. (E-Step) For each training case e :

(a) Update the log-likelihood: $L = L + \log P(e | \mu_l^q, \mu_l^y, \beta_{l,k}^q, \beta_l^y, \sigma_l^q, \sigma_l^y, \Sigma_p, \forall l, k)$.

(b) Compute the posterior marginal distributions over each variable given the evidence e .

(c) Update the ESS for each variable.



Fig. 6. Study region.

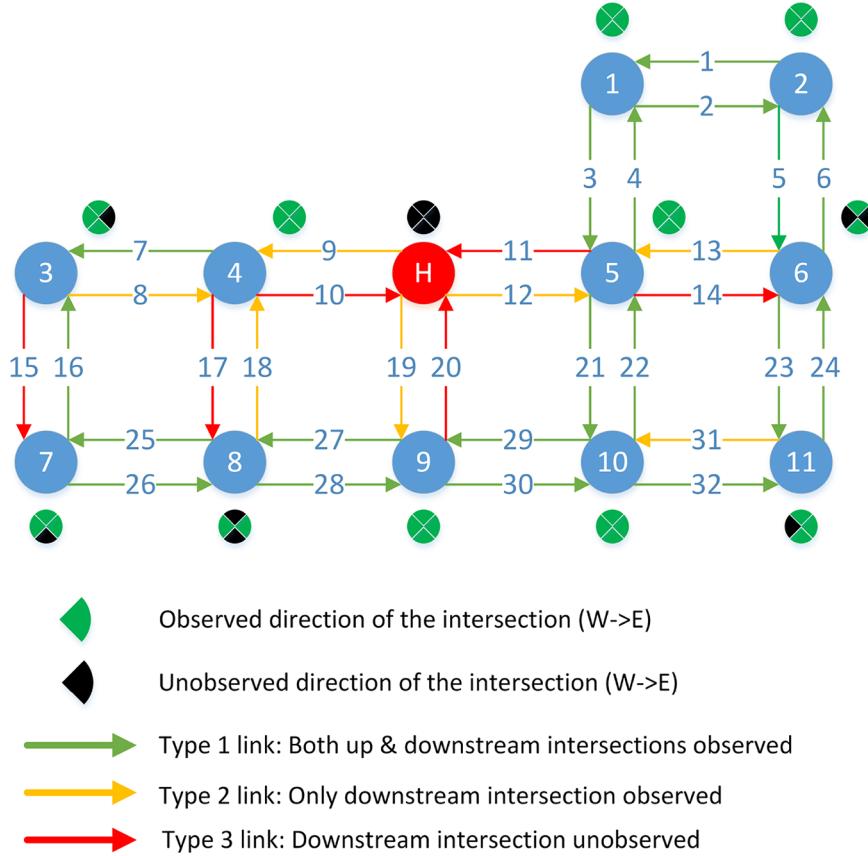


Fig. 7. Testing network.

3. (M-Step:) Compute the MLE of the parameters for each variable given the ESS.

Step 3: Terminate if L converges.

Step 2.2(a) and (b) in [Algorithm 2](#) are the core steps of the E-step, which can be computed using the junction tree inference algorithm discussed in the previous section. The detailed forms, update methods and the maximum likelihood estimation of the ESS for the three different types of variables in the proposed DLG model can be found in [Murphy \(1998\)](#). The technique of learning with shared parameters in DBN using the aggregate sufficient statistics can be found in [Koller and Friedman \(2009\)](#). To remove redundancy, these details will not be included in the paper. More information on EM algorithm and parameter learning in hybrid networks please refer to [Murphy \(1998\)](#) and [Koller and Friedman \(2009\)](#).

6. Experiment results

6.1. Experiment setup

To fully test and validate the proposed statistical modeling framework, we obtain two different datasets from the city of Langfang, China. The main dataset is a six-day network-level LPR dataset (2013/11/11–12, 11/14–17) from a small network in the city (see [Fig. 6](#)). The test network contains 12 signalized intersections and 32 links. 11 intersections are equipped with LPR cameras. However,

Table 1
Modeled paths of the DLG model.

Path ID	Links	Path ID	Links	Path ID	Links
1	10, 12	4	14, 23	7	10, 19
2	11, 9	5	15, 26	8	20, 9
3	14, 6	6	17, 25	9	20, 12

not all intersections are fully monitored. There are 6 intersections have unmonitored approaches, which are represented as the black sectors in Fig. 7. Some of these intersections do not have LPR cameras installed on certain approaches, as the roads of these approaches are not very important. More unmonitored approaches are caused by the failure of LPR cameras. This results in a partially observed arterial network with 19 fully observed (Type 1) links, 7 partially observed (Type 2) links and 6 unobserved (Type 3) links. A detailed illustration of the testing network with directional observability is presented in Fig. 7. This dataset is used to test and validate the proposed network-wide traffic state inference model. Due to the missing of ground truth information from other data sources, we focus on validating the average travel times inferred from the model against the actual average travel times that directly observed in the LPR data.

We also obtain a more detailed link-level LPR dataset from a field experiment conducted in a previous work of the authors (Zhan et al., 2015). The field experiment was conducted on a 720 m arterial segment (Link 4 of the testing network, see Fig. 7) on November 26th, 2014 from 7 am to 12 pm. In the field experiment, we obtained the LPR data from both the upstream and downstream intersections. Moreover, detailed ground truth information, including the actual timing plans of the downstream intersection (Intersection 1) from 7 am to 11:30 am as well as the actual cycle maximum queue lengths from 7 am to 10:15 am are collected. The signal timing inference model and the queue length approximation model are tested on the link-level LPR dataset and validated against the collected ground truth information.

In the actual implementation of the network-wide traffic state inference model, we set the time slice length as 180 s and considered 4 time slices in the DLG model. The reason that we only consider four time slices is due to the limited amount of data available. Using more time slices will result in smaller number of data cases for parameter learning, which can potentially impact model performance. We use five days' data (2013/11/12, 11/14, 11/16–17) to learn the parameters of the DLG model, and use one day's data (2013/11/15) for testing. For each day, data from four time periods are extracted and used: 7:30 am–9 am, 9:30 am–12 pm, 13:30 pm–16:30 pm, 17 pm–19 pm. Data from other time periods are not used mainly due to the existence of missing data in some intersections during these time periods. The link and path travel times with departure timestamps fall in every 180 s interval are extracted from the network-level LPR dataset and processed into average link and path travel times. To incorporate more information during the inference of traffic states of Type 2 and Type 3 links, we modeled 9 paths in the DLG model. Each path contains one or two Type 2 or 3 links. The detailed path information can be found in Table 1. The signal timing plans and the lane-based cycle maximum queue lengths are also computed using the signal timing inference model and the queue length approximation model. These

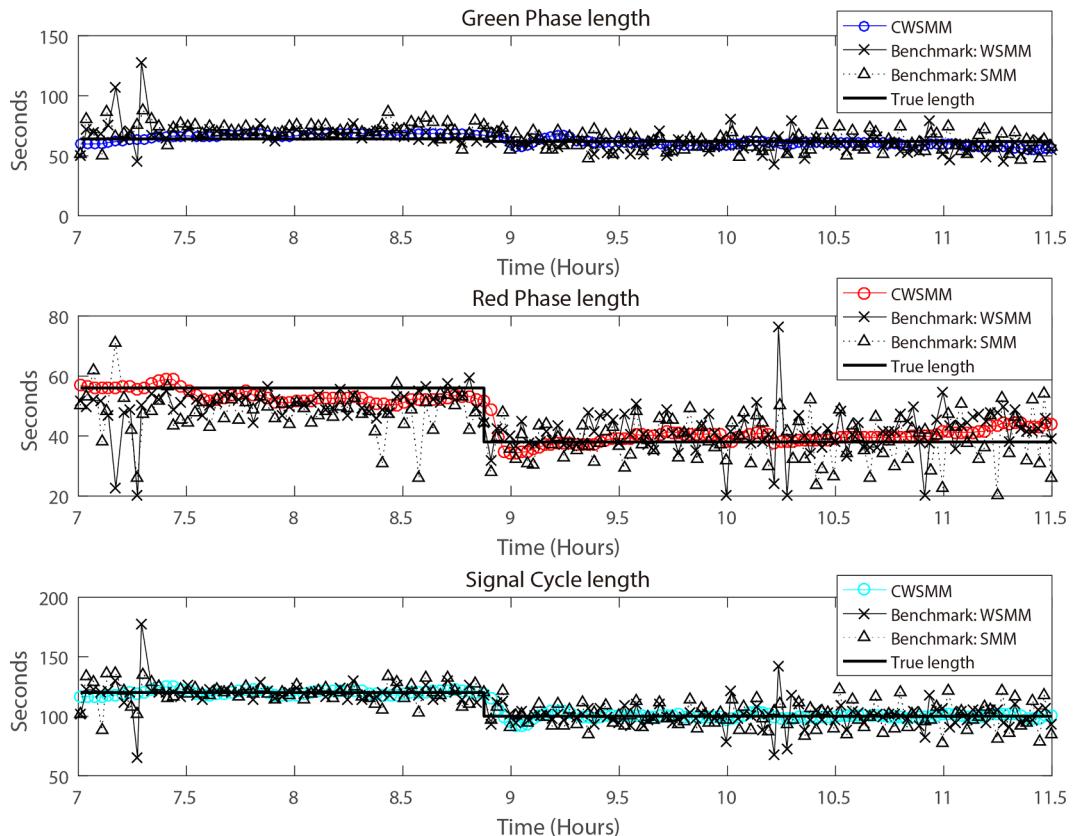


Fig. 8. Comparison of the signal cycle and phase length inference results of CWSMM and WSMM. The signal cycle length is obtained as the sum of the green and red phase lengths. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2

Overall signal timing inference results.

Model	Metrics	Signal cycle length	Green phase length	Red phase length
CWSMM	MAE	3.23s	5.88s	5.57s
	MRE	1.51%	4.69%	6.28%
WSMM	MAE	13.03s	12.53s	11.48s
	MRE	6.08%	9.99%	12.91%
SMM	MAE	15.73s	14.52s	14.46s
	MRE	7.33%	11.58%	16.26%

information along with the average link and path travel times served as the input to the DLG model.

Two metrics, namely mean absolute error (MAE) and the mean relative error (MRE) are used to evaluate the performance of each component in the proposed framework, which are calculated as follows:

$$MAE = \frac{\sum_{i=1}^n |d_i - \hat{d}_i|}{n}, \quad MRE = \frac{\sum_{i=1}^n |d_i - \hat{d}_i|}{\sum_{i=1}^n d_i} \quad (16)$$

where d_i is the ground truth value for observation i and \hat{d}_i is the predicted values; n is the total number of observations in the testing set.

6.2. Evaluation on the signal timing inference and queue length approximation

Using the detailed ground truth data in the link-level LPR dataset, we first present the validation results of the signal timing inference model and the queue length approximation model.

6.2.1. Evaluation on the signal timing inference model

The testing intersection (Intersection 1) adopted fixed timing plans with ground truth cycle length of 120s (green phase length: 64s, red phase length: 56s) during 7:00 am-8:55 am, and 100s (green phase length: 62s, red phase length: 38s) during 8:55 am - 11:30 am. We test the performance of the proposed customized WSMM model with temporal smoothing term (CWSMM) against two benchmark methods, which are the weighted soft margin maximization (WSMM) model (1-D reduced form of weighted SVM) and soft margin maximization (SMM) model (1-D reduced form of SVM without weight, i.e. $v(-1) = v(1) = 1$). The parameters of SMM, WSMM and CWSMM are set as $M = 0.1$, $\rho = 0.02$, and the initial phase lengths b^0 are set as 60s for both green and red phases. The detailed results are presented in Fig. 8 and Table 2.

The results show that the proposed CWSMM model achieves good performance in estimating signal timing plans. The MRE of the green and red phase length estimates are around 5% and 6% respectively. The MRE of signal cycle length estimates is even below 2%. CWSMM greatly outperforms the benchmark WSMM and SMM model with much lower MAE and MRE in all tests. From Fig. 8, it can

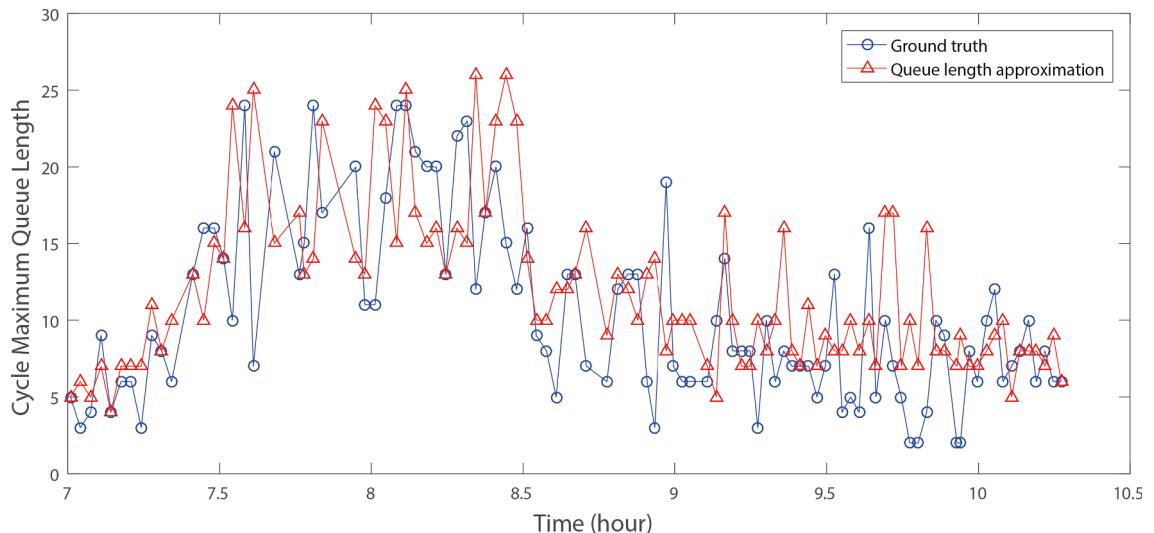


Fig. 9. Cycle maximum queue length approximation comparison. MAE = 2.34, MRE = 27.12%.

be observed that the both SMM and WSMM results have much larger variances and sometimes significantly deviate from the ground truth values. This typically occurs when there are very few vehicle timestamp observations in the signal cycle or cases with many overlapping green and red phase observations (see the discussion in Section 3). Under such cases, finding the decision boundary of the actual phase change time is much more difficult. By contrast, CWSMM overcomes this difficulty and produces more robust estimates. This is mainly due to the introduction of the temporal smoothing term, which penalizes the result if it deviates too much from the phase length of the previous signal cycle. This incorporates the prior information of fixed timing plan. Furthermore, compare the results of SMM and WSMM, it can be observed that using heterogeneous weight values ($v(-1) < v(1)$), WSMM achieves lower MAE and MRE in all tests. This also confirms the effectiveness of using heterogeneous weight values in the signal timing inference problem. It should also be noted that introducing the temporal smoothing term will also make the CWSMM less sensitive compared with WSMM to abrupt changes in signal timing plan. For example, when the signal timing plan changed at 8:55 am, the CWSMM needs 2–3 signal cycles to converge to the actual phase length, whereas such delay is not observed in the WSMM results. The sensitivity of CWSMM can be controlled by varying the magnitude of the temporal smoothing parameter ρ . For signalized intersections that adopt semi-actuated or fully actuated controls, a smaller ρ should be considered.

6.2.2. Evaluation on the queue length approximation model

The proposed queue length approximation model is validated against the ground truth cycle maximum queue lengths on the through lane of the experiment road. The signal timing data estimated from the signal timing inference model are used to run the queue length approximation model. The hyperparameters in the Gaussian process model are selected as $h_0 = 0.5$, $\lambda = 5$, $\eta = 2$, $r_{threshold} = 0.41\text{veh/s}$ which produce cumulative departure curves that are neither too smooth nor too spiky. When running the H-M algorithm to sample the parameter θ , we use 20,000 iterations with the burn-in ratio of 75%. Detailed queue length approximation results are presented in Fig. 9. The queue length approximation model achieves reasonable estimation accuracy, with MAE = 2.34 and MRE = 27.12%. The estimated cycle maximum queue length curve captures the pattern of the actual cycle maximum queue length curves well. The accuracy of the proposed queue length approximation model is even better compared with the model developed in Zhan et al. (2015) (MAE = 2.75), which used LPR data from both the upstream and downstream intersections and a embedded car-following model to capture queuing process. Compared with the model in Zhan et al. (2015), the queue length approximation model proposed in this study is light-weighted, calibration-free and only needs information from the downstream intersection. This makes it particularly desirable for our network-wide real-time traffic state inference problem. As the queue length approximation model are not able to estimate maximum queue length under over-saturation traffic condition (see discussion in Section 4), it is likely that the model will underestimate the actual queue length when heavy queuing is present, which lead to potential inaccuracies in the queue length estimates.

6.3. Evaluation on the network-wide traffic state inference model

As there is no ground truth data available for the network-level LPR dataset, we select two Type 1 links (Link 5 and 26) and 1 observed path (Path 1), and use their actual average travel times as ground truth for evaluation. The DLG model is trained using five days' LPR data (2013/11/12, 11/14, 11/16–17) and tested using one day's data (2013/11/15). In the subsequent discussion, we first validate the Gaussian assumption of the DLG model using the real-world dataset. Then, we perform two evaluation experiments on the network-wide traffic state inference model. The first experiment evaluates the accuracy of travel time estimation. In this experiment, the average travel times of the two test links and one test path are set as unobserved and estimated by performing inference on the DLG model using the test data. The detailed results of the link and path travel time estimations are analyzed in Section 6.3.2. The second experiment explores the prediction capabilities of the proposed network-wide traffic state inference model. In this experiment, we remove all the input data in the last 1 to 3 time slices in the test data (treated as unknown data in future time slices) and perform inference on the DLG model using the data only from the first few time slices. This is very close to the case when using DLG model for prediction. Detailed results of the prediction analysis are presented in Section 6.3.3.

6.3.1. Validation on the Gaussian assumption of the DLG model

A major assumption made in the DLG model is that the marginal distributions of the variables are Gaussian distributed. This property enables efficient inference in the DLG model and the possibility of using the model for large traffic network. To check if the Gaussian assumption is supported in the real-world dataset, we plotted the histograms and Normal distribution fittings of estimated average cycle maximum queue lengths (Type 1 and 2 links) and average link travel times (Type 1 links) for some links in the testing network, see Fig. A.11 and A.12 in Appendix. It can be observed that although not fitted perfectly, the average cycle maximum queue lengths and average travel times of most links can be fitted reasonably well using Normal distribution. This confirms the validity of the DLG model used in the network-wide traffic state inference problem.

6.3.2. Evaluation on travel time estimation

We perform exact inference on the DLG model to obtain the posterior marginal distribution of the average travel times of the two test links $P(y_l^T | D)$ and one test path (see Eq. (15)), which are Gaussian distributions $N(y_l^T | \mu_l^P, (\sigma_l^P)^2)$, $N(z_p^T | \mu_p^P, (\sigma_p^P)^2)$ with posterior

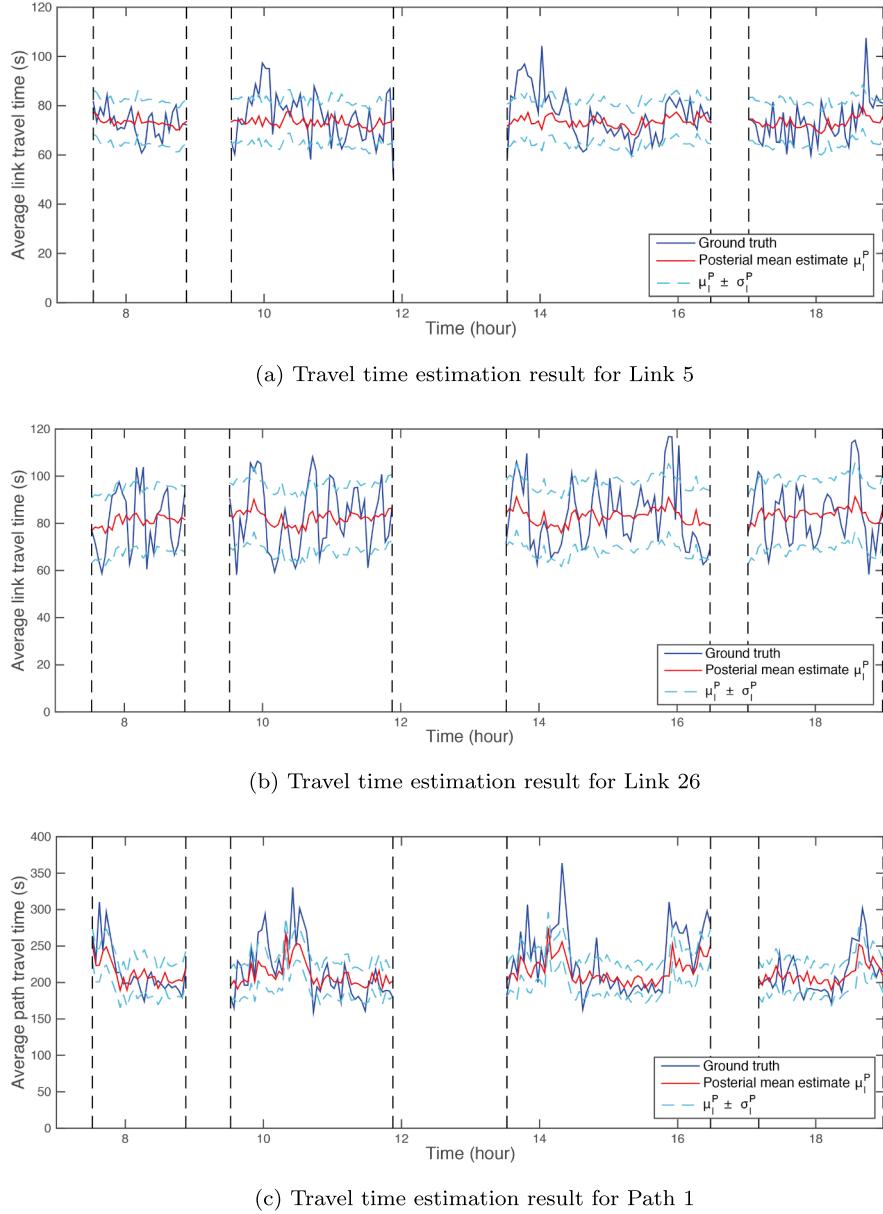


Fig. 10. Travel time estimation results for the Link 5, 26 and Path 1.

Table 3
Travel time estimation results.

Link	MAE	MRE	$P_{\mu_l^P \pm \sigma_l^P}$	$P_{\mu_l^P \pm 2\sigma_l^P}$
Link 5	6.791s	8.98%	71.43%	94.29%
Link 26	10.340s	12.493%	75.00%	100%
Path 1	20.535s	8.82%	70.06%	91.02%

mean μ_l^P , μ_p^P and standard deviation σ_l^P , σ_p^P . To evaluate the results, we compute the MAE and MRE between the estimated posterior mean values μ_l^P and the ground truth average link travel times. Moreover, as the model estimates the average travel time as a probability distribution, two additional metrics are reported: the proportion of data fall within $[\mu_l^P - \sigma_l^P, \mu_l^P + \sigma_l^P]$ interval (denoted

Table 4
Travel time prediction results

Link	T_0	T_P	MAE	MRE	$P_{\mu_l^P \pm \sigma_l^P}$	$P_{\mu_l^P \pm 2\sigma_l^P}$
Link 5	3	1	6.266s	8.39%	68.26%	95.81%
	2	1	6.268s	8.42%	66.47%	97.00%
	2	2	6.540s	8.28%	71.86%	95.81%
	1	1	6.498s	8.40%	70.66%	96.41%
	1	2	6.501s	8.64%	75.45%	95.21%
	1	3	6.711s	8.94%	71.26%	95.21%
Link 26	3	1	6.655s	8.30%	76.79%	96.43%
	2	1	6.728s	8.39%	76.19%	97.02%
	2	2	7.759s	9.70%	73.81%	95.83%
	1	1	6.761s	8.41%	77.38%	97.02%
	1	2	7.999s	10.01%	74.40%	95.83%
	1	3	8.747s	10.89%	73.81%	96.43%
Path 1	3	1	18.705s	8.16%	70.89%	94.30%
	2	1	18.786s	8.26%	69.62%	93.04%
	2	2	19.817s	8.56%	70.89%	91.14%
	1	1	20.406s	8.95%	65.41%	91.82%
	1	2	21.040s	9.20%	69.62%	90.51%
	1	3	21.738s	9.31%	68.99%	91.14%

as $P_{\mu_l^P \pm \sigma_l^P}$) and the proportion of data fall within $[\mu_l^P - 2\sigma_l^P, \mu_l^P + 2\sigma_l^P]$ interval (denoted as $P_{\mu_l^P \pm 2\sigma_l^P}$). These two measures reflect the explanatory power of the model results. The detailed estimation results for the two test links are presented in Fig. 10 and Table 3.

The results show that the estimated posterior marginal distribution well captures the ground truth average travel times. Comparing the posterior mean μ_l^P with the ground truth, the MRE for both test links are controlled under 13%, and the MAE are around 6–11s. For the test on Path 1, the MRE is less than 9% and the MAE is about 20s. Nearly all the ground truth average travel times are covered within the $[\mu_l^P - 2\sigma_l^P, \mu_l^P + 2\sigma_l^P]$ interval, and more than 70% of data are covered within the $[\mu_l^P - \sigma_l^P, \mu_l^P + \sigma_l^P]$ interval. This shows that the estimated posterior marginal distributions explain the ground truth average link travel times well. Compared with many travel time estimation methods that only provide point estimates, our approach estimates the posterior distributions of the average link travel times with both mean and standard deviation, thus provides a more robust characterization of the estimated values. In this study, we only have a six-day LPR dataset. The limited amount of training data could lead to potentially biased parameter estimates during learning the conditional transition distributions in the DLG model (e.g. $P(\mathbf{q}^{T-1}|\mathbf{q}^T)$, $P(\mathbf{y}^T|\mathbf{q}^T)$). The accuracy of the estimation results can be further improved if a larger LPR dataset with a longer temporal duration is used.

6.3.3. Prediction capabilities

We also explored the prediction capabilities of the proposed model. Here, we refer to the prediction at T_P time slice as the computation of the most likely traffic states of the network at time slice $T_0 + T_P$ given data observed up to and including the time slice T_0 . The prediction can be perceived as a special case of the estimation process, in which all the data are missing for all links and all time slices after T_0 . Again, the prediction can be achieved by performing exact inference on the DLG model with unknown observations at the predicting time slices. The assessment of the prediction capability is performed on the same test data (data from 2013/11/15) as the evaluation of the travel time estimation. We test the prediction performance of the model with T_0 ranges from 1 times slices ($T_P = 1, 2$ and 3) to 3 time slices ($T_P = 1$). For comparison purpose, the same test links are investigated. Table 4 presents the detailed evaluation metrics of the travel time prediction experiments.

The results show that the prediction error increases with the increase of the prediction steps (T_P), which is intuitive. Moreover, it is found that the increase of the error is relatively small for both links. For example, for the case of $T_0 = 1$, the differences in MRE between $T_P = 1$ and 3 are less than 0.5% for Link 5, about 2.5% for Link 26 and less than 0.5% for Path 1. This shows the prediction capabilities of the model remain accurate for several future time slices. It is also observed that the prediction accuracy generally

Table 5
Computational performance of the proposed model.

Components	Time
Online components	
Signal timing inference (per intersection and signal cycle)	0.53s
Queue length approximation (per lane and signal cycle)	1.60s
HDBN-inference (per 4 time slices: 180s × 4)	29.78s
Offline components	
HDBN-learning	21.85 h

improves (smaller MAE and MRE) with the increase of observed time slices T_0 . This is as expected, as more observed data from past time slices will provide more accurate prior information about the network condition, thus facilitates the traffic state prediction. Compared with the travel time estimation results in [Table 3](#), it is observed that the prediction results achieve better accuracy when the observed time slices are long enough and the prediction steps are small. For Link 26, all the prediction results are found to outperform the estimation results. This observation is not surprising, as in the estimation experiment, the data from the two test links are completely missing. The traffic states of the two links need to be inferred completely from the traffic states of neighboring links as well as the path travel times. On the prediction settings, the traffic states of the two links are given in the first T_0 time slices as prior knowledge, which greatly improves the inference even when the information from neighboring links are unknown in the future.

6.4. Computational performance

[Table 5](#) presents the computational performance of each component of the proposed framework. The experiments are conducted on a Quad-Core 3.4 GHz CPU and 16 GB RAM desktop computer. All the four components are implemented in Matlab. The inference and learning of the DLG model are implemented based on the Bayes Net Toolbox library ([Murphy et al., 2001](#)). The online components of the framework are very efficient. For different roads and intersections, all the online components can be computed in parallel, which is important for real-time traffic state estimation and prediction. The largest amount of the time is spent on the inference of the DLG model. For the implementation of the framework on larger networks, it is suggested to partition the network into smaller sub-networks and apply the DLG on each sub-network. This will reduce the computation complexities and also enable parallel running different DLG models for each sub-network, which will further reduce the overall computation time when modeling large arterial networks.

The parameter learning of the DLG model is very slow compared with the online components, however, it only needs to run offline. The learning process can be performed periodically (e.g. monthly) using the latest data. An advantage of the DLG model is that it can be incrementally improved. That is, the parameters of the previously learned DLG model can be used as the initial parameters when learning with the EM algorithm. This can also greatly improve the convergence speed of future model learning processes.

7. Conclusion and discussion

LPR data are emerging data sources that have huge potential in estimating and forecasting urban arterial network performances. We develop a data-driven statistical framework to infer real-time average cycle maximum queue lengths and average link travel times for an arterial network partially monitored by LPR cameras. The proposed framework contains three components, namely the signal timing inference model which infers signal timing information from vehicle departure timestamps; the queue length approximation model which uses a Gaussian process model to estimate lane-based queue lengths; and finally a network-wide traffic state inference model based on DLG model performs estimation and prediction on links with unobserved traffic states. A six-day network-level LPR dataset and a more comprehensive link-level dataset collected in a field experiment are used to test and validate the model. The proposed framework achieves good inference accuracy in the numerical experiments. The computational performance also demonstrates the efficiency of the framework, which allows for real-time implementation in urban networks with reasonable LPR camera coverage. The proposed framework, as well as its sub-modules are also applicable to other types of vehicle re-identification data that share similar characteristics with LPR data (e.g. RFID data). The recorded vehicle identifiers (e.g. real or virtual license-plate number, RFID tags, etc.) and the passing timestamps information in such data sources can be used in the same way as LPR data in the proposed framework.

This work provides an integrative statistical modeling framework to incorporate the rich information of LPR data in network-wide traffic state estimation and prediction. It contributes to the literature in following aspects:

1. We present the first study in literature that exploits and utilizes the unique features of LPR data for urban network-level traffic condition estimation and prediction.
2. We develop a complete solution for network-wide link average cycle maximum queue lengths and average travel time estimation and prediction using a single source of data. The developed statistical framework combines both well-established traffic flow theory and highly customized statistical machine learning models.
3. New methodologies of using LPR data for signal timing inference, queue length approximation and network inferences from partially available data are developed. The methodologies are not restricted to LPR data, but also applicable to other vehicle re-identification data with similar characteristics.
4. The proposed framework is highly efficient and calibration-free, which can be easily implemented in real-world arterial networks.
5. LPR data from a six-day network-level dataset and a link-based field experiment dataset are used to test and validate the model. LPR data of such scale and comprehensiveness have never been examined in the literature.

Several future works can be done to further improve this research. First, the network-level LPR dataset used in this study contains only six days' data, which limits the amount of data usable when learning the DLG model. It will be meaningful to test the proposed

framework with larger LPR datasets and examine the performance improvement under different sizes of training data. Second, as there is no ground truth information from other data sources in the network-level LPR dataset, we only validated the average travel times in the network-wide traffic state inference model. Additional field experiments can be conducted in the future to collect ground truth queue lengths in order to further validate the estimation and prediction accuracy of the average cycle maximum queue length.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China (Grant No. 71361130015). We also thank the Bureau of Traffic Management of Langfang for providing the data.

Appendix A. Distributions of average cycle maximum queue lengths and average link travel times

Fig. A.11 and A.12

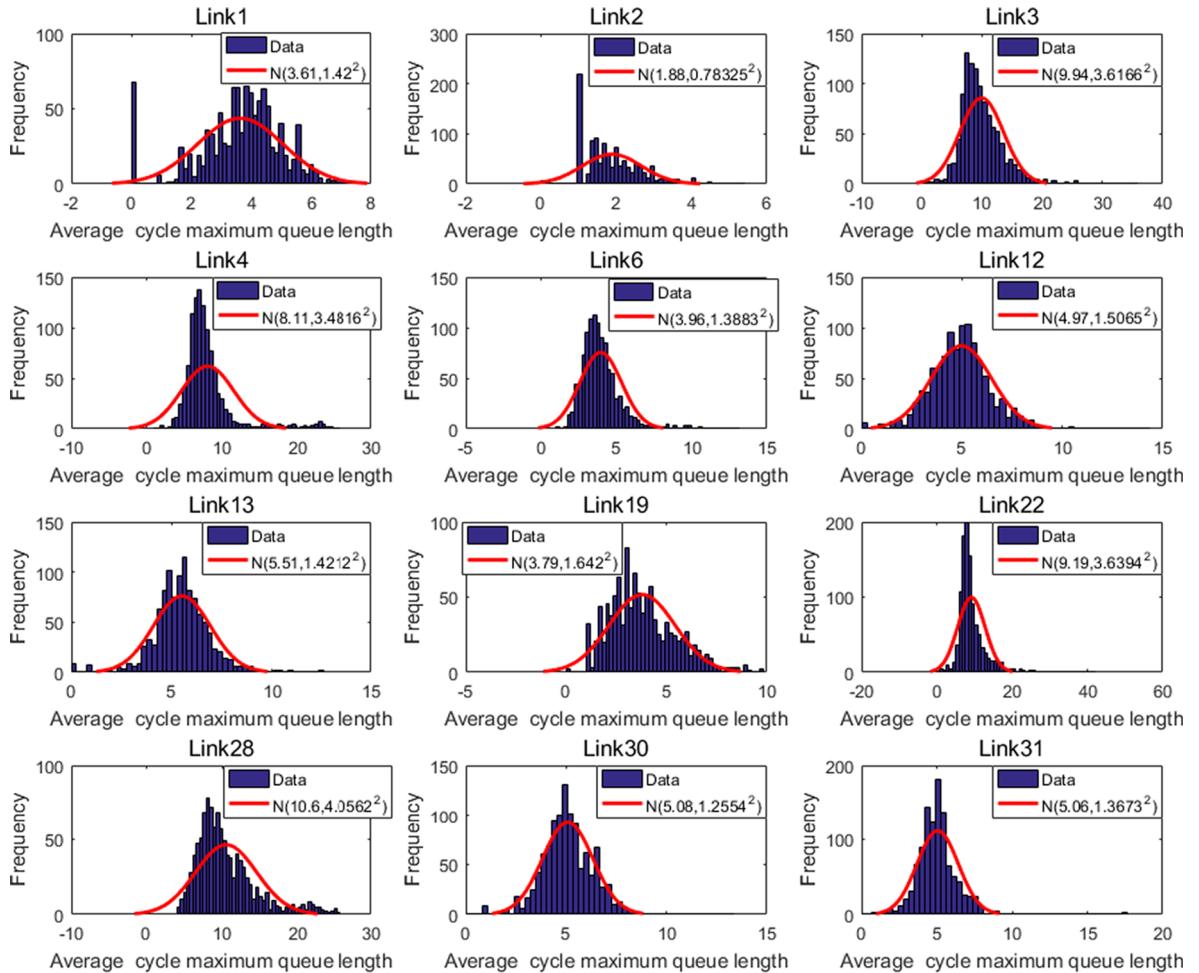


Fig. A.11. Histograms and normal distribution fittings of the estimated average cycle maximum queue lengths for some Type 1 and 2 links in the testing network.

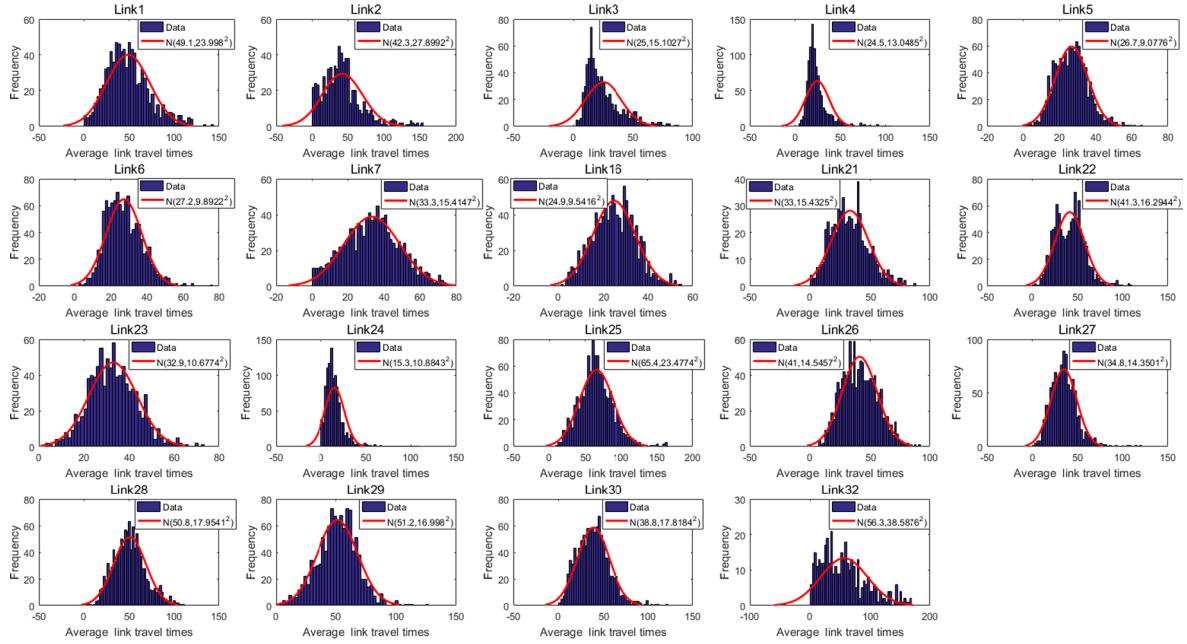


Fig. A.12. Histograms and normal distribution fittings of the average link travel times for Type 1 links in the testing network.

Appendix B. Summary of notations used in the article

2.1. Problem definition

D	LPR dataset.
I	Intersection ID.
ID^n	License-plate number of vehicle n .
t_n^d	Departing timestamp of vehicle n .
ln^n	Departing lane of vehicle n .
T	Index of time slice. Also used to represent vector or matrix transpose.
q_l^T	Average cycle maximum queue length of link l in time slice T .
y_l^T	Average link travel time of link l in time slice T .
x_i^T	Departing timestamp of vehicle i of a lane in time slice T .
z_p^T	Average path travel time for path p in time slice T .
\mathbf{q}^T	Vector of queue length states of all links in time slice T .
\mathbf{y}^T	Vector of average travel time of all links in time slice T .
\mathbf{z}^T	Vector of average path travel time of all paths in time slice T .
\mathbf{x}^T	Departing timestamp sequence of a specific lane from an observed intersection.

3. Traffic signal timing inference

x_i	Observed vehicle departing timestamp for vehicle i in a signal cycle.
t_i	Label of vehicle i indicating whether it belongs to the first phase (-1) or the second phase ($+1$).
b	Actual phase change time, which serves as the decision boundary for two signal phases.
b^c	Estimated phase change time of signal cycle c .
$g(\cdot)$	A decision function to classify vehicle departing timestamps to different signal phases.
$v(\cdot)$	A weight mapping from the observed label to a specific set of positive weight values.
O_G^T, O_R^T	Empirically observed longest green and red phase intervals in LPR data.
w	A scale parameter in decision function $g(\cdot)$.
ξ_i	The slack variable for the departing timestamp of vehicle i .
M	A positive penalty term that controls the trade-off between the penalties of the slack variables and the margin size.
ρ	A positive regularization term for the estimated phase change time b .

 4. Cycle maximum queue length approximation

ζ_i	Cumulative departing index of vehicle i .
n_c	Total number of departing vehicles at the downstream intersection of a lane within signal cycle c .
T_R	Red phase length.
T_G	Green phase length.
T_C	Signal cycle length.
τ	Queue clearance time.
r_n	Departure rate for normal departure flow.
r_s	Departure rate for saturation discharging flow.
q_{max}	Cycle maximum queue length.
θ	Departure process parameters $\theta = \{r_n, r_s, \tau\}$.
$\mu_D(t \theta)$	Mean cumulative departure process for a lane of a link at the downstream intersection during a signal cycle, characterized by parameter θ .
$\zeta(x \theta)$	The cumulative departure curve given the observed departing timestamps at an intersection.
$N(\mu, \Sigma)$	Multivariate Gaussian distribution with mean μ and covariance matrix Σ .
ϵ	A multivariate Gaussian distributed disturbance term.
η	A scale parameter for the Gaussian distributed disturbance term.
$k(\cdot)$	Kernel function used to construct the Gaussian process.
$K(x, x \eta)$	Covariance matrix constructed using the kernel function $k(\cdot)$.
$\delta(x_n, x_m)$	Indicator function with value 1 if $x_n = x_m$; 0 otherwise.
h_0	Amplitude parameter, which is a scaling factor determines variation of kernel function value from their means.
λ	Lengthscale parameter, which controls the smoothness of the kernel function.
$r_{threshold}$	Departure rate threshold for identifying overly large normal departure rate.
$U[a, b]$	Uniform distribution on interval $[a, b]$.

 5. Network-wide traffic state inference

$q_{c,ln}^{max}$	Lane-level queue length estimate for signal cycle c and lane ln .
L_{lane}	The set of all lanes of a link.
$N_{c,ln}$	Number of vehicles recorded on the lane ln during signal cycle c .
q^{max}	Link-level average cycle maximum queue length of a time slice.
$\Gamma(l)$	The set of upstream links for link l .
β_l^q	Weight parameter in the linear-Gaussian distribution for q_l^T of link l .
μ_l^q	Offset parameter in the linear-Gaussian distribution for q_l^T of link l .
σ_l^q	Standard deviation for q_l^T of link l .
L_l	Length of link l .
v_l^f	Free-flow speed of link l .
β_l^y	Weight parameter in the linear-Gaussian distribution for y_l^T of link l .
μ_l^y	Offset parameter in the linear-Gaussian distribution for y_l^T of link l .
σ_l^y	Standard deviation for y_l^T of link l .
\mathbf{A}	Link-path correspondence matrix.
Σ_p	A diagonal covariance matrix for path travel time distribution. $\Sigma_p = diag(\sigma_{p_1}^2, \dots, \sigma_{p_{n_p}}^2)$, where n_p is the number of path modeled.
NT	Number of time slices.
L	Log-likelihood of the DLG model.

 6. Experiment Results

MAE	Mean absolute error.
MRE	Mean relative error.
d_i	Ground truth value for observation i .
\hat{d}_i	Predicted value for observation i .
$P_{\mu_l^P \pm \sigma_l^P}$	The proportion of data fall within $[\mu_l^P - \sigma_l^P, \mu_l^P + \sigma_l^P]$ interval.
$P_{\mu_l^P \pm 2\sigma_l^P}$	The proportion of data fall within $[\mu_l^P - 2\sigma_l^P, \mu_l^P + 2\sigma_l^P]$ interval.
T_O	The time slices with observed data.
T_P	The time slices to perform prediction.

References

- Ahmadi, A., Jahangiri, A., Berardi, V., Machiani, S.G., 2018. Crash severity analysis of rear-end crashes in California using statistical and machine learning classification methods. *J. Transport. Safety Secur.* 1–25.
- Axer, S., Friedrich, B., 2017. Signal timing estimation based on low frequency floating car data. *Transport. Res. Proc.* 25, 1645–1661.
- Ban, X.J., Hao, P., Sun, Z., 2011. Real time queue length estimation for signalized intersections using travel times from mobile sensors. *Transport. Res. Part C: Emerg. Technol.* 19, 1133–1156.
- Beijing Traffic Management Bureau, 2017. Location table of fixed traffic monitoring equipment. <<http://www.bjjtl.gov.cn/zhuanti/09sxt/040.html>>.
- Berger, J.O., 2013. Statistical decision theory and Bayesian analysis. Springer Science & Business Media.
- Bertini, R.L., Lasky, M., Monsere, C.M., 2005. Validating predicted rural corridor travel times from an automated license plate recognition system: Oregon's frontier project. *IEEE Proc. Intell. Transport. Syst.* 296–301.
- Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.
- Chen, H., Yang, C., Xu, X., 2017. Clustering Vehicle Temporal and Spatial Travel Behavior Using License Plate Recognition Data. *J. Adv. Transport.* 2017, 1738085.
- Coifman, B., Cassidy, M., 2002. Vehicle reidentification and travel time measurement on congested freeways. *Transport. Res. Part A: Policy Pract.* 36, 899–917.
- Coifman, B., Krishnamurthy, S., 2007. Vehicle reidentification and travel time measurement across freeway junctions using the existing detector infrastructure. *Transport. Res. Part C: Emerg. Technol.* 15, 135–153.
- Cowell, R.G., Dawid, P., Lauritzen, S.L., Spiegelhalter, D.J., 2006. Probabilistic networks and expert systems: Exact computational methods for Bayesian networks. Springer Science & Business Media.
- Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the em algorithm. *J. Royal Stat. Soc. Ser. B Methodol.* 1–38.
- Du, Z., Yan, X., Zhu, J., Sun, W., 2019. Signal timing parameters estimation for intersections using floating car data. *Transport. Res. Rec.* 2673, 189–201.
- Fayazi, S.A., Vahidi, A., Mahler, G., Winckler, A., 2014. Traffic signal phase and timing estimation from low-frequency transit bus data. *IEEE Trans. Intell. Transp. Syst.* 16, 19–28.
- Ghahramani, Z., 1998. Learning dynamic bayesian networks, in: Adaptive processing of sequences and data structures. Springer, pp. 168–197.
- Hao, P., Ban, X., Bennett, K.P., Ji, Q., Sun, Z., 2012. Signal timing estimation using sample intersection travel times. *IEEE Trans. Intell. Transp. Syst.* 13, 792–804.
- He, X., Das, R., 2015. RFID in China 2015–2025: forecasts, players, opportunities. Technical Report. IDTechEX Research.
- Herrera, J.C., Work, D.B., Herring, R., Ban, X.J., Jacobson, Q., Bayen, A.M., 2010. Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment. *Transport. Res. Part C: Emerg. Technol.* 18, 568–583.
- Herring, R., Hofleitner, A., Abbeel, P., Bayen, A., 2010. Estimating arterial traffic conditions using sparse probe data, in: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, IEEE. pp. 929–936.
- Hofleitner, A., Herring, R., Bayen, A., 2012. Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning. *Transport. Res. Part B: Methodol.* 46, 1097–1122.
- Huang, C., Darwiche, A., 1994. Inference in belief networks: a procedural guide. *Int. J. Approx. Reason.* 11, 158.
- Hunter, T., Herring, R., Abbeel, P., Bayen, A., 2009. Path and travel time inference from gps probe vehicle data. NIPS Analyzing Networks and Learning with Graphs 12.
- Jahangiri, A., Rakha, H.A., 2015. Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. *IEEE Trans. Intell. Transport. Syst.* 16, 2406–2417.
- Jeng, S.T., Tok, Y.C.A., Ritchie, S.G., 2010. Freeway corridor performance measurement based on vehicle reidentification. *IEEE Trans. Intell. Transp. Syst.* 11, 639–646.
- Koller, D., Friedman, N., 2009. Probabilistic graphical models: principles and techniques. MIT press.
- Kwong, K., Kavalier, R., Rajagopal, R., Varaiya, P., 2009. Arterial travel time estimation based on vehicle re-identification using wireless magnetic sensors. *Transport. Res. Part C: Emerg. Technol.* 17, 586–606.
- Li, R., Rose, G., 2011. Incorporating uncertainty into short-term travel time predictions. *Transport. Res. Part C: Emerg. Technol.* 19, 1006–1018.
- Liu, H.X., Wu, X., Ma, W., Hu, H., 2009. Real-time queue length estimation for congested signalized intersections. *Transport. Res. Part C: Emerg. Technol.* 17, 412–427.
- Mo, B., Li, R., Zhan, X., 2017. Speed profile estimation using license plate recognition data. *Transport. Res. Part C: Emerg. Technol.* 82, 358–378.
- Murphy, K., et al., 2001. The bayes net toolbox for matlab. *Comput. Sci. Stat.* 33, 1024–1034.
- Murphy, K.P., 1998. Inference and learning in hybrid Bayesian networks. University of California, Berkeley, Computer Science Division.
- Murphy, K.P., 2002. Dynamic bayesian networks: representation, inference and learning. Ph.D. thesis. University of California, Berkeley.
- Ni, Y., Wang, M., Sun, J., Li, K., 2016. Evaluation of pedestrian safety at intersections: a theoretical framework based on pedestrian-vehicle interaction patterns. *Acc. Anal. Prevent.* 96, 118–129.
- Oh, C., Ritchie, S.G., Jeng, S.T., 2007. Anonymous vehicle reidentification using heterogeneous detection systems. *IEEE Trans. Intell. Transp. Syst.* 8, 460–469.
- Park, D., Rilett, L., 1998. Forecasting multiple-period freeway link travel times using modular neural networks. *Transport. Res. Rec.: J. Transp. Res. Board* 163–170.
- Roberts, S., Osborne, M., Ebden, M., Reece, S., Gibson, N., Aigrain, S., 2013. Gaussian processes for time-series modelling. *Phil. Trans. R. Soc. A* 371, 20110550.
- Roweis, S., Ghahramani, Z., 1999. A unifying review of linear gaussian models. *Neural Comput.* 11, 305–345.
- Shafique, M.A., Hato, E., 2015. Use of acceleration data for transportation mode prediction. *Transportation* 42, 163–188.
- Sharma, A., Bullock, D., Bonneson, J., 2007. Input-output and hybrid techniques for real-time prediction of delay and maximum queue length at signalized intersections. *Transport. Res. Rec.: J. Transp. Res. Board* 69–80.
- Sherali, H.D., Desai, J., Rakha, H., 2006. A discrete optimization approach for locating automatic vehicle identification readers for the provision of roadway travel times. *Transport. Res. Part B: Methodol.* 40, 857–871.
- Skabardonis, A., Geroliminis, N., 2008. Real-time monitoring and control on signalized arterials. *J. Intell. Transport. Syst.* 12, 64–74.
- Suykens, J.A., Vandewalle, J., 1999. Least squares support vector machine classifiers. *Neural Process. Lett.* 9, 293–300.
- Tseng, F.H., Hsueh, J.H., Tseng, C.W., Yang, Y.T., Chao, H.C., Chou, L.D., 2018. Congestion prediction with big data for real-time highway traffic. *IEEE Access* 6, 57311–57323.
- Vickrey, W.S., 1969. Congestion theory and transport investment. *Am. Econ. Rev.* 59, 251–260.
- Vigos, G., Papageorgiou, M., Wang, Y., 2008. Real-time estimation of vehicle-count within signalized links. *Transport. Res. Part C: Emerg. Technol.* 16, 18–35.
- Wang, Y., Zheng, Y., Xue, Y., 2014. Travel time estimation of a path using sparse trajectories, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 25–34.
- Williams, C.K., Rasmussen, C.E., 2006. Gaussian processes for machine learning. the MIT Press 2, 4.
- Wu, C.H., Ho, J.M., Lee, D.T., 2004. Travel-time prediction with support vector regression. *IEEE Trans. Intell. Transport. Syst.* 5, 276–281.
- Wu, X., Srihari, R., 2004. Incorporating prior knowledge with weighted margin support vector machines, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM. pp. 326–333.
- Xumei, C., Huibo, G., Wang, J., 2012. Brt vehicle travel time prediction based on svm and kalman filter. *J. Transport. Syst. Eng. Inform. Technol.* 12, 29–34.
- Yasin, A.M., Karim, M.R., Abdullah, A.S., 2010. Travel time measurement in real-time using automatic number plate recognition for Malaysian environment. *J. East. Asia Soc. Transport. Stud.* 8, 1738–1751.
- Yeon, J., Elefteriadou, L., Lawphongpanich, S., 2008. Travel time estimation on a freeway using discrete time markov chains. *Transport. Res. Part B: Methodol.* 42, 325–338.
- Zhan, X., Hasan, S., Ukkusuri, S.V., Kamga, C., 2013. Urban link travel time estimation using large-scale taxi data with partial information. *Transport. Res. Part C: Emerg. Technol.* 33, 37–49.
- Zhan, X., Li, R., Ukkusuri, S.V., 2015. Lane-based real-time queue length estimation using license plate recognition data. *Transport. Res. Part C: Emerg. Technol.* 57, 85–102.

- Zhan, X., Ukkusuri, S.V., Yang, C., 2016. A bayesian mixture model for short-term average link travel time estimation using large-scale limited information trip-based data. *Autom. Constr.* 72, 237–246.
- Zhan, X., Zheng, Y., Yi, X., Ukkusuri, S.V., 2017. Citywide traffic volume estimation using trajectory data. *IEEE Trans. Knowl. Data Eng.* 29, 272–285.
- Zhang, X., Rice, J.A., 2003. Short-term travel time prediction. *Transport. Res. Part C: Emerg. Technol.* 11, 187–210.
- Zhang, Y., Xie, Y., 2007. Forecasting of short-term freeway volume with v-support vector machines. *Transp. Res. Rec.* 2024, 92–99.
- Zheng, F., VanZuylen, H., 2013. Urban link travel time estimation based on sparse probe vehicle data. *Transport. Res. Part C: Emerg. Technol.* 31, 145–157.
- Zhu, H., Li, M., Zhu, Y., Ni, L.M., 2009. Hero: Online real-time vehicle tracking. *IEEE Trans. Parallel Distrib. Syst.* 20, 740–752.