Estimating Travel Time Distributions by Bayesian Network Inference

Anatolii Prokhorchuk, Justin Dauwels, and Patrick Jaillet

***Abstract*—Travel time estimation is an important aspect of intelligent transportation systems (ITS). In urban environments, travel times can exhibit much variability due to various stochastic factors. For this reason, we focus on estimating travel time distributions, in contrast to the more commonly studied estimation of mean expected travel times. We present algorithms to infer travel time distributions from Floating Car Data; specifically, from sparse GPS measurements. The framework combines Gaussian copulas and network inference to estimate marginal and joint distributions of travel times. We perform an extensive set of numerical experiments on one month of GPS trajectories. We benchmark the proposed models in terms of Kullback-Leibler (KL) divergence and Hellinger distance for the 50 most common trajectories. Combining Gaussian Copulas and Bayesian Inference of Sparse Networks method achieves 4.9% reduction in KL divergence and 2% reduction in Hellinger distance compared to baseline methods.**

***Index Terms*—Travel time distribution estimation, Gaussian copulas, graphical lasso, Bayesian network inference.**

I. Introduction

INTELLIGENT Transportation Systems (ITS) aim to decrease traffic congestion, reduce fuel consumption, and improve other aspects of transportation. Travel time information plays a crucial role in different ITS applications, such as stochastic routing [1], ridesharing [2], and traffic monitoring [3]. Better estimates of travel time can improve the overall performance of such systems, leading to more accurate information for the users including better routes and better transportation options [4]. Another crucial aspect for ITS users is the variation of travel time, affecting all of the above use cases. However, most of the previous studies [5]–[9] attempt to estimate expected travel times. Recent studies [10]–[14] on distribution estimation usually deal with origin-destination (OD) pairs or dense GPS data, instead of sparse GPS trajectories. Sparse GPS data is commonly acquired by vehicles, therefore, estimating travel time distributions from such data is an important topic. Nevertheless, the literature on this topic is very limited.

In this paper, we present novel ways to infer the travel time distributions from sparse GPS data. We aim to develop a method which is able to reliably estimate travel time distributions from sparse GPS data. This will allow utilizing more easily obtainable data to provide the benefits to ITS applications. In our approach, we infer the covariance structure of travel times in the road network. As a result, the method is able to reliably learn the distribution between any origin-destination pair. To estimate the covariance structure, we combine Gaussian copula with the recently developed Bayesian network inference algorithm [15] which yields more accurate network structures and precision matrices compared to graphical lasso [16]. The main advantage of this approach is the ability to handle variables with a different number of observations. The proposed method does not require more expensive, high-resolution data and is well-suited for sparse, low-resolution GPS trajectories. We perform extensive numerical experiments on a large dataset of sparse GPS recordings to validate the proposed method. Interestingly, the proposed approach may also be applied to high-resolution data, after it has been down-sampled to low-resolution data; such approach might be viable when the available computational resources are limited.

We consider a dataset provided by one of the major taxi companies in Singapore. It contains GPS positions sampled by a fleet of over 15,000 taxis during the period of August 2010. In total, approximately 12 million trips are included in the dataset. Each position (latitude and longitude) is accompanied by a timestamp and the status of the taxi. Each vehicle records this data with a certain sampling interval. Due to the large size of the network (several thousand links) and large sampling intervals (60 seconds on average), we divide the dataset into 1-hour intervals. For each interval, we infer travel time distributions from 70% of the paths, randomly selected in that interval, and then evaluate the inferred distributions by computing Kullback-Leibler divergence and Hellinger distance with the empirical distribution of travel time for the remaining 30% of the paths. Our proposed model utilizes Gaussian copulas and Bayesian inference of sparse networks (BISN, [15]) framework and outperforms baseline methods by 4.9% in terms of the KL divergence and 2% in terms of the Hellinger distance. We also investigate the performance of the proposed model by training travel time models on data from time intervals of varying length, and comparing those models to empirical travel time distributions from later time intervals.

Manuscript received June 11, 2018; revised October 12, 2018 and December 18, 2018; accepted January 31, 2019. Date of publication March 5, 2019; date of current version May 1, 2020. This work was supported in part by the Singapore National Research Foundation through the Singapore-MIT Alliance for Research and Technology (SMART) Centre for Future Urban Mobility (FM). The Associate Editor for this paper was G. Mao. *(Corresponding author: Anatolii Prokhorchuk.)*

A. Prokhorchuk and J. Dauwels are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: anatolii001@e.ntu.edu.sg; jdauwels@ntu.edu.sg).

P. Jaillet is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: jaillet@mit.edu).

Digital Object Identifier 10.1109/TITS.2019.2899906

TABLE I

BRIEF SUMMARY OF THE RELEVANT APPROACHES FOR TRAVEL TIME DISTRIBUTION ESTIMATION

Method	Data Source/Type	Model for individual link	Model for dependence	Comments
Ramezani & Geroliminis [24]	Several sources (traffic cameras, loop detectors, probe vehicles)	Markov chains for states of 2D diagrams of consecutive links	A mixture of distributions, where each base distribution is a convolution of partial TTD of each link state	Good results for TT estimation for arterials but the method requires high resolution GPS data
Hunter et al. [11]	GPS trajectories	Gaussian random variables	Gaussian Markov Random Field (GMRF) based on Partial Empirical Covariance Matrix (PECM)	Good results for TT estimation for a large network, however, the method requires high resolution GPS data
Wan & Kornhauser [12], [13]	GPS trajectories	Copulas	Kernel method, lagged Gaussian Copula; Gaussian copula mixture model	Incorporated time lag into the dependence structure and used graphical lasso for copula estimation, but the evaluation was done on a small dataset
Feng et al [28]	GPS trajectories	Gaussian mixtures	Markov chain	Applied to traffic state estimation. Gaussian mixtures fixed to 2 peaks, Markov chains considered only a neighbouring link, experiments done on limited datasets
Prokhorchuk et al. [14]	GPS trajectories	Scaling method	Gaussian Copula and Graphical lasso	Combined Gaussian copulas and graphical lasso for travel time distribution estimation, but the test dataset was limited to only one day

The rest of the paper is organized as follows. In Section II we summarize the literature on the problem of estimating travel times and travel time distributions. In Section III we describe our approach in detail. In Section IV we present and discuss the results of our experiments on Singapore taxi data. In Section V we offer concluding remarks and ideas for future research.

II. RELATED WORK

Travel time estimation is a well-studied topic in the literature [5]–[9], [17]–[23]. Most of the previous studies deal with the estimation and prediction of travel times, without providing confidence intervals on these estimates [5]–[9]. Recently, travel time distribution estimation received more attention [10]–[14]. The problem of estimating the travel times and travel time distributions heavily depends on the sources of data. These sources include inductive loop detector data, probe vehicles, and video streams from traffic cameras.

One of the most common sources is data from loop detectors and similar equipment installed on the roads. Depending on the type of the detector, loop detectors can record different traffic parameters such as flow and speed. There are several common travel time estimation techniques for this type of data. Certain models are rooted in traffic flow theory [5], [6], while other techniques are data-driven and do not require theoretical traffic models. Time-series analysis is a family of data-driven techniques [7]–[9] that include SVR [7] and Artificial Neural Networks (ANN) [18], [19]. Kalman filters, another time-series based approach, have also been applied to travel time estimation; for example, Dailey [20] apply Kalman filters to estimate travel time based on occupancy and volume data from loop detectors.

Probe vehicles are another viable source of travel time data. These vehicles are supplied with GPS systems that record location data and timestamps (and sometimes other information such as speed) with a certain sampling interval, usually from every few seconds to more than a minute. GPS data also poses a number of different challenges related to the nature of data, including location data errors, and the sparsity of the signals both in terms of sampling interval and number of probe vehicles. Due to the popularity of GPS devices, there is an increasing number of studies about this kind of data [10]–[13], [23]–[27] aiming to overcome the mentioned challenges. Statistical models such as Gaussian mixtures [28], copulas [12], [13], and regression methods [23] are commonly applied to GPS data, in addition to graphical models such as Markov chains [24], [28] and Bayesian Networks [25].

In the following, we briefly review studies concerning estimation of travel time distributions. An important consideration for studies with probe vehicles data is the sampling interval, i.e., how often vehicles report their location. Sampling intervals in the literature range from around one second [11] to only reporting origin and destination [26]. In Table I we summarize some of the more relevant studies.

Ramezani and Geroliminis [24] proposed modeling travel time distributions with Markov chains. For each pair of consecutive links they build a 2D diagram of travel times. Next, they apply heuristic grid clustering to this diagram to compute a state of the travel time for these particular links. They apply Markov chains to compute the transition probabilities and travel time distributions. This approach produces good distribution estimates even for arterial links, however, it requires high-resolution travel time data. Hunter *et al.* [25] proposed to apply Bayesian networks for travel times estimation using only origin-destination pairs as input data. They applied Expectation-Maximization (EM) algorithm to handle path uncertainty. The approach showed promising results, however, it assumes independence of link travel times. As a next step, Hunter *et al.* [11] proposed a different method for travel time distributions estimation. They developed an approach for dense GPS data sampled at 1Hz that infers the number of stops made by a vehicle on a particular road segment. In that study, travel time distributions were estimated using Markov model for the number of stops coupled with a Gaussian Markov Random Field. The proposed model performed better than the baseline ones, however, the method requires high-resolution data.

Another promising approach for travel time distribution estimation from probe vehicle data is graphical models [12]–[14], [25]. Wan and Kornhauser [13] applied graphical lasso and Gaussian copulas for travel time estimation. They approximate the path travel time distribution by the sum of a series of conditional path travel time distributions. These distributions are conditioned on a lag vector (a vector of entering times experienced by a vehicle to each link in the path). In this framework the dependence between the travel times at different edge takes into account the time lag along the trip. To address the possible low number of observations, lagged Gaussian copula parameters are estimated by the graphical lasso method to achieve a sparse precision matrix. In [12] Wan introduced Gaussian Copula Mixture Models (GCMM) for path travel time estimation. In this approach, the travel times are modeled for different scenarios. To compute the total travel time for a given path, they integrate the conditional path travel time distributions over a set of path scenarios. Each scenario represents certain traffic conditions and the statistical dependence between travel times at different links in each scenario is stationary. The parameters of each Gaussian copula are again estimated via the graphical lasso. However, the numerical results in [12] are quite limited, since only three paths consisting of less than 15 links each are considered.

In earlier work [14], we combined Gaussian copulas and the graphical lasso approach to infer the travel time distributions for parts of the Singapore network. In this paper, we conduct a more in-depth study with a larger dataset, containing more trajectories and spanning a larger time period. We also propose a Bayesian framework that goes beyond the graphical lasso, and is able to deal with sparse GPS data in a large network. Sparse GPS data is easier to obtain and more common than high-resolution GPS data. Therefore, the proposed framework can more widely be utilized. To address the drawbacks of the graphical lasso method [12]–[14], we propose to apply a novel Bayesian Inference of Sparse Networks (BISN) [15] method. The graphical lasso method does not take into account the difference in the number of vehicles traversing different links, while BISN allows us to accurately model travel time distributions for the usual scenario where some links are more frequently traversed than others. At the same time, it has lower computational complexity. We investigate the effect of using copulas for path travel time modeling and how the estimation performance depends on various factors such as path length, day of the week, and hour of the day.

Previous studies described above are either limited to small networks or time frames [12], [13], [28], require hard-to-obtain high-resolution GPS data [11], [24] or both. Moreover, some of the studies require strong assumptions on travel time distributions, such as independence, Gaussianity and others. In contrast, the proposed method is able to estimate the travel time distributions for paths in a large traffic network, based on sparse data, without assumptions on link travel time distributions while also taking the variable link coverage across the network into account.

III. Methodology

A. Approach

We will first describe the problem statement, next we will outline our statistical modeling approach. We aim to estimate travel time distributions of paths in a transportation network from sparse GPS data. We are provided with 1 month (August of 2011) of taxi GPS data from one of the taxi companies in Singapore. Each data point contains geographical coordinates, taxi identifier, timestamp, and current state. The possible values for the latter are *FREE* (the taxi is looking for passengers), *ON CALL* (the taxi has received a booking and is on its way to pick up the client), *POB* (passenger on board), and periods of inactivity. We are interested only in trips made in the POB state, since they typically represent the real traffic conditions. This particular dataset has been studied in [17], where it is shown that traffic patterns can be inferred from taxi trajectories, and only 700 (out of more than 15,000) to obtain a network coverage of 70%. From this collection of GPS trajectories, we wish to infer the travel time distribution for any path in the network for one-hour intervals.

In the following, we will outline our statistical modeling approach for travel time distributions. We model the travel times associated with paths in a network as a multivariate Gaussian or copula Gaussian random variable. A path consists of multiple links, and its travel time is a sum of individual link travel times, each modeled as a random variable; in multivariate copula Gaussian models, the travel times at individual links are not necessarily Gaussian distributed. In order to determine the distribution of the path travel time, we need to infer how the travel time at each link depends on the travel time at the other links. In both Gaussian and copula Gaussian models, this statistical relationship is fully captured by the covariance matrix Σ and its inverse (precision matrix K). The precision matrix encodes the dependency among different variables in an elegant manner: if two variables x_i and x_j are conditionally independent, the corresponding element $K_{i,j}$ in the precision matrix is zero.

We model the network as a graph $G = (V, E)$, where V is a set of vertices, representing nodes in the network (such as intersections), and E is a set of arcs, representing road links (26 972 in the network at hand). We define the path as a sequence of consecutive road links and the trajectory as a path traversed by a taxi. For each trajectory, we know the timestamp of the taxi arriving at the first link in the path and the timestamp of it leaving the last one. The general overview of our proposed approach is displayed in Fig. 1. First, we convert the raw GPS coordinates data into paths matched onto the graph G by means of the HMM-based algorithm proposed in [29].

Next, from these projected paths along with the reported timestamps we compute travel times for each individual link. We assume a constant speed of the taxi throughout all links in a given path. Then we calculate travel time for each link proportionate to the link length. We will refer to this approach as the "scaling method" (cf. Section C). Approaches based on proportions are commonly considered in studies dealing with sparse GPS trajectories [30], [31]. In the dataset at hand,

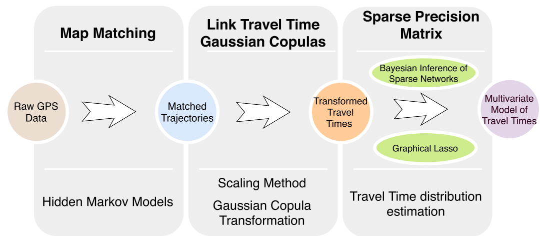

Fig. 1. Diagram of the proposed approach for travel time modeling.

all road links are covered by a large number of different trajectories. In some cases, only a subset of links is congested for a given path during the current time interval. However, when we compute travel times of individual links we do this irrespective of any particular path. This implies that travel times for each congested link will be partially computed from trajectories covering only congested links. Due to the amount of data, congested links will have larger travel times and vice versa. At the same time, the proposed distribution estimation method (namely, combining Gaussian copulas with Bayesian network inference) can be applied together with any other individual link travel time computation approach. For example, a tensor-based approach [10] can be utilized.

In order to model the non-Gaussian distribution of these travel times estimates, we apply Gaussian copulas, transforming the non-Gaussian travel time estimates into Gaussian random variables (cf. Section F).

After this transformation, we compute the Partial Empirical Covariance Matrix (PECM; cf. Section E). It serves as a baseline method to infer the covariance structure of the travel times at the different links in the transportation network. Next, we apply the graphical lasso algorithm [16] (graphical lasso; cf. Section G) to the PECM, in order to impose sparsity on the precision matrix, yielding a more accurate estimate of the covariance matrix of the travel times. Besides the graphical lasso method, we also apply an alternative to graphical lasso, named "Bayesian Inference of Sparse Networks" (BISN; cf. Section H), as it has been shown to lead to more accurate estimates of sparse networks from data [15]. As a result, this method is able to take into account the fact that the number of data points (travel times) typically varies across the links. By contrast, in the graphical lasso approach, one implicitly assumes that each link has the same number of data points (travel times). Indeed the graphical lasso is directly applied to the PECM, and no confidence levels on the elements of the PECM are provided; these confidence levels depend on the number of data points associated with each pair of links. Another important advantage of BISN is its lower computational complexity; the computational complexity of BISN only scales quadratically with the number of variables (links in the transportation network), while the graphical lasso has cubic computational complexity.

From the transformed travel times obtained by the scaling method, we compute the mean travel times and their empirical covariance matrix; we apply the graphical lasso and BISN to the latter, in order to obtain more reliable estimates of the covariance matrix. As a result, we obtain a multivariate copula model of the travel times specified by the mean vector of travel times and the covariance matrix; the latter is either obtained by the graphical lasso or BISN, and we will assess both approaches in this paper. Path travel time estimates are computed by sampling from the joint distribution of the links in the path. The approach is described in Sections H and I. For evaluation, we select the 50 most common trajectories in the network, and compare the obtained distribution with the empirical one. Specifically, we compute two different measures, i.e., the Kullback-Leibler (KL) divergence and the Hellinger distance between the empirical distribution of travel times and the marginal distributions obtained from the different travel time models.

In the following, we will explain each step in the travel time modeling pipeline.

B. Preprocessing

Data from probe vehicles tends to be noisy; the location information typically only has an accuracy of 5m to 10m. Therefore, it is important to preprocess the data and project the GPS points onto the road network. This procedure is called map matching (MM), and several methods have been proposed for this purpose, including geometric, topological, probabilistic and artificial intelligence based approaches [32]. Since the GPS data considered in this paper is sparse, we apply a map matching algorithm based on Hidden Markov Models that is particularly well suited for such sparse data [29]. For this algorithm, each possible road segment is represented as a hidden state with its observation probability $p(\text{observation})$ depending on the distance between the trajectory point and the segment d, road segment width $2w$ and standard deviation of GPS error δ_g:

$$p(\text{observation}) = \frac{1}{2w}\int_{-w}^{w}\frac{1}{2\pi\delta_g^2}e^{-\frac{(l-d)^2}{2\delta_g^2}}\,d\ell. \qquad (1)$$

The emission probability $p(\text{observation}|r)$ is defined as:

$$p(\text{observation}|r)=\frac{v_{\max}}{\max(0, v_{\text{obs}}-v_{\max})+v_{\max}}p(\text{observation}), \qquad (2)$$

where r is the candidate segment, v_{obs} is the observed speed and v_{max} is the speed limit. The ratio in (2) represents the speeding penalty factor.

Goh *et al.* [29] apply SVM to compute the transition probability between states. They determine the most likely sequence of road segments by means of a modified Viterbi algorithm. We refer the reader to [29] for more details.

C. Link Travel Time Estimation Using Scaling Method

After obtaining map matched data, we compute travel times for each link. One of the most straightforward methods to achieve this is to assume that a taxi vehicle has a constant average speed throughout the whole multi-link path. We compute the travel time t_i for a link i with length ℓ_i by rescaling

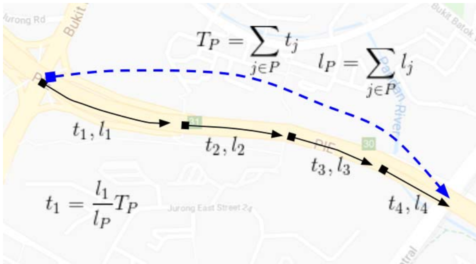

Fig. 2. Illustration of the scaling method (3) for computing link travel times from path travel times. In this example, the path (blue dashed line) contains four links (black arrows).

the path travel time T_P as follows:

$$t_i = \frac{\ell_i T_P}{\sum_{j \in P} \ell_j}. \tag{3}$$

We provide an illustration of this calculation in Fig. 2. After computing these estimates from all the trajectories (from different taxis) we average values for each link to compute the mean travel time.

To improve the accuracy of these intermediate travel time estimates we incorporate information about the position of the trajectory point on the road segment. Let the mapped trajectory point on the first segment $j_1 \in P$ be A and the point on the last segment $j_K \in P$ be B, where K is the total number of GPS points along path P. Let $\tilde{\ell}_1$ be the distance between A and the last point of j_1 and $\tilde{\ell}_K$ be the distance between B and the last point of j_K. We calculate the travel times estimate t_i as follows:

$$t_i = \frac{\ell_i T_P}{\tilde{\ell}_1 + \tilde{\ell}_K + \sum_{j=2}^{K-1} \ell_j}. \tag{4}$$

By taking into account the location of the first and last GPS point along the first and last link more carefully, the estimate (4) becomes more accurate compared to (3).

D. Gaussian Copula

An obvious choice for the travel time distribution might be Gaussian or lognormal distributions. However, as Fig. 3 shows for the most common trajectory, both distributions are poor fits for the empirical travel time data. Instead we applied Gaussian copulas, allowing to fit a more flexible distribution to the data. A Gaussian copula with p dimensions is defined as:

$$C(t_1, \ldots, t_p) = \Phi_p(\Phi^{-1}(t_1), \ldots, \Phi^{-1}(t_p); \Sigma), \tag{5}$$

where Φ is the CDF of standard normal distribution and $\Phi_p(*; \Sigma)$ is the CDF for a p-dimensional multivariate Gaussian distribution with zero mean and covariance matrix Σ.

We transform observed travel times t_i at link i, obtained by the scaling method (4), into Gaussian distributed values by the probability integral transform:

$$\tilde{t}_i = \psi^{-1}(F_i(t_i)), \tag{6}$$

where F_i is the distribution function of the travel times at link i, and ψ^{-1} is the inverse error function.

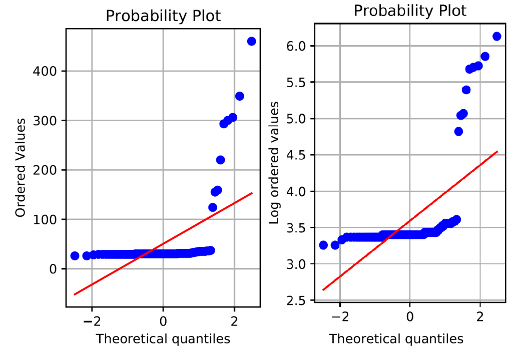

Fig. 3. QQ (quantile-quantile) plots of travel times and log travel times for the most traversed trajectory during one time interval.

The distribution F_i is inferred from a finite number of samples. In practice, we compute F_i by first calculating the cumulative histogram of the samples, and applying linear interpolation to the values of the cumulative histogram.

E. Empirical Covariance Matrix

In the proposed travel time modeling pipeline, we estimate the covariance of the travel times at the links in the transportation network. To this end, we compute the empirical covariance matrix S. Specifically, we compute the Partial Empirical Covariance Matrix (PECM) [11] by calculating covariances for every pair (i, j) of links separately:

$$\hat{S}_{i,j} = \beta_{i_j} \langle t_i t_j \rangle - \langle t_i \rangle \langle t_j \rangle, \tag{7}$$

where the $\langle t_i \rangle$ is the average travel time (obtained by the scaling method (3)) computed across all paths P that contain the link i, while the average $\langle t_i t_j \rangle$ is computed across all paths P that contain links i and j. To avoid the matrix being negative semi-definite, we introduce scaling coefficient $\beta_{i,j}$ so that all elements $\hat{S}_{i,j}$ have the same variance as proposed by Hunter *et al.* [11]:

$$\beta_{i,j} = \sqrt{\frac{\langle t_i^2 \rangle \langle t_j^2 \rangle}{\langle t_i^2 \rangle_{i,j} \langle t_j^2 \rangle_{i,j}}}, \tag{8}$$

where $\langle\rangle_{i,j}$ is computed across paths that contain both link i and link j. We set $\hat{S}_{i,j}$ to be zero if less than 5 trajectories contain these two links. Indeed if two links rarely occur together in the same path, most likely the travel times along those links are to a good approximation independent.

F. Graphical Lasso

In the setting of sparse GPS data, the amount of data is limited and the network is large. Therefore, it is hard to infer the full precision matrix reliably. For such scenarios, it is a common practice to assume that the precision matrix is sparse, and to apply the graphical lasso approach [16]:

$$\hat{K} = \underset{K}{\operatorname{argmin}} (\operatorname{Tr} SK - \log\det K + \alpha ||K||_1). \tag{9}$$

This procedure imposes an L_1 penalty, resulting in a sparse precision matrix $\hat{K}$. The inverse of the estimate $\hat{K}$ is a reliable estimate of the covariance matrix. We apply the coordinate descent implementation [16] to solve the optimization problem (9). To estimate the precision matrix, the algorithm is applied to PECM introduced in the previous section (7).

G. Bayesian Inference of High-Dimensional Sparse Networks

Up to this point, the model did not take into account the fact that some links are traversed by taxis more often than others. This affects the estimates of the covariances between links due to the difference in the uncertainty about the estimate. For example, if one link pair has a few simultaneous observations and another link pair has several times more observations, these covariance estimates should be treated differently when inferring the sparse precision matrix. To address this, we employ Bayesian Inference of High-Dimensional Sparse Networks framework (BISN) [15]. Instead of modeling precision matrix like the graphical lasso, BISN deals with LDU (lower-upper) decomposition of the precision matrix. In that case, the probability density function (PDF) of the Gaussian model with precision matrix $K = LDL^T$, where L is a lower triangle matrix and D is diagonal, can be written as:

$$p(t|L, D) = \prod_{j=1}^{p} D_{j,j} \exp(-\frac{1}{2} t^T LDL^T t). \tag{10}$$

BISN imposes a spike and slab prior to obtain sparse precision matrix K. The posterior distribution of the elements in the LDU decomposition are computed approximately by a variational Bayes approach. BISN introduces a coefficient γ for each element in K and treats γ as a random variable. This allows BISN to take into account the fact that different variables can have a different number of observations. In contrast with the graphical lasso, where the corresponding α parameter is a fixed constant multiplier for the matrix. As a result, BISN obtains more reliable estimates of the sparsity pattern of K, and better estimates of the non-zero elements in K, ultimately leading to a more reliable estimate of the covariance matrix. The algorithm only has quadratic computational complexity compared to the cubic complexity of graphical lasso; this substantial reduction in computational complexity is important for real-time applications. The computational complexity is reduced in BISN by applying the stochastic approximation based on matrix randomization.

Whereas the graphical lasso is applied to PECM, the BISN model directly infers the sparse precision matrix from the data (travel times at each link for different taxi rides). In order to apply BISN, we construct a matrix $X \in \mathbb{R}^{n \times p}$, where n is a number of observations (i.e., taxi trajectories) and p is a toyal number of links, and each element $X_{i,j}$ is an observed travel time for a link j. If a particular link j is not a part of the trajectory i, the corresponding value $X_{i,j}$ is missing.

The performance of the algorithm depends on several factors, including sample size and the amount of missing data. In experiments on the synthetic dataset, it was observed that the performance degrades significantly with the increase of missing data percentage. Based on experiments with synthetic data, no covariance structure could be recovered with more than 60% of the values missing.

To control the amount of the missing data in X we make the following assumption: different taxis can be treated as one vehicle during a short time interval. Under this assumption, we can collapse rows of X to reduce the proportion of missing values. We collapse rows $i_1 \dots i_t$ if start times of all observed trajectories in these rows are within two minutes interval, and none of the links occur in more than one trajectory, and hence each column has at most one non-missing element across rows $i_1 \dots i_t$.

We apply BISN in two different ways. In the first case, we construct matrix X for the 50 trajectories simultaneously. In this way, the number of columns p equals the number of unique links across 50 trajectories. The algorithm will be evaluated only once for the whole network. We refer to this approach as the "BISN network model".

In the second approach, which we will refer to as "BISN path model", we construct a matrix X^P for each path P separately. The matrix X^P is obtained as a submatrix of X by selecting columns associated with links that occur in P.

H. Proposed Model

Next, we describe the overall proposed model. We compute a set of travel time values for every link in the network by the scaling method (4). Next, we calculate the empirical covariance matrix (PECM) S and apply the graphical lasso algorithm. As an alternative, we apply BISN to the data matrix X (network model) and X^P (path model). Both the graphical lasso and BISN approach lead to sparse precision matrices. The covariance matrices $\hat{\Sigma}$ in the Gaussian models are computed as the inverse of those sparse precision matrices. For the copula Gaussian models, we apply the same calculations to the integral-transformed travel times (cf. (9)), resulting in covariance matrices $\hat{\Sigma}^{\mathcal{C}}$. The Gaussian model is offered as a comparison, however, the main proposed model is copula Gaussian. We obtain multivariate Gaussian models TT_P and copula Gaussian models $TT_P^{\mathcal{C}}$ for the travel times:

$$\begin{aligned} TT_P &\sim \mathcal{N}(\begin{bmatrix} t_1 & \dots & t_{|P|} \end{bmatrix}, \hat{\Sigma}) \\ TT_P^{\mathcal{C}} &\sim \mathcal{CN}(\begin{bmatrix} \tilde{t}_1 & \dots & \tilde{t}_{|P|} \end{bmatrix}, \hat{\Sigma}^{\mathcal{C}}), \end{aligned} \tag{11}$$

where t_i are link travel times from equation (4), $\tilde{t}_i$ are travel times transformed with the integral transformation (8) and $\hat{\Sigma}$ is the covariance matrix obtained by graphical lasso or BISN. Further reading on the transformation of non-Gaussian variables and Gaussian copulas can be found in [33].

From those models, we generate samples of path travel times as follows. First, we draw samples from the link travel time distributions (3). Specifically, we sample from the (multi-variate) marginal distribution associated with the links contained in the path. In other words, we do not sample from the individual marginal distributions associated with each link, which would be inappropriate, but from the joint travel time distribution instead of all links in the path, which is a marginal distribution of the models (3). For the Gaussian models, we can generate such samples via a standard procedure: first, we sample a vector z of independent standard normal variables, and then perform spectral decomposition of Σ : $\Sigma = U \Lambda U^\top$. Then, the desired samples x can be calculated

TABLE II

PERFORMANCE METRICS FOR DIFFERENT GAUSSIAN AND COPULA GAUSSIAN MODELS FOR 50 TRAJECTORIES. LEFT AND RIGHT OF /, RESULTS ARE SHOWN FOR GAUSSIAN AND COPULA GAUSSIAN MODELS RESPECTIVELY

Metric	Independent	PECM	Neighbors	Glasso + PECM	BISN (network)	BISN (path)
Mean of KL	1.06 / 0.80	0.95 / 0.72	1.02 / 0.78	0.99 / 0.73	0.98 / 0.74	0.90 / **0.69**
Std of KL	0.18 / 0.17	0.14 / 0.15	0.14 / 0.15	0.15 / 0.15	0.17 / 0.16	0.15 / 0.18
Mean of Hellinger	0.78 / 0.519	0.57 / 0.511	0.60 / 0.513	0.55 / 0.506	0.57 / 0.510	0.54 / **0.504**
Std of Hellinger	0.1 / 0.09	0.09 / 0.08	0.1/ 0.08	0.09/ 0.08	0.09 / 0.08	0.09 / 0.08

as $x = \mu + Az$, where $A = U\Lambda^{\frac{1}{2}}$ and μ is the mean vector. To sample from the copula Gaussian models, we apply the inverse transform to the Gaussian samples. At last, we obtain samples for the path travel time by summing the samples of the link travel times, leading to a sample of the path travel time for each sample of the multivariate link travel time distribution.

I. Evaluation

We evaluate the travel time models by comparing them to the empirical pathwise travel time distributions. To assess the deviation between the models and empirical distributions, we compute the Kullback Leibler divergence:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx, \tag{12}$$

where Q is the distribution generated from the models, and P is the empirical distribution. We can describe Q analytically. For trajectory P, let $t_1 \dots t_{|P|}$ be the average travel times for each link in the trajectory and $\hat{\Sigma} = \{\hat{\sigma}_{i,j}\}$ be the estimated covariance matrix. Then, for the Gaussian models, the distribution of travel times for trajectory P can be described as:

$$TT_P \sim \mathbb{N}(\sum_{i=1}^{|P|} t_i, \sum_{i=1}^{|P|}\sum_{j=1}^{|P|} \hat{\sigma}_{i,j}). \tag{13}$$

However, there are no analytical forms for the copula models and the empirical distribution. Therefore, we consider the formula of Kullback Leibler divergence for discrete distributions:

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}, \tag{14}$$

where $P = \sum_i P(i)$ and $Q = \sum_i Q(i)$ are two discrete probability distributions. To apply this formulation we first discretize both distributions by computing histograms. Due to the presence of $\log \frac{P(i)}{Q(i)}$ in the formula, discrete KL divergence is sensitive to discretization. We compute the KL divergence by means of histograms, and tested various numbers of histogram bins. We use a fixed number of uniform bins for each histogram, collapsing them when (14) is undefined. To determine the number of bins for our experiments, we used a maximum between two standard estimators, namely Sturges method and Freedman-Diaconis rule. We set the default number of bins to 11, as it was the most common number of bins computed by the estimator across different inputs.

Despite choosing an appropriate number of bins, the numerical evaluation of the KL divergence may lead to instabilities due to the ratio in (14). Therefore, we also considered an alternative, i.e., the Hellinger distance:

$$H(P, Q) = \frac{1}{\sqrt{2}}\sqrt{\sum_i (\sqrt{P(i)} - \sqrt{Q(i)})^2}. \tag{15}$$

It does not require computing the ratio of probabilities hence it avoids the mentioned problem of the KL divergence. The Hellinger distance for discrete distributions is similar to the root mean square difference between the square roots of vectors.

IV. RESULTS AND DISCUSSION

After obtaining trajectories by map-matching we split the whole dataset into 1-hour intervals. We chose intervals of 1 hour in order to have sufficient data points for evaluation. We are dealing with sparse trajectories so that a particular path (which consists of several links) will be traversed by a limited number of probe vehicles. We need to consider a larger time interval to be able to compute the empirical distribution. This is not a limitation of the proposed method, but a limitation of the available dataset; for shorter intervals there are fewer trajectories with sufficient coverage for the evaluation. For each of the 50 paths in the test set, we consider only travel times from exact trajectories to be able to compare our model with the ground truth. However, the proposed method does not require full 1-hour data for training which follows from the results in later sections (see Fig. 6), but the evaluation is again done for the 1-hour intervals.

For each time period, we randomly select 70% of the trajectories for training the travel time models. We compare the following models:

- Covariance matrix is diagonal (all links are independent) [Independent],
- PECM is used as a covariance matrix [PECM],
- PECM with values for non-neighboring links set to zero [Neighbors],
- Graphical lasso applied to PECM [Glasso],
- BISN applied to the entire network [BISN (network)],
- BISN applied separately for each path [BISN (path)].

It is possible to transform travel times into Gaussian variables as a preprocessing step with any of the above methods.

A. Gaussian Copulas

First, we study the importance of using Gaussian copulas. It is clear from Table II that copula models have lower KL divergence and hence describe the travel time data more accurately. This can be explained by the fact that observed path

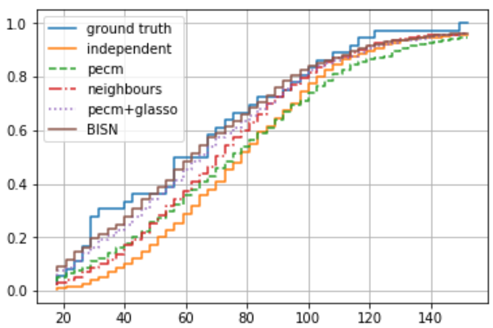

Fig. 4. Travel time distributions for one random path.

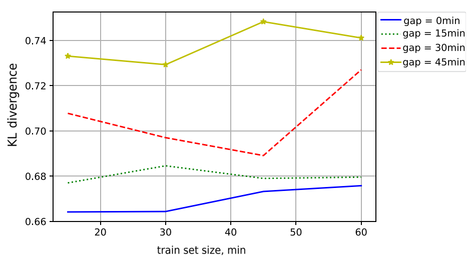

Fig. 5. KL divergence for BISN as a function of the length of the training data [min], and for different gaps between the training and test period.

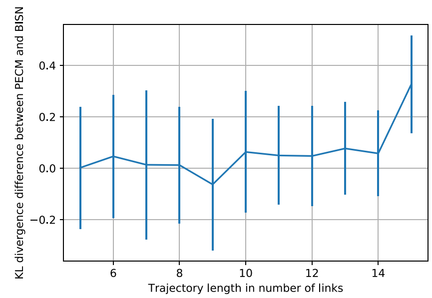

Fig. 6. Difference in KL divergence between BISN and PECM models for various path lengths. Error bars represent the standard deviation of the difference.

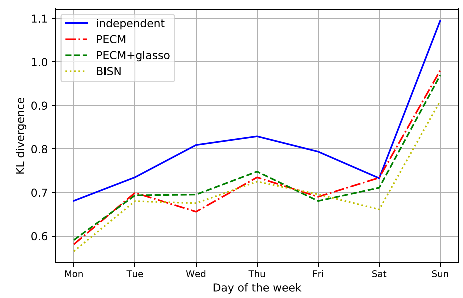

Fig. 7. Performance of four models depending on the day of the week.

travel times do not necessarily follow the Gaussian distribution (see Fig. 3).

B. Graphical Lasso and BISN

Fig. 4 shows empirical and estimated cumulative distribution functions (CDF) for one of the paths. For all methods, travel times were first transformed into Gaussian random variables. It can be observed that the shape of the BISN CDF is the closest to the empirical CDF.

To compare the graphical lasso algorithm and BISN with baseline methods we analyze the dataset containing 4 full weeks (28 days) of data. Average KL divergence and Hellinger distance for the 50 most traversed trajectories (for which a sufficient number of observations is available) are shown in Table II. It can be seen that the graphical lasso and BISN outperform the baseline methods and BISN (path) performs best. The Independent method performs worst as expected since it doesn't take into account the dependence between links. BISN (network) performs worse than the graphical lasso, possibly due to the inability to accurately estimate large precision matrices with very limited data. Given such performance of BISN (network), all of the following experiments are performed only for BISN (path) which is later referred to as BISN. PECM and the graphical lasso achieve comparable KL divergence values, however, the Hellinger distance is smaller for the graphical lasso.

Next, we investigate the performance of the proposed models on the future time intervals. To this end, we train the model using the data preceding the 1-hour test interval. We compare different training set sizes (15, 30, 45 and 60 minutes) as well as different gaps between training and test sets (0, 15, 30 and 45 minutes). The results are shown in Fig. 5. It follows that larger gaps between training and test times result in larger errors. At the same time, increasing the duration of the training data beyond 30 min is not effective, as relatively old data is then included in the training dataset. The traffic conditions may have changed in the meantime.

We also investigate how the models perform for trajectories of different lengths. Fig. 6 shows that BISN method performs better especially for longer trajectories. This can be explained by the fact that the BISN approach is able to model the covariance structure better which plays a bigger role for trajectories with more links.

We also found that there exist temporal patterns affecting model performance across different times of the day (see Fig. 7). There is an increase in error during the peak hours. Both the KL divergence and the Hellinger distance show similar patterns. We also observed performance patterns across different days of the week, with Sunday having the largest error.

C. Computational Complexity

Travel time estimation is a real-time task, hence the computational complexity is an important factor. Both the graphical lasso algorithm and BISN require intensive computations. The original formulation of the graphical lasso [16] requires $\mathcal{O}(p^3)$ operations, where p is the number of variables (road links). On the other hand, BISN has quadratic instead of cubic complexity, since it relies on stochastic approximations by matrix randomization. However, BISN complexity is also dependent on the amount of missing data. The running time of these algorithms is compared in Table III. Experiments

TABLE III

AVERAGE PROCESSING TIME FOR ONE PATH FOR ONE TIME PERIOD IN SECONDS

Features	Computation time [s]	Algorithm	Computation time [s]	Algorithm	Computation time [s]
Independent	0.522	**Glasso**	5.338	**PECM**	0.716
BISN (path)	5.926	**Neighbors**	0.734	**BISN (network)**	49.88

were performed on a server with Intel Xeon CPU E5-2690v2 @ 3.00 GHz and 32 GB of RAM. The table shows the average processing time including the necessary preprocessing. It should be noted, however, that the performance of both graphical lasso and BISN heavily depends on the algorithm parameters. We set the α parameter for the graphical lasso to 10^{-4} (based on preliminary tuning), maximum iterations as 10^3 and 10^4 for the graphical lasso and BISN respectively, and we set the convergence tolerance to 10^{-4} for both methods.

D. Discussion

We performed several experiments aimed at estimating both current and future states of the road network. Overall, the results show that it is possible to obtain reliable estimates of path travel time distributions using covariance estimation techniques such as the graphical lasso and BISN even on a dataset with a low sampling interval of 1 minute. Fig. 4 shows that the models can accurately estimate the shape of the distribution for a path. Moreover, we observed that BISN on average performs better than the common graphical lasso method (cf. Table II). This table also provides a performance comparison with several baseline methods such as the independent travel times model and the empirical covariance matrix. It also includes results for non-copula models. We confirmed that real-world travel time distributions are typically non-Gaussian (cf. Fig. 3), and can be modeled by Gaussian copulas. We investigated the ability of the proposed model to estimate travel time distributions for future time intervals. As can be seen from Table 5, due to its reasonable computational time, the proposed model can be applied in real-time settings.

For example, the model can be applied to predict travel time distributions based on origin-destination pairs. Previous approaches for path and time inference (such as [25]) can be improved by relaxing assumptions on the distributions, while Fig. 6 shows that the performance on future time intervals can be comparable to estimating the current distributions. The proposed model can also be utilized for notifying travelers about changes in the traffic situation. By updating the current travel time estimates when the new data becomes available, the model would be able to compute expected travel times together with quantiles and other statistics.

We investigated the computation complexity of both graphical lasso and BISN. Despite the theoretical advantage, in our experiments, BISN showed slightly worse running times compared to the graphical lasso. This can be explained by the extreme sparsity of the dataset, since BISN requires more computational time in the presence of missing values. This means that for certain other datasets (with a different structure of sparsity) BISN can become less computationally tractable for real-time use.

V. CONCLUSION AND FUTURE WORKS

In this work, we explored several approaches to modeling travel time distributions by combining Gaussian copulas, the graphical lasso method, and the Bayesian Inference of Sparse Networks (BISN) method. By analyzing GPS data from the Singapore road network, we observed that Gaussian and lognormal distributions are poor approximations of travel time distributions. As an alternative to those standard distributions, we investigated Gaussian copulas. The covariance matrices in those copula models allow us to capture the statistical dependence of travel times across different links. In order to obtain reliable estimates of the covariance matrices from the sparse GPS data, we applied the graphical lasso and, as an alternative, the recently proposed BISN approach. We compared BISN approach with the several baseline models. Numerical results show that the proposed framework yields more reliable travel time models compared to baseline models in terms of KL divergence and Hellinger distance. In our future work, we will build upon these results as part of stochastic routing decision-making.

REFERENCES

[1] S. Samaranayake, S. Blandin, and A. Bayen, "A tractable class of algorithms for reliable routing in stochastic networks," *Transp. Res. C, Emerg. Technol.*, vol. 20, no. 1, pp. 199–217, 2012.

[2] S. Ma, Y. Zheng, and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *Proc. IEEE 29th Int. Conf. Data Eng. (ICDE)*, Apr. 2013, pp. 410–421.

[3] T. Choe, A. Skabardonis, and P. Varaiya, "Freeway performance measurement system: Operational analysis tool," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1811, no. 1, pp. 67–75, 2002.

[4] M. Ben-Akiva, J. Bottom, and M. S. Ramming, "Route guidance and information systems," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 215, no. 4, pp. 317–324, 2001.

[5] D. H. Nam and D. R. Drew, "Traffic dynamics: Method for estimating freeway travel times in real time from flow measurements," *J. Transp. Eng.*, vol. 122, no. 3, pp. 185–191, 1996.

[6] J. Long, Z. Gao, and W. Y. Szeto, "Discretised link travel time models based on cumulative flows: Formulations and properties," *Transp. Res. B, Methodol.*, vol. 45, no. 1, pp. 232–254, 2011.

[7] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004.

[8] D. Billings and J.-S. Yang, "Application of the ARIMA models to urban roadway travel time prediction—A case study," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2006, pp. 2529–2534.

[9] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transp. Res. C, Emerg. Technol.*, vol. 11, nos. 3–4, pp. 187–210, 2003.

[10] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 25–34.

[11] T. Hunter *et al.* (2013). "Arriving on time: Estimating travel time distributions on large-scale road networks." [Online]. Available: https://arxiv.org/abs/1302.6617

[12] K. Wan, "Estimation of travel time distribution and travel time derivatives," Ph.D. dissertation, Princeton Univ., Dept. Oper. Res. Financial Eng., Princeton, NJ, USA, 2014.

[13] K. Wan and A. L. Kornhauser, "Link-data-based approximation of path travel time distribution with Gaussian copula estimated through LASSO," in *Proc. Transp. Res. Board 89th Annu. Meeting*, 2010. [Online]. Available: https://orfe.princeton.edu/~alaink/Papers/AggregationforPathtravelTimeShortv7.pdf

[14] A. Prokhorchuk, V. P. Payyada, J. Dauwels, and P. Jaillet, "Estimating travel time distributions using copula graphical lasso," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Tokyo, Japan, Oct. 2017, pp. 1–6.

[15] H. Yu, S. Wu, and J. Dauwels, "Fast Bayesian inference of sparse networks with automatic sparsity determination," 2019.

[16] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Dec. 2007.

[17] J. Aslam, S. Lim, X. Pan, and D. Rus, "City-scale traffic estimation from a roving sensor network," *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2012, pp. 141–154.

[18] D. Park, L. R. Rilett, and G. Han, "Spectral basis neural networks for real-time travel time forecasting," *J. Transp. Eng.*, vol. 125, no. 6, pp. 515–523, 1999.

[19] D. Park and L. R. Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network," *Comput. Aided Civil Infrastruct. Eng.*, vol. 14, no. 5, pp. 357–367, Sep. 1999.

[20] D. J. Dailey, "Travel time estimates using a series of single loop volume and occupancy measurements," in *Proc. Transp. Res. Board 76th Annu. Meeting*, Washington, DC, USA, Jan. 1997, pp. 1–14.

[21] S. Blandin, L. El Ghaoui, and A. Bayen, "Kernel regression for travel time estimation via convex optimization," in *Proc. 48th Jointly With 28th Chin. Control Conf. IEEE Conf. Decis. Control (CDC)*, Dec. 2009, pp. 4360–4365.

[22] B. S. Westgate, D. B. Woodard, D. S. Matteson, and S. G. Henderson, "Large-network travel time distribution estimation for ambulances," *Eur. J. Oper. Res.*, vol. 252, no. 1, pp. 322–333, 2016.

[23] E. Jenelius and H. N. Koutsopoulos, "Travel time estimation for urban road networks using low frequency probe vehicle data," *Transp. Res. B, Methodol.*, vol. 53, pp. 64–81, Jul. 2013.

[24] M. Ramezani and N. Geroliminis, "On the estimation of arterial route travel time distribution with Markov chains," *Transp. Res. B, Methodol.*, vol. 46, no. 10, pp. 1576–1590, 2012.

[25] T. Hunter, P. Abbeel, R. Herring and A. Bayen, "Path and travel time inference from GPS probe vehicle data," *NIPS Anal. Netw. Learn. Graphs*, vol. 12, no. 1, pp. 1–8, 2009.

[26] X. Zhan, S. Hasan, S. V. Ukkusuri, and C. Kamga, "Urban link travel time estimation using large-scale taxi data with partial information," *Transp. Res. C, Emerg. Technol.*, vol. 33, pp. 37–49, Aug. 2013.

[27] W. Pu, J. Lin, and L. Long, "Real-time estimation of urban street segment travel time using buses as speed probes," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2129, no. 1, pp. 81–89, 2009.

[28] Y. Feng, J. Hourdos, and G. A. Davis, "Probe vehicle based real-time traffic monitoring on urban roadways," *Transp. Res. C, Emerg. Technol.*, vol. 40, pp. 160–178, Oct. 2014.

[29] C. Goh, J. Dauwels, N. Mitrovic, M. T. Asif, A. Oran, and P. Jaillet, "Online map-matching based on hidden Markov model for real-time traffic sensing applications," in *Proc. 15th Int. IEEE Conf. Intell. Transp. Syst.*, Sep. 2012, pp. 776–781.

[30] I. Sanaullah, M. Quddus, and M. Enoch, "Developing travel time estimation methods using sparse GPS data," *J. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 532–544, 2016.

[31] J. Miller, S.-I. Kim, M. Ali, and T. Menard, "Determining time to traverse road sections based on mapping discrete GPS vehicle data to continuous flows," in *Proc. IEEE Intell. Vehicles Symp.*, San Diego, CA, USA, Jun. 2010, pp. 615–620.

[32] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transp. Res. C, Emerg. Technol.*, vol. 15, no. 5, pp. 312–328, 2007.

[33] H. Yu, J. Dauwels, and X. Wang, "Copula Gaussian graphical models with hidden variables," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 2177–2180.

Anatolii Prokhorchuk received the bachelor's degree in applied mathematics and computer science from the Moscow Institute of Physics and Technology, Russia, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical and Electronic Engineering, Nanyang Technological University. He is also a Graduate Fellow of the Singapore-MIT Alliance for Research and Technology. His research interests include machine learning and intelligent transportation systems.

Justin Dauwels received the Ph.D. degree in electrical engineering from the Swiss Polytechnical Institute of Technology, Zurich, in 2005. He was a Post-Doctoral Fellow at the RIKEN Brain Science Institute (2006–2007), and a Research Scientist at the Massachusetts Institute of Technology (2008–2010). He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. He also serves as the Deputy Director of the ST Engineering NTU Corporate Lab, which comprises of over 100 Ph.D. students, research staff, and engineers, developing novel autonomous systems for airport operations and transportation. His research interests are in data analytics with applications to intelligent transportation systems, autonomous systems, and analysis of human behavior and physiology. His research on intelligent transportation systems has been featured by the BBC, Straits Times, Lianhe Zaobao, Channel 5, and numerous technology websites. Besides his academic efforts, his team also collaborates intensely with local start-ups, SMEs, and agencies, in addition to MNCs, in the field of data-driven transportation, logistics, and medical data analytics. He is a JSPS Post-Doctoral Fellow (2007), a BAEF Fellow (2008), a Henri-Benedictus Fellow of the King Baudouin Foundation (2008), and a JSPS invited Fellow (2010 and 2011).

Patrick Jaillet received the Ph.D. degree in operations research from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1985. He is currently the Dugald C. Jackson Professor with the Department of Electrical Engineering and Computer Science, and also the Co-Director of the Operations Research Center, Massachusetts Institute of Technology. His research interests include algorithm design and analysis for on-line problems, real-time and dynamic optimization, network design and optimization, and probabilistic combinatorial optimization.