

Estimating urban traffic states using iterative refinement and Wardrop equilibria

ISSN 1751-956X
 Received on 5th January 2018
 Revised 3rd April 2018
 Accepted on 8th May 2018
 doi: 10.1049/iet-its.2018.0007
www.ietdl.org

Weizi Li¹✉, Meilei Jiang², Yaoyu Chen², Ming C. Lin¹

¹Department of Computer Science, University of North Carolina at Chapel Hill, 201 S. Columbia St., Chapel Hill, NC, USA

²Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, 318 Hanes Hall, Chapel Hill, NC, USA

✉ E-mail: weizili@cs.unc.edu

Abstract: Traffic has become a major problem in metropolitan areas across the world. It is critical to understand the complex interplay of a road network and its traffic states so that researchers and planners can improve the city planning and traffic logistics. The authors propose a novel framework to estimate urban traffic states using GPS traces. Their approach begins with an initial estimation of network travel times by solving a convex optimisation programme based on Wardrop equilibria. Then, they iteratively refine the estimated network travel times and vehicle traversed paths. Lastly, using the refined results as input, they perform a nested optimisation process to derive traffic states in areas without data coverage to obtain full traffic estimations. The evaluation and comparison of their approach over two state-of-the-art methods show up to 96% relative improvements. In order to study urban traffic, the authors have further conducted field tests in Beijing and San Francisco using real-world GIS data, which involve 128,701 nodes, 148,899 road segments, and over 26 million GPS traces.

1 Introduction

Traffic has become a serious topic in metropolitan areas across the world. The extra cost due to traffic congestion and accidents is assessed over 1 trillion dollars worldwide. As the situation continues to deteriorate due to rapid urbanisation and growth in vehicle production [1], an intelligent system that can understand the complex interplay of a road network and its traffic states has been demanded in many contexts, including analysing urban infrastructure [2], understanding human mobility [3], and designing better routing strategies [4]. In addition, being able to accurately estimate traffic states is necessary for navigation and planning of autonomous vehicles [5], which have been tested on public roads in many countries and hold the promise to revolutionise transportation industry.

In the present, mobile sensor data set is one of the most effective sources for estimating citywide traffic states attributing to its ubiquity [6]. However, there are several limitations of GPS data preventing the immediate usage. One such feature is *low-sampling rate*, i.e. there exists a large time gap (e.g. >60 s) between two consecutive GPS reports. Another feature is *spatial-temporal sparsity*, i.e. data are scarce in certain areas and time periods. These features give rise to several challenges in referencing traffic states via GPS data. To begin with, GPS points need to be mapped onto a road network and traversed paths need to be inferred. This is difficult given that in a complex urban environment, multiple paths could connect two GPS points. Next, after a traversed path is determined, the aggregate travel times (i.e. GPS timestamp differences) need to be decomposed and distributed to individual road segments. Lastly, in order to achieve a full estimation, the missing traffic measurements in certain areas and time periods require interpolation.

The aforementioned issues are addressed using *map-matching*, *travel-time inference*, and *missing-value completion*, respectively. These processes, however, are commonly executed *in tandem* and result in possible cascading errors in the estimation. To be specific, many map-matching approaches have adopted the shortest-distance criterion [7–9] to infer the traversed path. However, this assumption can lead to a considerable bias in a congested network where the shortest-distance path may differ from the shortest travel-time path [10–12] and the latter is more likely to be selected by GPS devices and experienced drivers. As a result, in a sequential pipeline, such a bias will affect the following

procedures. In addition, while there exists abundant research on missing-value completion, the interpolation of spatial missing values in data-lacking areas can be improved. Existing approaches commonly take a static data mining perspective. However, the dynamic aspects of traffic such as flow characteristics can be incorporated to enhance estimation accuracy.

We propose a novel framework for estimating traffic states at city scale using sparse GPS data. In a discretised time interval, we first obtain a coarse inference of travel times on road segments with GPS data coverage by solving a convex optimisation program inspired by Wardrop equilibria [13]. Next, we refine the inferred traffic states and vehicle paths via *iteratively* performing map-matching and travel-time inference. In order to handle spatial sparsity, we embed the refined traffic states into a nested optimisation procedure [14], which ensures certain traffic flow characteristics. Finally, by considering all time intervals, we address temporal sparsity via a compressed sensing-based algorithm [12]. The schematic diagram of our framework is shown in Fig. 1 and our contributions are listed as follows:

- a convex optimisation formulation that models similarity of traffic patterns among adjacent roads and provides a coarse inference of network travel times;
- an iterative refinement process that alternates between map-matching and travel-time inference to mitigate errors in both procedures and simultaneously infer vehicle paths and network travel times;
- a novel perspective that incorporates estimated traffic flows via GPS data into nested optimisation to obtain metropolitan-scale traffic estimations.

We evaluate our framework using a real road network that consists of 5407 nodes and 1612 road segments, 34 heuristic network travel times corresponding to various congestion levels and times of a day, and over 10 million synthetic GPS traces. The effectiveness of our approach has been compared to state-of-the-art methods, namely Hunter [11] and Rahmani *et al.* [15], resulting in up to 96% relative improvements. In order to study real-world traffic, we have conducted field tests in Beijing and San Francisco using actual GIS data sets, which include 128,701 nodes, 148,899 road segments, and over 26 million GPS traces.

The poster version of this paper appeared in [16]: this is the full version. The rest of the paper is organised as follows. We survey

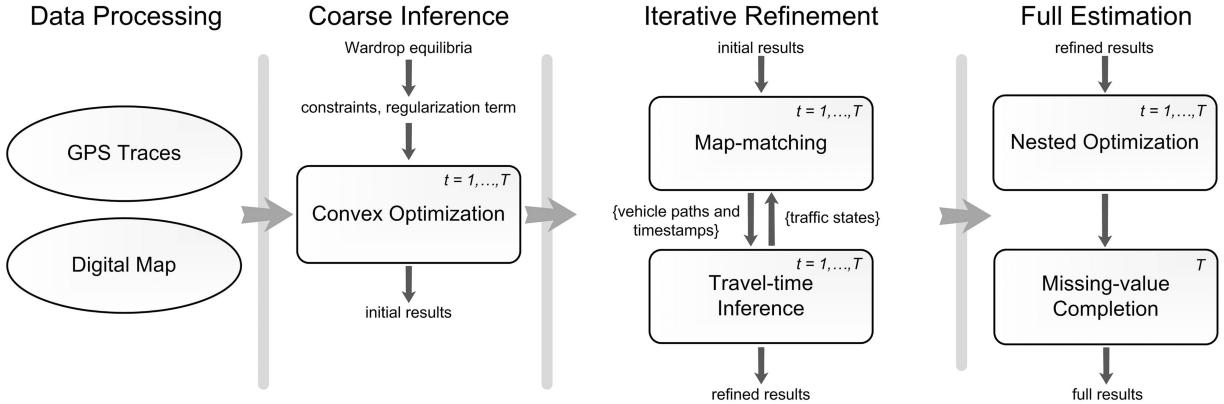


Fig. 1 Overview of our system: convex optimisation coupled with Wardrop equilibria is used to derive initial results; refined results are obtained through joint map-matching and travel-time inference; the nested optimisation is applied to individual discretised time intervals for estimating a citywide traffic condition; missing-value completion is performed over all time intervals to obtain final results

related work in Section 2. We discuss background information in Section 3 and our approach in detail in Section 4. We provide evaluation and comparison of our approach to existing methods in Section 5. We present the results of field tests in Section 6. Finally, we conclude in Section 7 with the discussion of future work.

2 Related work

The estimation of traffic states has received increasing attention over the past decades. In order to improve the estimation accuracy, researchers have adopted a variety of data types and simulation models [17, 18]. While these methods have achieved significant results, their usage is mostly restricted to highway segments where stationary sensors are available. In order to estimate urban traffic conditions, existing studies have resorted to GPS data. Nevertheless, because of the inherent noise and spatial-temporal sparsity, multiple steps, namely *map-matching*, *travel-time inference*, and *missing-value completion*, are required and commonly assembled into a sequential pipeline to derive an estimation [6, 19–21].

As the first procedure, *map-matching* handles noisy GPS points by mapping them back to a road network and inferring the most likely traversed path of a vehicle. Many techniques have adopted the shortest-distance criterion by treating the shortest-distance path between two consecutive GPS points as the traversed path [7–9, 22]. This assumption, however, leads to possible errors in a busy road network where the shortest-distance path and the shortest travel-time path may differ [10–12]. The reason being, in a congested network, GPS devices and drivers may prefer the shortest-travel time path instead. This observation also serves as the foundation for Wardrop equilibria [13]. Acknowledging a vehicle can drive on a slower route, a better approach would be comparing GPS timestamps with the network travel times. However, being the first step in a sequential pipeline, map-matching is usually performed on an empty road network where travel times are unknown, which results in potential mapping errors.

After map-matching, a collection of map-matched vehicle paths and their corresponding GPS timestamps is generated. Owing to a low-sampling rate, in a complex urban environment, a map-matched path can span multiple road segments and the difference in its GPS timestamps needs to be allocated to individual road segments. This procedure is known as *travel-time inference*. To list few examples of this subject, Wang *et al.* [21] use a tensor-based decomposition approach to infer travel times of a road network. Hellinga *et al.* [23] have proposed an analytical solution based on observations of real-world traffic. Rahmani *et al.* [15] take a non-parametric approach, while Herring *et al.* [24] and Hofleitner *et al.* [42] both adopt probabilistic techniques. Given many proposed algorithms are effective, they are usually complex in order to compensate errors produced in the prior step (i.e. map-matching).

As the third procedure, *missing-value completion* has also been studied to various extents. For example, tensor-based approaches [21, 25] were developed to explore shared characteristics of road

segments in close proximity. In [12, 26], researchers have interpolated missing values via compressed sensing-based algorithms. Despite recent advances, a systematical framework that incorporates traffic flow characteristics and is able to provide estimations at city scale and over an entire traffic period is demanded [27].

There are many other studies related to our work. Traffic states are formed by individual vehicles and affected by road infrastructures such as intersections. Therefore, by having the information of individual vehicles/drivers [28, 29] and their responses to various road designs [30, 31], better estimations can be achieved. Additionally, the traffic estimation in road networks resembles the corresponding efforts in communication networks, especially considering the inference of origin–destination (O–D) flows and traffic matrix that describe network behaviours and status [32–35].

There are three key differences between our work and previous studies. First, for addressing low-sampling-rate data, we use an iterative refinement rather than a sequential computation to reduce the errors generated during map-matching and travel-time inference. Second, for addressing the spatial sparsity, we treat probe vehicles as mobile traffic sensors. By incorporating the sensing results into a large number of probe vehicles into the traffic assignment program (explained in Section 3), we can compute travel times and flows of all road segments in a network, including those are not covered by GPS data. Third, our approach relies heavily on transportation engineering studies, which allows the traffic modelling one step closer to real-world traffic.

3 Preliminaries

We represent a road network as a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$, where \mathcal{E} represents road segments and \mathcal{V} the end points of the road segments. Based on census trip information, a city can be separated into geographical units known as traffic analysis zones (TAZs) (see Fig. 2 Middle for an example). In transportation planning, the centres of TAZs are treated as locations where the traffic flow departs and arrives. The former are termed *origins* $\mathcal{O} \subseteq \mathcal{V}$ and the latter are termed *destinations* $\mathcal{D} \subseteq \mathcal{V}$. Traffic flow is typically observed between every origin–destination pairs (O–D pairs). We denote the average flow from $r \in \mathcal{O}$ to $s \in \mathcal{D}$ over a certain time interval (e.g. 1 h) as u_{rs} . Usually, there exist multiple paths connecting an origin and a destination in an urban environment. By denoting all paths from r to s as J_{rs} and the flow on a single path $k \in J_{rs}$ as u_{rs}^k , u_{rs} can then be computed as follows:

$$u_{rs} = \sum_{k \in J_{rs}} u_{rs}^k, \quad u_{rs}^k \geq 0, \quad \forall r \in \mathcal{O}, s \in \mathcal{D}. \quad (1)$$

Differing from the definition of a path flow, the flow of a road segment $f_{e \in \mathcal{E}}$ is the aggregation of all path flows traversing e .



Fig. 2 Left: Sample GPS points from the Cabspotting data set [44]. Middle: Road maps of downtown San Francisco overlaid with TAZs. Right: Heuristic network travel times (converted to VOC) established via the timestamp model

$$f_e = \sum_r \sum_s \sum_{k \in J_{rs}} \delta_e^k u_{rs}^k, \quad \forall r \in \mathcal{O}, s \in \mathcal{D}, \quad (2)$$

where $\delta_e^k \in \{0, 1\}$ indicates whether k contains e . By arranging all O-D pairs in $\mathbf{u} = [\dots, u_i, \dots]^T$ and their assignment proportions on e as $\mathbf{P}_e = [\dots, p_{ei}, \dots]$, we have $f_e = \mathbf{P}_e \mathbf{u}$. More compactly, by denoting all road-segment flows as $\mathbf{f} = [\dots, f_e, \dots]_{e \in \mathcal{E}}^T$ and their assignment proportions for each e as $\mathbf{P} = [\dots, \mathbf{P}_e, \dots]_{e \in \mathcal{E}}^T$, we obtain the general form of road-segment flows in a network as $\mathbf{f} = \mathbf{P}\mathbf{u}$, where \mathbf{P} is termed *assignment matrix*.

In principle, providing \mathbf{u} , we can compute travel times and flows of all road segments by solving the *traffic assignment* problem. One commonly used criterion is *user equilibrium* [13], under which the traffic assignment problem takes the following form:

$$\begin{aligned} \text{minimise}_{\mathbf{u}} \quad & z(\mathbf{f}) = \sum_{e \in \mathcal{E}} \int_0^{f_e} t_e(\omega) d\omega, \\ \text{subject to} \quad & u_{rs} = \sum_{k \in J_{rs}} u_{rs}^k, \quad \forall r \in \mathcal{O}, s \in \mathcal{D}, \\ & f_e = \sum_r \sum_s \sum_{k \in J_{rs}} \delta_e^k u_{rs}^k, \quad \forall e \in \mathcal{E}, \\ & u_{rs} \geq 0 \quad \forall r \in \mathcal{O}, s \in \mathcal{D}, \end{aligned} \quad (3)$$

where $t_e = t_e(f_e)$ is the travel time of the road segment e .

Traditionally, O-D pairs (i.e. \mathbf{u}) are obtained via a large-scale survey which is conducted infrequently due to prohibitive cost. Recent approaches are taking flow measurements provided by monitoring sensors (e.g. loop detectors and video cameras) into computation [36]. To illustrate the idea formally, we denote the *target O-D pairs* from survey data as $\bar{\mathbf{u}}$, the set of road segments with installed traffic sensors as \mathcal{A} , and the flow measurements on these road segments as $\bar{\mathbf{f}} = [\dots, \bar{f}_a, \dots]_{a \in \mathcal{A}}^T$. Then, we can compute the *estimated O-D pairs* $\hat{\mathbf{u}}$ through estimated road-segment flows $\hat{\mathbf{f}}$ by solving the following formula [37]:

$$\begin{aligned} \text{minimise}_{\hat{\mathbf{u}}} \quad & \mathcal{F}_1(\hat{\mathbf{u}}, \bar{\mathbf{u}}) + \mathcal{F}_2(\hat{\mathbf{f}}, \bar{\mathbf{f}}), \\ \text{subject to} \quad & \hat{\mathbf{f}} = \mathcal{M}(\hat{\mathbf{u}}), \\ & \hat{\mathbf{u}} \geq 0, \end{aligned} \quad (4)$$

where \mathcal{F}_1 and \mathcal{F}_2 represent distance functions, and \mathcal{M} is termed *assignment map*. If \mathcal{M} takes the form of (3), then (4) becomes a nested optimisation scheme [14], in which the upper level tries to minimise the differences between existing data and estimates, while the lower level ensures that the estimates conform a certain criterion. The key element of (4) is $\hat{\mathbf{f}}$, as it represents up-to-date traffic measurements and affects the estimation accuracy of $\hat{\mathbf{u}}$ and $\hat{\mathbf{f}}$ to a large extent. According to [38], regardless of the quality of $\bar{\mathbf{u}}$, better estimations are achieved as the number of monitored road-segments (i.e. $|\bar{\mathbf{f}}|$) approaches the number of O-D pairs (i.e. $|\bar{\mathbf{u}}|$).

This is difficult to satisfy using stationary traffic sensors such as loop detectors and video cameras, since they are installed mostly on major roads and highways – which constitute only a small portion of a city. As a result, their measurements are insufficient for estimating citywide traffic states [36, 39].

4 Traffic condition estimation

Our goal is to estimate traffic conditions of a city-scale network. To compensate the inadequacy of stationary sensors, we use data from the ubiquitous mobile sensor (i.e. GPS device). Consequently, the number of road segments (i.e. $|\bar{\mathbf{f}}|$) that are being ‘measured’ is greatly increased. In order to further improve the accuracy of $\hat{\mathbf{f}}$, we need to address two challenges posed by GPS data, namely low-sampling rate and spatial-temporal sparsity. The first causes the inference of travel times on individual road segments difficult and the second causes missing data in certain areas and time intervals.

We propose three procedures to alleviate the above-mentioned issues: *coarse inference*, *iterative refinement*, and *nested optimisation*. The first two procedures are used to generate accurate inference of traffic conditions in areas with GPS data coverage (i.e. handle low-sampling rate). The third procedure is used to provide estimations of traffic conditions in areas without GPS data coverage (i.e. handle spatial sparsity). The temporal sparsity is addressed using the interpolation technique developed in [12].

Traffic is commonly assumed to be quasi-static [11] and has a weekly period [27]. Based on these observations, we divide 1 week into hourly time intervals and treat the traffic within each interval as static. The three procedures are applied separately in each time interval. Thus, in Sections 4.1–4.3, we explain these procedures without referring to the time interval. In Section 4.4, we discuss the computational complexity and overhead of our approach.

4.1 Coarse inference

The travel time of a road segment is intrinsically stochastic as a result of many factors: fluctuations in traffic demand, varying weather conditions, and heterogeneous driver behaviours. It is natural to model a travel time as a random variable that subjects to a probability distribution $Z_{e \in \mathcal{E}}$. Our objective of Sections 4.1 and 4.2 is then to derive an accurate estimation of Z , the joint distribution of travel times of all road segments in a network that have GPS data coverage. Since the procedure described in Section 4.2 is an iterative scheme, having a good initial point becomes essential and this is the goal of *coarse inference*.

To be concrete, we obtain the coarse inference of $\mathbb{E}(Z)$ by solving a convex optimisation program. Inspired by Wardrop equilibria [13], we treat the time difference Δt between two GPS points s_i and s_{i+1} as the minimum travel time of all paths that connect s_i and s_{i+1} . Thus, $\forall k$ if $t_k = t_k(k) \leq \Delta t$, we raise $\{t_e\}_{e \in k}$ until collectively $t_k \geq \Delta t$. Denoting $\mathbb{E}(Z)$ as $\mathbf{t} = [t_1, t_2, \dots, t_{|\mathcal{E}|}]^T$ and corresponding indicator variables as $\mathbf{x} = [\xi_1, \xi_2, \dots, \xi_{|\mathcal{E}|}]^T$, where $\xi_e = 1$, $\forall e \in k$ and $\xi_e = 0$, $\forall e \notin k$, this constraint is represented as $\mathbf{x}^T \mathbf{t} \geq \Delta t$. The collection of these constraints, \mathcal{J} , then takes the

form $\mathcal{B}^T \mathbf{t} \geq \mathcal{H}$, where $\mathcal{B}_{|\mathcal{J}| \times |\mathcal{E}|}$ consists of $\{\xi_j\}_{j=1,\dots,|\mathcal{J}|}$ and $\mathcal{H}_{|\mathcal{J}| \times 1}$ consists of $\{\Delta t_j\}_{j=1,\dots,|\mathcal{J}|}$.

In order to establish a feasible region, we set the upper bound of a travel time t_e as the free-flow travel time, $t_{e,\min}$ (taking 120% of the speed limit), and the lower bound as the jam-density travel time, $t_{e,\max}$ (taking speed 0.5 m/s). To model the correlation of traffic patterns among nearby roads, we further propose a regularisation term, \mathcal{D} , as a graph-guided-fused-lasso penalty [40, 41] by enumerating all pairs of road segments at each $v \in \mathcal{V}$. The final optimisation program is as follows:

$$\begin{aligned} & \underset{\mathbf{t}, \mathbf{z}}{\text{minimise}} \quad \|\mathbf{z}\|_1, \\ & \text{subject to} \quad \mathbf{z} = \mathcal{D}\mathbf{t}, \\ & \quad \mathcal{B}^T \mathbf{t} \geq \mathcal{H}, \\ & \quad \mathbf{t}_{\min} \leq \mathbf{t} \leq \mathbf{t}_{\max}. \end{aligned} \quad (5)$$

In the above formulation, the objective function is convex as well as continuous. The constraints consist of linear functions and are bounded in a convex feasible region. Therefore, (5) represents a convex optimisation program and the minimal value exists in the feasible region.

4.2 Iterative refinement

Using the previous results as input, we refine $\mathbb{E}(Z)$ using a nested iterative process. The outer loop alternates between map-matching and travel-time inference, while the inner loop (within in travel-time inference) performs an expectation–maximisation (EM) algorithm. The rationale of this design is that as we obtain better estimations of travel times, we also obtain better map-matching results, and vice versa.

By leveraging gradually updated travel times of a network, we adopt the shortest-travel time criterion for map-matching. First, for a GPS point s , a set of candidate mapping positions \mathcal{Q} (e.g. $|\mathcal{Q}| = 5$) including end points and positions on road segments is formed based on their Euclidean distances to s . Next, for any pair of candidate positions of consecutive GPS points $q_i \in \mathcal{Q}_i$ and $q_{i+1} \in \mathcal{Q}_{i+1}$, the shortest travel-time path connecting q_i and q_{i+1} is recorded. Among all such paths, the one with the least difference to $\Delta t = \Delta t(s_i, s_{i+1})$ is treated as the traversed path and denoted as k .

After map-matching all GPS points, we obtain a collection of aggregate measurements, $\mathcal{N} = \{(\bar{k}_r, \Delta t_r)\}_{r=1,\dots,|\mathcal{N}|}$, which consist of map-matched paths and their corresponding travel times. In order to estimate Z parameterised by θ given \mathcal{N} , we set up the learning problem of $\theta = \{\theta_e\}_{e \in \mathcal{E}}$ to be *maximum likelihood estimation*:

$$\max_q \mathcal{L}(\theta | \mathcal{N}) = \sum_r \log \pi(\Delta t_r | \bar{k}_r; \theta), \quad (6)$$

where π represents the distribution for modelling a travel time and is assumed univariate and pairwise independent to travel times of other road segments [42]. With these assumptions, we can factorise (6) as follows (the precedent term $\sum_r \log$ is omitted for simplicity):

$$\begin{aligned} \pi(\Delta t_r | \bar{k}_r; \theta) &= \int \pi(\Delta t_r | Z, \bar{k}_r) \pi(Z; \theta) dZ \\ &= \int \pi(\Delta t_r | Z, \bar{k}_r) \left(\prod_{e: \xi_e = 1} \pi(Z_e; \theta_e) dZ_e \right). \end{aligned} \quad (7)$$

Equation (7) can be solved via the EM algorithm. In the E-step, for a pair $(\bar{k}_r, \Delta t_r)$, Δt_r is decomposed and distributed to road segments to suffice $\mathbf{x}^T \mathbf{t}^* = \Delta t_r$, where \mathbf{t}^* is a possible decomposition of Δt_r (i.e. $\sum \mathbf{t}^* = \Delta t_r$). In addition, the likelihood values, w^* , are computed based on the estimated θ so far. We call a pair (\mathbf{t}^*, w^*) a *random allocation*. The realisation of \mathbf{t}^* is essentially a fulfilment of conditional expectation calculation and can be generated

infinitely. In the M-step, all random allocations are gathered and regrouped by individual road segments. For a road segment e with regrouped samples $\{(t_{e,i}, w_{e,i})\}_{i=1,\dots,I}$, we learn θ_e using the following formula:

$$\max_{\theta_e} \sum_{i=1}^I w_{e,i} \log \pi(t_{e,i}; \theta_e). \quad (8)$$

To solve (8), following [11], we select π to be gamma distribution, as its positive domain and robustness against long-tail traffic data are more suitable to model the travel time of a road segment. We refer interested readers to [11] for more details. Subsequently, the conditional sampling to suffice $\mathbf{x}^T \mathbf{t}^* = \Delta t_r$ using the shape factor α and the scale factor β is realised as follows:

$$\begin{aligned} A_e &\sim \Gamma\left(\alpha_e, \frac{\xi_{r,e} \beta_e}{\Delta t_r}\right), \\ t_e &= \frac{\Delta t_r}{\xi_{r,e}} \frac{A_e}{\sum_e A_e}, \quad e: \xi_{r,e} = 1, \quad e \in \mathcal{E}, \end{aligned} \quad (9)$$

where $\{t_e\}_{e=1,\dots,|\mathcal{E}|} = \mathbf{t}^*$ and $\{\xi_{r,e}\}_{e=1,\dots,|\mathcal{E}|} = \mathbf{x}_r$. For initialising $\theta_e = (\alpha_e, \beta_e)$, we take the current $\mathbb{E}(Z)$ as the expectation and 60 s as the standard deviation.

After solving (8) for all road segments with GPS data coverage, we have obtained an estimation of Z and our framework performs map-matching again. The iteration continues until certain stop criteria are reached (e.g. ten iteration runs) and the final $\mathbb{E}(Z)$ is treated as \bar{f} to serve as the input to *nested optimisation*. In our computation, the conversion between travel times and flows is attained by inverting the road-segment *performance function* proposed by the US Bureau of Public Roads:

$$t_e = t_{e,\min} \left(1 + 1.5 \left(\frac{f_e^4}{c_e} \right) \right), \quad \forall e \in \mathcal{E}, \quad (10)$$

where c_e is the capacity of a road segment e computed as $c_e = 1700 + 10t_{e,\min}$ if $t_{e,\min} \leq 70$ mph and $c_e = 2400$ otherwise.

4.3 Nested optimisation

The goal of this procedure is to address spatial sparsity within GPS data by deriving traffic states on road segments without data coverage.

We achieve this by solving (4) using \bar{u} (i.e. the target O–D pairs), \bar{f} (i.e. the target road-segment flows), and \mathcal{M} (i.e. the assignment map). While \bar{u} is usually obtained from existing data and \mathcal{M} is chosen to be user equilibrium [i.e. (3)], \bar{f} is obtained using the procedures explained in Sections 4.1 and 4.2. Following [36], we select the distance function in (4) to be the generalised least squares (GLS) estimator. By further assuming \bar{u} and \bar{f} are results of the following stochastic system of equations:

$$\begin{aligned} \bar{u} &= \hat{u} + \epsilon_1, \\ \bar{f} &= \hat{f} + \epsilon_2. \end{aligned} \quad (11)$$

Equation (4) now has an explicit form: (see (12)) where U and V are variance–covariance matrices of ϵ_1 and ϵ_2 , respectively. We take $\mathbb{E}(\epsilon_1) = 0$ and $\mathbb{E}(\epsilon_2) = 0$. The additional factor η is introduced to leverage the importance of \bar{u} and \bar{f} in the estimation. We solve (12) iteratively using quadratic programming and the Frank–Wolfe solver [43].

4.4 Computational complexity

Here, we analyse the time complexity of our approach. A road network is denoted as a directed graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$ where \mathcal{E} represents edges and \mathcal{V} the nodes. For each GPS point, during the map-matching, we have a set of candidate mapping positions denoted as \mathcal{Q} . So, given a GPS trajectory which has $|\mathcal{S}|$ points, the

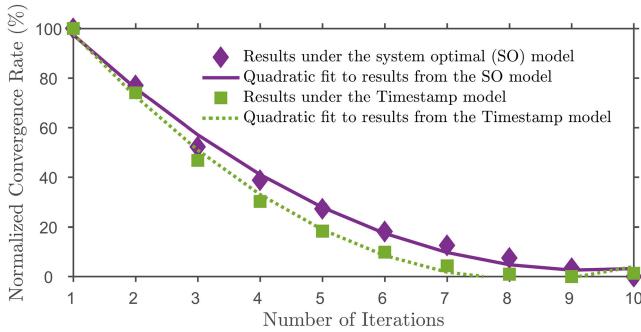


Fig. 3 Relationship between the normalised convergence rate (%) and the number of iterations of the outer loop of our iterative process is shown. The convergence rate decreases quadratically as the number of iterations increases and tends to flat after ten iterations

map-matching complexity is $O(|\mathcal{S}||\mathcal{Q}|^2|\mathcal{E}|\log|\mathcal{E}|)$. If we choose a small value for $|\mathcal{Q}|$ (e.g. 5), the actual complexity can be reduced to $O(|\mathcal{S}||\mathcal{E}|\log|\mathcal{E}|)$.

The complexity of conditional sampling is $O(mn|\mathcal{S}|)$, where n represents the number of road segments that a path covers and m the number of random allocations we would like to generate. Since m is a constant and can be a small number, the complexity is $O(n|\mathcal{S}|)$ in practice. The complexity of learning given gamma distribution is the same as of the conditional sampling and can be reduced to $O(n|\mathcal{S}|)$ as well.

In solving the traffic assignment problem, we use a line search within the Frank–Wolfe solver. This is also the key step to evaluate the objective function. The complexity of a line search algorithm is $O(m\log n)$, where m represents the number of intervals and n the number of iterations. Again, by choosing a small m , the algorithmic complexity can be decreased to $O(\log n)$. We refer interested readers to [43] for a thorough discussion of the rest aspects regarding the Frank–Wolfe solver. We have used the CVX package (obtained from <http://cvxr.com/cvx/>) for solving the convex optimisation described in (5) and the GLS estimator in (12). The complexity information can be found in the package's website. Lastly, in order to facilitate the computation, we need to store a road network. By using the adjacency list presentation, the space complexity is $O(|\mathcal{E}| + |\mathcal{V}|)$.

5 Experiments

For evaluating our approach, we compare our method to Hunter [11] and Rahmani *et al.* [15] on a real road network with extensive heuristic traffic conditions and synthetic GPS traces. In Section 5.1, we discuss the generation of our data sets. In Section 5.2, we provide the comparison and evaluation to the two methods mentioned previously. Lastly, in Section 5.3, we analyse the parameters of our nested optimisation program.

5.1 Data sets

We use the road network from downtown San Francisco (obtained from openstreetmap.org, Fig. 2) as the benchmark. The network contains 5407 nodes, 1612 road segments, and 296 TAZs (obtained from data.sfgov.org). We have also generated two sets of heuristic network travel times using the Cabspotting data set [44] via the technique developed in [39] (Fig. 2 Left) and abundant synthetic GPS traces based on the generated heuristic network travel times.

The first set of heuristic network travel times is generated via the system optimal (SO) model [45], which solves the traffic assignment problem by minimising the entire travel time of a road network. The result is a set of flows and travel times of all road segments in a network. Since the Cabspotting data set only represents partial vehicle population, we multiply the result by ten constants resulting in ten congestion levels ranging uniformly from 0.19 to 1.85. [The congestion level is measured by volume over capacity (VOC) and computed as $\sum_{e \in \mathcal{E}} (f_e/c_e)$.]

The second set of heuristic network travel times is generated through GPS timestamps. Using the Cabspotting data set, we equally distribute the time difference of a pair of GPS points to all paths that connect them. For a road segment that is covered by multiple GPS traces, the average travel time is computed. Using this approach, we have produced 24 network travel times representing 24 h of a typical weekday. An example can be seen at Fig. 2 Right. We refer to this method as the *timestamp* model.

Using the established network travel times, we generate 20 collections of GPS traces. Each collection contains 30 sets GPS traces and all sets in a collection share the same number of traces, which goes from 50 to 1000 in increments of 50. As a result, we obtain over 10 million synthetic GPS traces for our experiments. A sampled GPS trace is created by selecting a random source and a target in the network and planning the route using the shortest travel time criterion. To mimic features of the Cabspotting data set, the sampling rate is set to be 60 s and all coordinates are perturbed by the Gaussian noise (0, 20m) [22].

5.2 Evaluation and comparison

We evaluate our technique by comparing to Hunter [11] and Rahmani *et al.* [15]. The first method is equivalent to the inner loop of our *travel-time inference*. The number of EM iterations is set to 5 and the number of random allocations per aggregate measurement is set to 100. These settings are responsible for the highest estimation accuracy in [11]. The second method takes a non-parametric perspective, using a kernel-based technique to estimate travel times. The weights used to allocate travel times on individual road segments are set to be the ratio of free-flow travel times among road segments [23].

The parameters of our nested iterative process are set as follows: retaining the same settings for the inner loop as in [11], we empirically set the number of iterations for the outer loop to 10. This setting is based on the analysis that shows the relationship between the normalised convergence rate and the number of iterations for both types of network travel times. As a result, the convergence rate decreases quadratically as the number of iterations increases and tends to flatten after ten iterations. This can be seen in Fig. 3. Each datum in the plot is the average value computed using all network travel times across all sets of synthetic GPS traces of either the SO model (6000 trails) or the timestamp model (14,400 trails). The measurement of each trial is the mean squared error (MSE) between an estimated and a heuristic traffic condition, i.e. $(\sum_{e \in \mathcal{E}} (t_e - \hat{t}_e)^2)/|\mathcal{E}|$, where t_e represents a heuristic travel time and \hat{t}_e an estimated travel time.

We evaluate our technique using three metrics. The first metric is the performance gain of our technique over the existing methods on travel times by considering all road segments of a network. We compute this metric based on $M = (\sum_{e \in \mathcal{E}} (t_e - \hat{t}_e)^2)/|\mathcal{E}|$ as $(M_{\text{other}} - M_{\text{our}})/M_{\text{our}}$, where M_{our} represents the error between a recovered traffic condition and its corresponding heuristic traffic condition computed using our technique, and M_{other} is the same as

$$\begin{aligned}
 & \underset{\hat{u}}{\text{minimise}} && \eta(\hat{u} - \hat{u})^T U^{-1}(\hat{u} - \hat{u}) + (1 - \eta)(\hat{f} - \hat{f})^T V^{-1}(\hat{f} - \hat{f}), \\
 & \text{subject to} && \hat{f} = \mathcal{M}(\hat{u}), \\
 & && \hat{u} \geq 0, \\
 & && \hat{f} \geq 0, \\
 & && 0 \leq \eta \leq 1,
 \end{aligned} \tag{12}$$

M_{our} but computed using an existing method. The maximum relative improvements over Hunter [11] and Rahmani *et al.* [15] under the SO model are 78 and 97% (Fig. 4 Top-left), and under the timestamp model are 54 and 49% (Fig. 4 Top-right), respectively. In general, when more GPS traces used in estimation, our technique achieves better performance gains. Such effects are more apparent with the SO model than the timestamp model.

The second metric is the error rate of the aggregate travel time of a network, computed as

$$\frac{|\sum_e \hat{t}_e - \sum_e t_e|}{\sum_e t_e} \quad \forall e \in \mathcal{E}.$$

Using the SO model, the minimum error rate of our technique is 18%, of Rahmani *et al.* [15] is 34%, and of Hunter [11] is 48% (Fig. 4 Middle-left). Using the timestamp model, the corresponding minimum error rates are 8, 28, and 37% (Fig. 4 Middle-right). As the number of GPS traces used in estimation increases, our technique demonstrates consistent advantages in performance over the other two methods.

The third metric is the map-matching accuracy computed as

$$\text{SR} = \frac{\#\text{successfully identified road segments}}{\#\text{actual road segments in the trace}}.$$

We compute the relative improvement as $(\sum \text{SR}_{\text{our}} - \sum \text{SR}_{\text{other}}) / \sum \text{SR}_{\text{our}}$ by summing the success rates of all road segments. The maximum gain of our method over Hunter [11] and Rahmani *et al.* [15] under the SO model are 28 and 34%, and under the timestamp model are 19 and 25%, correspondingly (Fig. 4 Bottom). Again, as the number of GPS traces used in computing increases, gains in the improvements are observed.

5.3 Analysis of nested optimisation

We have shown that our approach outperforms the existing methods on estimating travel times in areas with GPS data coverage. In turn, by inverting (10), we obtain better estimations of traffic flows. Using them as inputs to the nested optimisation program [i.e. (12)], we can derive traffic states in areas without data coverage. In the following, we analyse the factors that influence the estimation accuracy of the nested optimisation.

The accuracy of (12) is affected by η (i.e. the weighting parameter), the noise level of \bar{u} (i.e. target O-D pairs), and the noise level of \bar{f} (i.e. target flows). In reality, the noise level of \bar{u} is difficult to assess because the true values of \bar{u} are seldom [37]. For this reason, in the analysis of η and noise levels of \bar{f} , we set the normalised noise level of \bar{u} to 50%. In addition, we assume the noises of \bar{u} and \bar{f} have zero mean and diagonal variance-covariance matrices [37].

As we mentioned in Section 3, among the parameters, \bar{f} influences estimation accuracy the most. Here, as illustrated in Fig. 5, we show the normalised noise levels of \bar{f} computed based on MSE of the estimated travel times to their corresponding heuristic travel times, using either the SO model (Left) or the timestamp model (Middle). Overall, our technique produces lower noise levels of \bar{f} than the other two methods, especially with the SO model, which better approximates real-world traffic than the timestamp model [45].

We further analyse the impact of η and the noise level of \bar{f} on the estimation accuracy of \hat{u} (i.e. the estimated O-D pairs). Specifically, we compute the normalised MSE of \hat{u} using different values of η and various noise levels of \bar{f} . The results are shown in Fig. 5 Right. When η takes a small value (e.g. 0.1), the impact of \bar{u} is restricted, thus, the MSE of \hat{u} becomes sensitive to perturbations of \bar{f} . As we gradually increase the value of η , the impact of \bar{f} attenuates. Nevertheless, the MSE of \hat{u} increases as the noise level of \bar{f} progresses. This result highlights the need of a lower noise level of \bar{f} , which can be fulfilled with our technique as shown in Fig. 5 Left and Middle.

6 Field tests

In order to study urban traffic dynamics, we conduct field tests in two diverse cities in two continents, namely Beijing and San Francisco.

The GPS data sets used in the field tests are from the Cabspotting project [44] and the T-drive project [46]. The TAZs of San Francisco are obtained from data.sfgov.org and the area considered is from $\langle 37.7083, -122.514 \rangle$ to $\langle 37.812, -122.358 \rangle$ (in latitude and longitude). The TAZs of Beijing are constructed by dividing the area from $\langle 39.8043, 116.1904 \rangle$ to $\langle 40.035, 116.5673 \rangle$ into 1 km by 1 km grids. More information regarding our data sets can be found in Table 1.

We first demonstrate the estimated traffic states of the two cities in Fig. 6. All computations are conducted in epoch time and the target O-D pairs are estimated using the technique from [39]. We observe that the recovered traffic patterns show clearly periodic phenomena over the course of a week, which is considered as one of the hallmarks of traffic [27], for the entire network and decomposed road types. In San Francisco, saddle shapes appear showing mid-day traffic relief over several days of a week. Such phenomena are more evident on major roads (i.e. motorway and truck), but less on the other types of roads. In Beijing, we do not observe such saddle structures. This suggests the similar usage of various types of road as transportation infrastructure and the congestion staying severe throughout daytime.

We further visualise estimation results over four time intervals of Beijing in Fig. 7: *weekend morning traffic* represented by Sunday 9 AM, *weekday morning traffic* represented by Tuesday 9 AM, *weekday mid-day traffic* represented by Thursday Noon, and *weekday evening traffic* represented by Friday 7 PM. We have observed two traffic features in this visualisation. First, the Sunday morning's congestion tends to be the least severe and the Friday night's congestion tends to be the most severe. Second, the congestion situation of Thursday Noon is slightly better than Tuesday 9 AM, especially considering the traffic between the fourth and the fifth ring roads, where more residential units are found than the region inside the fourth ring road.

7 Conclusion and future work

We have presented a novel framework for estimating urban traffic states using iterative refinement and Wardrop equilibria on mobile-sensor data.

First, we obtain the coarse inference of network travel times using convex optimisation. Next, we take an iterative approach to jointly estimate traversed paths of probe vehicles and travel times of road segments in a network. Then, we incorporate the previous estimated results into a nested optimisation program to estimate traffic states in areas without data coverage. By further interpolating the temporal missing values, we obtain full citywide traffic dynamics.

We have evaluated our approach using a real road network resulting in consistent and notable improvements over the state-of-the-art methods. In order to study urban traffic patterns, two large-scale field tests were conducted in Beijing and San Francisco. The estimated results can further enable traffic simulations and animations in various formats [47, 48].

There are several possible future directions. First of all, the coordination of probe vehicles can be explicitly taken into account to improve estimation efficiency. Second, with estimated traffic states, a *real-time* probabilistic mapping technique for GPS traces can be developed. Lastly, by fusing estimated results from historical data with fine-grained traffic simulations, it is possible to derive even more accurate forecasting of citywide traffic.

7 Acknowledgments

The authors would like to thank the National Science Foundation and US Army Research Office.

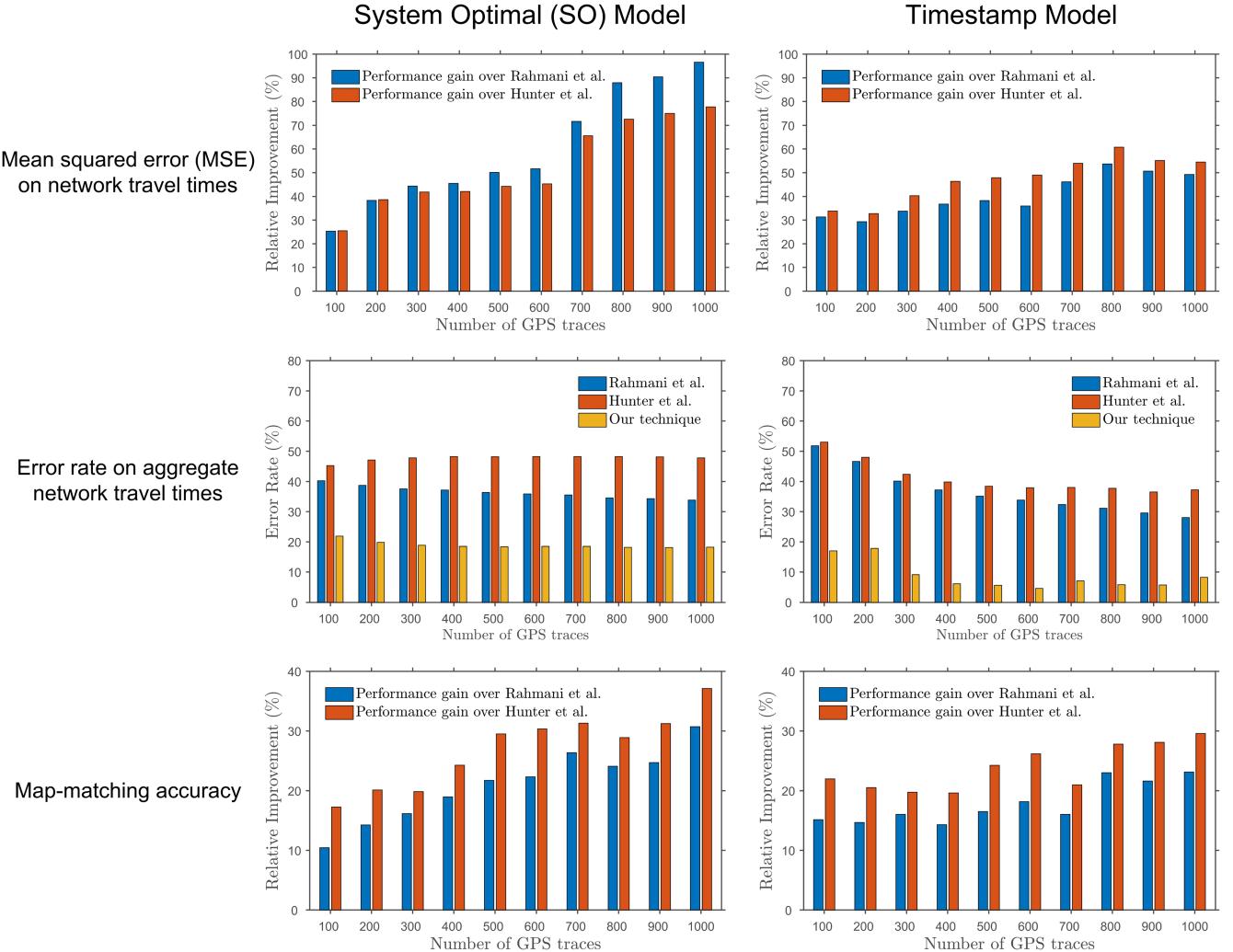


Fig. 4 Left diagrams show results using the SO model while Right diagrams show results using the timestamp model. TOP, the performance gain of network travel times measured in MSE. Middle: The error rates of all three methods on aggregate network travel times. Bottom: The performance gain of map-matching accuracy measured using all synthetic GPS traces. Our technique achieves consistent improvements over the other two methods in all measurements

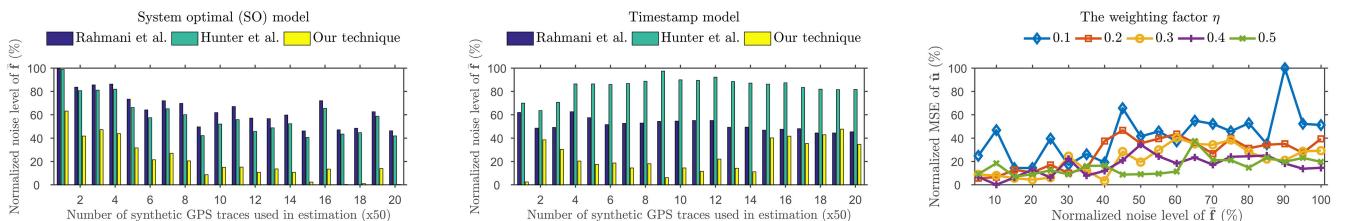


Fig. 5 Left and middle: The normalised noise levels of \bar{f} (i.e. target flows) computed using our technique and the existing methods. In general, for both models, our technique produces lower noise levels than the other two methods. Right: The normalised MSE of \hat{u} (i.e. target O-D pairs) using different values of η and various noise levels of \bar{f} . When η is small, the error is more sensitive to perturbations of \bar{f} . Overall, the error increases as the noise level of \bar{f} increases. This highlights the need for a lower noise level of \bar{f} and can be fulfilled using our technique as shown in Left and Middle

Table 1 Statistics of road networks and GPS data sets used in our field tests. The road networks are obtained from <http://openstreetmap.org>. The TAZ file of San Francisco is obtained from <http://data.sfgov.org> and the area considered is from $\langle 37.7083, -122.514 \rangle$ to $\langle 37.812, -122.358 \rangle$ in latitude and longitude. The TAZs of Beijing are constructed by dividing the area from $\langle 39.8043, 116.1904 \rangle$ to $\langle 40.035, 116.5673 \rangle$ into $1 \text{ km} \times 1 \text{ km}$ grids

| City | Road network | | | | Source | Period | Type | #Vehicles | #Points | Distance travelled |
|---------------|--------------|---------|-------|------------------|---------|-----------|--------|-------------|-------------------|--------------------|
| | #Nodes | #Edges | #TAZs | GPS data set | | | | | | |
| San Francisco | 37,635 | 43,778 | 786 | Cabspotting [44] | 3 weeks | taxi cabs | 536 | ~11 million | >4.7 million (km) | |
| Beijing | 91,066 | 105,121 | 839 | T-drive [46] | 1 week | taxi cabs | 10,357 | ~15 million | >9 million (km) | |

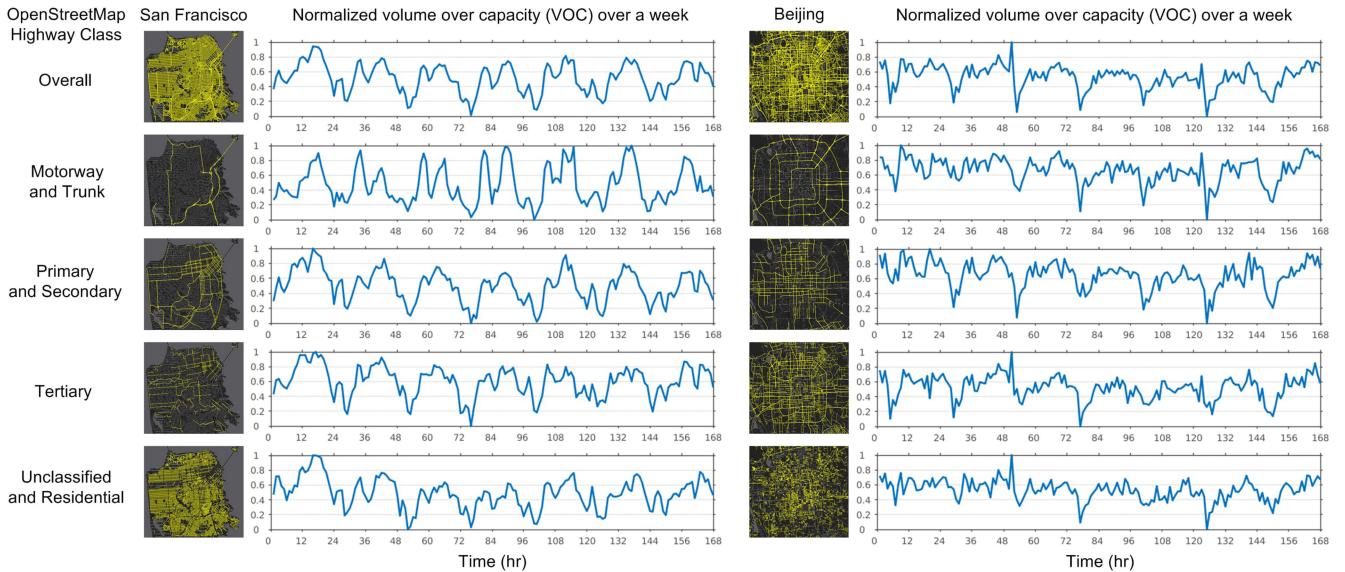


Fig. 6 Estimated traffic states measured in average VOC of San Francisco and Beijing for various types of roads. All computations are conducted in epoch time. Our technique successfully recovers the periodic phenomena in all cases. In San Francisco, saddle shapes appear on motorway and truck and overall roads for several days indicating mid-day traffic relief. Such phenomena are not observed in Beijing, which suggests the similar usage of different types of roads and the congestion forming throughout daytime

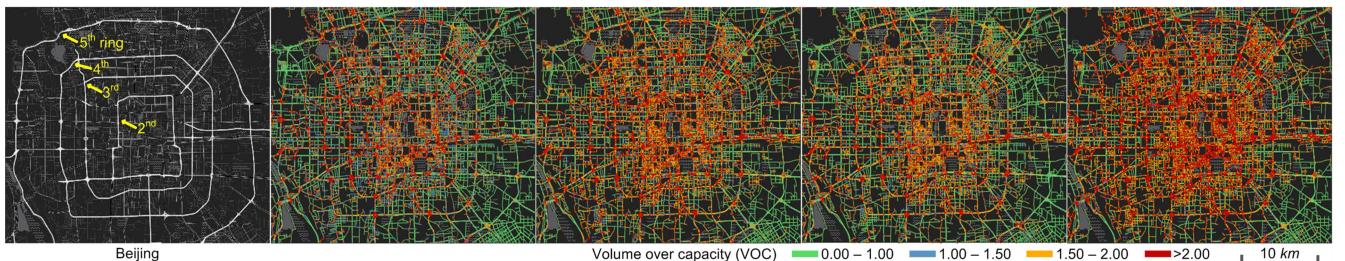


Fig. 7 Field test in Beijing. Different colours represent different ranges of VOC. Four time periods, namely Sunday 9 AM, Tuesday 9 AM, Thursday Noon, and Friday 7 PM, are displayed as examples to illustrate weekend versus weekday and morning versus evening traffic

8 References

- [1] Schrank, D., Eisele, B., Lomax, T., et al.: ‘2015 urban mobility scorecard’ (Texas A&M Transportation Institute and INRIX, 2015)
- [2] Zheng, Y., Capra, L., Wolfson, O., et al.: ‘Urban computing: concepts, methodologies, and applications’, *ACM Trans. Intell. Syst. Technol.*, 2014, **5**, (3), pp. 38:1–38:55
- [3] Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.-L.: ‘Understanding individual human mobility patterns’, *Nature*, 2008, **453**, (7196), p. 779
- [4] Chen, C., Zhang, D., Guo, B., et al.: ‘Triplanner: personalized trip planning leveraging heterogeneous crowdsourced digital footprints’, *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (3), pp. 1259–1273
- [5] Richter, C., Ware, J., Roy, N.: ‘High-speed autonomous navigation of unknown environments using learned probabilities of collision’. IEEE Int. Conf. Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 6114–6121
- [6] Castro, P.S., Zhang, D., Chen, C., et al.: ‘From taxi gps traces to social and community dynamics: a survey’, *ACM Comput. Surv.*, 2013, **46**, (2), pp. 17:1–17:34
- [7] Yuan, J., Zheng, Y., Zhang, C., et al.: ‘An interactive-voting based map matching algorithm’. Proc. 11th Int. Conf. Mobile Data Management, Kansas City, MO, USA, 2010, pp. 43–52
- [8] Hunter, T., Abbeel, P., Bayen, A.: ‘The path inference filter: model-based low-latency map matching of probe vehicle data’, *IEEE Trans. Intell. Transp. Syst.*, 2014, **15**, (2), pp. 507–529
- [9] Quddus, M., Washington, S.: ‘Shortest path and vehicle trajectory aided map-matching for low frequency GPS data’, *Transp. Res. C, Emerg. Technol.*, 2015, **55**, pp. 328–339
- [10] Tang, J., Song, Y., Miller, H., et al.: ‘Estimating the most likely space–time paths, dwell times and path uncertainties from vehicle trajectory data: a time geographic method’, *Transp. Res. C, Emerg. Technol.*, 2016, **66**, pp. 176–194
- [11] Hunter, T.: ‘Large-scale, low-latency state estimation of cyberphysical systems with an application to traffic estimation’, PhD thesis, University of California, Berkeley, 2014
- [12] Li, W., Nie, D., Wilkie, D., et al.: ‘Citywide estimation of traffic dynamics via sparse GPS traces’, *IEEE Intell. Transp. Syst. Mag.*, 2017, **9**, (3), pp. 100–113
- [13] Wardrop, J.: ‘Some theoretical aspects of road traffic research’, *Proc. Inst. Civ. Eng.*, 1952, **1**, (3), pp. 325–362
- [14] Yang, H., Sasaki, T., Iida, Y., et al.: ‘Estimation of origin-destination matrices from link traffic counts on congested networks’, *Transp. Res. B, Methodol.*, 1992, **26**, (6), pp. 417–434
- [15] Rahmani, M., Jenelius, E., Koutsopoulos, H.: ‘Non-parametric estimation of route travel time distributions from low-frequency floating car data’, *Transp. Res. C, Emerg. Technol.*, 2015, **58**, pp. 343–362
- [16] Li, W., Jiang, M., Chen, Y., et al.: ‘Estimating traffic conditions at metropolitan scale using traffic flow theory’. 97th Transportation Research Board Annual Meeting, Washington, DC, USA, 2018
- [17] Work, D., Blandin, S., Tossavainen, O.-P., et al.: ‘A traffic model for velocity data assimilation’, *Appl. Math. Res. Express*, 2010, **2010**, (1), pp. 1–35
- [18] Li, L., Chen, X., Zhang, L.: ‘Multimodel ensemble for freeway traffic state estimations’, *IEEE Trans. Intell. Transp. Syst.*, 2014, **15**, (3), pp. 1323–1336
- [19] Kong, Q.-J., Zhao, Q., Wei, C., et al.: ‘Efficient traffic state estimation for large-scale urban road networks’, *IEEE Trans. Intell. Transp. Syst.*, 2013, **14**, (1), pp. 398–407
- [20] Zhang, J.-D., Xu, J., Liao, S.S.: ‘Aggregating and sampling methods for processing GPS data streams for traffic state estimation’, *IEEE Trans. Intell. Transp. Syst.*, 2013, **14**, (4), pp. 1629–1641
- [21] Wang, Y., Zheng, Y., Xue, Y.: ‘Travel time estimation of a path using sparse trajectories’. Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, New York, NY, USA, 2014, pp. 25–34
- [22] Lou, Y., Zhang, C., Zheng, Y., et al.: ‘Map-matching for low-sampling-rate gps trajectories’. Proc. 17th ACM SIGSPATIAL Int. Conf. Advances in Geographic Information Systems, Seattle, WA, USA, 2009, pp. 352–361
- [23] Hellings, B., Izadpanah, P., Takada, H., et al.: ‘Decomposing travel times measured by probe-based traffic monitoring systems to individual road segments’, *Transp. Res. C, Emerg. Technol.*, 2008, **16**, (6), pp. 768–782
- [24] Herring, R., Hofleitner, A., Abbeel, P., et al.: ‘Estimating arterial traffic conditions using sparse probe data’. 13th Int. IEEE Conf. Intelligent Transportation Systems (ITSC), 2010, pp. 929–936
- [25] Asif, M.T., Mitrovic, N., Dauwels, J., et al.: ‘Matrix and tensor based methods for missing data estimation in large traffic networks’, *IEEE Trans. Intell. Transp. Syst.*, 2016, **17**, (7), pp. 1816–1825
- [26] Mitrovic, N., Asif, M.T., Dauwels, J., et al.: ‘Low-dimensional models for compressed sensing and prediction of large-scale traffic data’, *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (5), pp. 2949–2954
- [27] Hofleitner, A., Herring, R., Bayen, A., et al.: ‘Large scale estimation of arterial traffic and structural analysis of traffic patterns using probe vehicles’. 91st Transportation Research Board Annual Meeting, Washington, DC, USA, 2012
- [28] Gupta, A.K., Redhu, P.: ‘Analyses of the driver’s anticipation effect in a new lattice hydrodynamic traffic flow model with passing’, *Nonlinear Dyn.*, 2014, **76**, (2), pp. 1001–1011

- [29] Tang, T.-Q., Huang, H.-J., Shang, H.-Y.: ‘Influences of the driver’s bounded rationality on micro driving behavior, fuel consumption and emissions’, *Transp. Res. D, Transp. Environ.*, 2015, **41**, pp. 423–432
- [30] Tang, T.-Q., Yi, Z.-Y., Zhang, J., *et al.*: ‘Modelling the driving behaviour at a signalised intersection with the information of remaining green time’, *IET Intell. Transp. Syst.*, 2017, **11**, (9), pp. 596–603
- [31] Zhao, J., Li, P., Zheng, Z., *et al.*: ‘Analysis of saturation flow rate at tandem intersections using field data’, *IET Intell. Transp. Syst.*, 2018, **12**, (5), pp. 394–403
- [32] Jiang, D., Xu, Z., Chen, Z., *et al.*: ‘Joint time–frequency sparse estimation of large-scale network traffic’, *Comput. Netw.*, 2011, **55**, (15), pp. 3533–3547
- [33] Jiang, D., Xu, Z., Xu, H.: ‘A novel hybrid prediction algorithm to network traffic’, *Ann. Telecommun.*, 2015, **70**, (9–10), pp. 427–439
- [34] Nie, L., Jiang, D., Lv, Z.: ‘Modeling network traffic for traffic matrix estimation and anomaly detection based on Bayesian network in cloud computing networks’, *Ann. Telecommun.*, 2017, **72**, (5–6), pp. 297–305
- [35] Nie, L., Jiang, D., Guo, L., *et al.*: ‘Traffic matrix prediction and estimation based on deep learning in large-scale ip backbone networks’, *J. Netw. Comput. Appl.*, 2016, **76**, pp. 16–22
- [36] Bera, S., Rao, K.V.K.: ‘Estimation of origin–destination matrix from traffic counts: the state of the art’, *Eur. Transp.*, 2011, **49**, pp. 3–23
- [37] Cascetta, E., Nguyen, S.: ‘A unified framework for estimating or updating origin/destination matrices from traffic counts’, *Transp. Res. B, Methodol.*, 1988, **22**, (6), pp. 437–455
- [38] Marzano, V., Papola, A., Simonelli, F.: ‘Limits and perspectives of effective O–D matrix correction using traffic counts’, *Transp. Res. C, Emerg. Technol.*, 2009, **17**, (2), pp. 120–132
- [39] Yang, X., Lu, Y., Hao, W.: ‘Origin destination (OD) estimation using probe vehicle trajectory and link counts’, *J. Adv. Transp.*, 2016, **2016**
- [40] Tibshirani, R., Saunders, M., Rosset, S., *et al.*: ‘Sparsity and smoothness via the fused lasso’, *J. R. Stat. Soc. B*, 2005, **67**, (1), pp. 91–108
- [41] Chen, X., Lin, Q., Kim, S., *et al.*: ‘Smoothing proximal gradient method for general structured sparse regression’, *Ann. Appl. Stat.*, 2012, **6**, (2), pp. 719–752
- [42] Hofleitner, A., Herring, R., Bayen, A.: ‘Probability distributions of travel times on arterial networks: a traffic flow and horizontal queuing theory approach’. 91st Transportation Research Board Annual Meeting, Washington, DC, USA, 2012
- [43] LeBlanc, L., Morlok, E., Pierskalla, W.: ‘An efficient approach to solving the road network equilibrium traffic assignment problem’, *Transp. Res.*, 1975, **9**, (5), pp. 309–318
- [44] Piorkowski, M., Sarafijanovic-Djukic, N., Grossglauser, M.: ‘CRAWDAD dataset epfl/mobility (v. 2009-02-24)’, 2009. Available at <http://crawdad.org/epfl/mobility/20090224>
- [45] Sheffi, Y.: ‘Urban transportation network’ (Prentice Hall, Englewood Cliffs, NJ, USA, 1985)
- [46] Yuan, J., Zheng, Y., Zhang, C., *et al.*: ‘T-drive: driving directions based on taxi trajectories’. Proc. 18th SIGSPATIAL Int. Conf. Advances in Geographic Information Systems, San Jose, CA, USA, 2010, pp. 99–108
- [47] Wilkie, D., Sewall, J., Li, W., *et al.*: ‘Virtualized traffic at metropolitan scales’, *Front. Robot. AI*, 2015, **2**, p. 11
- [48] Li, W., Wolinski, D., Lin, M.C.: ‘City-scale traffic animation using statistical learning and metamodel-based optimization’, *ACM Trans. Graphics (SIGGRAPH Asia 2017)*, 2017, **36**, (6), pp. 200:1–200:12