Estimating Traffic Conditions At Metropolitan Scale Using Traffic Flow Theory

Weizi Li, Corresponding Author
Department of Computer Science
University of North Carolina at Chapel Hill
201 S Columbia St, Chapel Hill, NC 27599
Email: weizili@cs.unc.edu

Meilei Jiang
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Hanes Hall, 318 E Cameron Ave #3260, Chapel Hill, NC 27599
Email: jiangm@live.unc.edu

Yaoyu Chen
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill
Hanes Hall, 318 E Cameron Ave #3260, Chapel Hill, NC 27599
Email: yaoyuchen@unc.edu

Ming C. Lin
Department of Computer Science
University of North Carolina at Chapel Hill
201 S Columbia St, Chapel Hill, NC 27599
Email: lin@cs.unc.edu

Project website: gamma.cs.unc.edu/CityEst/

Word count: 5,978 words text + 6 figures x 250 words (each) = 7,478 words
Submission Date: August 1, 2017
ABSTRACT

Traffic has become a major problem in metropolitan areas around the world. It is of great importance to understand the complex interplay of road networks and traffic conditions. We propose a novel framework to estimate traffic conditions at metropolitan scale using GPS traces. Our approach begins with a coarse inference of network travel times by solving a convex optimization program based on traffic flow theory. Then, we iteratively refine the inferred network conditions and vehicle paths. Last, we perform a nested optimization to estimate traffic conditions on road segments that are not covered by GPS data. The evaluation and comparison of our approach over two state-of-the-art methods show up to 96.57% relative improvements. We have further conducted field tests by coupling road networks of Beijing and San Francisco with real-world GIS data, which involve 128,701 nodes, 148,899 road segments, and over 26 million GPS traces.

Keywords: Traffic conditions estimation, GPS data, Map-matching, Travel-time Inference, Nested optimization
INTRODUCTION

Traffic has become a major problem in metropolitan areas around the world. The extra cost due to traffic congestion and injuries are assessed over one trillion dollars worldwide. Therefore, to understand the complex interplay of road networks and travel conditions has been of a great interest in many contexts, including analyzing urban infrastructure (27), understanding human mobility (28), and designing better routing strategies (23). These applications highlight a need for developing a systematical framework that is capable of estimating traffic conditions with real-time sensor data.

So far, GPS dataset is one of the most valuable sources for urban computing due to its ubiquity (4). Nevertheless, in terms of estimating traffic conditions, several features of GPS datasets need to be addressed. One such feature is the low-sampling rate, i.e. there exists a large time gap (e.g., >60 seconds) between two consecutive GPS reports. Another feature is the spatial-temporal sparsity, i.e. for certain areas where the population density is low and certain time periods such as late-night hours, GPS traces are scarce. These features give rise to several challenges in estimating traffic conditions. To begin with, GPS points need to be mapped onto a road network and the traversed path connecting the points needs 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, since the path is likely to span several road segments, the aggregate travel time (i.e., difference of timestamps) needs to be decomposed and distributed to the individual road segments so that travel times of a network can be derived. Last, in order to achieve an accurate estimation, the missing travel times on certain road segments and over certain time periods due to the spatial-temporal sparsity require interpolation.

The three aforementioned challenges are referred to as map-matching, travel-time inference, and missing-value completion, respectively. These processes are commonly executed in tandem for deriving network travel times. Many map-matching approaches have adopted the shortest-distance criterion (16,25) to infer the traversed path. However, this assumption can lead to a considerable bias in a congested network where the shortest-distance path is no longer the shortest travel-time path (9,13). As a result, in the sequential view, such a bias will get cascaded into travel-time inference and affect the overall estimation accuracy. In addition, despite the abundant research on missing-value completion, the interpolation of spatial-temporal sparsity in data-lacking areas has not been well addressed.

We propose a novel framework for estimating traffic condition at city scale. To be specific, in each discretized time period, we first obtain a coarse inference of travel times on road segments with GPS data coverage by solving a convex optimization program with constraints inspired by the Wardrop Principles (20). Then, we refine the inferred vehicle paths and traffic conditions via iteratively performing map-matching and travel-time inference. Next, to handle the spatial sparsity, we treat the probe vehicle as mobile sensors and the generated GPS reports as monitoring results. Then, we perform a citywide estimation of traffic flows including road segments that are not covered by GPS data. This is achieved via the nested optimization: the upper level aims to derive the optimal trip distributions among different areas in a road network while the lower level satisfies the constraints imposed by the Wardrop Principles (18,20). Finally, we address the temporal sparsity by considering all time intervals and interpolating missing values via the Compressed-Sensing algorithm (13). The schematic diagram of our framework is shown in Figure 1. Our contributions are listed as follows:

- A convex optimization formulation that models similarity of traffic patterns among nearby road segments observed from real-world traffic 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 traffic conditions of a road network;
- A novel approach that incorporates estimated traffic flow into nested optimization to address spatial sparsity embedded in GPS data to obtain metropolitan-scale traffic estimations.

![Diagram](https://via.placeholder.com/150)

**FIGURE 1** Overview of our system: Convex Optimization coupled with traffic flow theory is used to derive initial results; refined results are obtained through joint Map-matching and Travel-time Inference; the Nested Optimization is applied to individual discretized time intervals for estimating citywide traffic condition; Missing-value Completion is performed over all time intervals to obtain final results.

We have evaluated our prototype implementation using a real road network that consists of 5,407 nodes and 1,612 road segments, 34 heuristic network travel times corresponding to various congestion levels and times of a day, and over 10 million sampled GPS traces. The effectiveness of our approach has been compared to state-of-the-art methods, namely Hunter et al. (9) and Rahmani et al. (17), resulting in up to 96.57% relative improvements. In order to understand urban traffic, we have conducted field tests in Beijing and San Francisco using real-world GIS datasets, which have 128,701 nodes, 148,899 road segments, and over 26 million GPS traces.

**RELATED WORK**

The estimation of traffic conditions has received increasingly attention over the past decades. In order to improve the estimation accuracy, researchers have adopted a variety of data types and simulation models (12,15). While these methods result in major improvements, their effectiveness is restricted on highway segments. In order to conduct a city-scale estimation, existing studies have resorted to GPS data. Nevertheless, because of the inherent noise and spatial-temporal sparsity, multiple steps are required and are commonly assembled into a sequential pipeline (10,26).

As the first procedure, map-matching handles noisy GPS points by mapping them back to a road network and inferring the most likely traversed trajectories of vehicles. Among many techniques, the most prominent one treats the shortest-distance path for connecting two consecutive GPS points as the “true” path (16,25). This assumption, however, leads to errors in a busy road network where the shortest-distance path and the shortest travel-time path differ (9,13). In such a case, many drivers are likely to prefer the shortest travel-time route rather than the
shortest-distance route, especially considering that modern GPS devices provide the former. Acknowledging a vehicle can drive on a slower route, by knowing travel times of a road network, we can compare road conditions with GPS timestamps to determine the actual traversed route of the vehicle. However, being the first step in the sequential pipeline, map-matching is usually performed on an empty road network where the network travel times are largely missing resulting in possibly many mapping errors.

After map-matching, a collection of map-matched vehicle paths is generated. Because of the low-sampling rate, in a complex urban environment, a path that connects two consecutive GPS points can consist of multiple road segments. In order to estimate travel times on all road segments, the difference between the two GPS timestamps needs to be distributed to individual road segments. This procedure is referred to as travel-time inference. To list few examples of this subject, Wang et al. (31) use a tensor-based decomposition approach to infer travel times of a road network. Hellinga et al. (5) have proposed an analytical solution based on observations of traffic flows. Rahmani et al. (17) take a non-parametric approach to infer network travel times. Herring et al. (6) and Hofleitner et al. (7) both adopt graphical-model based techniques. While many proposed algorithms are effective, they are usually complex in order to compensate the errors produced in the prior step (i.e., map-matching). In addition, as the timestamp differences can be assigned to a set of illy map-matched road segments, the inferred results are sometimes difficult to justify.

As the third step, missing-value completion has also been studied to various extents. For example, tensor-based approaches (1) were developed to explore shared characteristics of road segments in close proximity. In (13), researchers have interpolated missing values via Compressed Sensing-based algorithms. Despite of recent advances, there still exists a need for a systematical and replicable framework that incorporates Traffic Flow Theory (18) and conducts estimations of traffic conditions at the city scale and over an entire traffic period, as little progress has been made in this regard (8).

PRELIMINARIES
We represent a road network as a directed graph $G = (\mathcal{E}, \mathcal{V})$, in which $\mathcal{E}$ represents road segments and $\mathcal{V}$ represents end points of road segments. According to census data on trip information, a city can be separated into geographical units known as traffic analysis zones (TAZs). In planning transportation, the centers of TAZs are treated as locations where the traffic flow departs and arrives. The former are termed origins and the latter are termed destinations. We extract both origins $\mathcal{O}$ and destinations $\mathcal{D}$ from $\mathcal{V}$ (i.e., $\mathcal{O} \subseteq \mathcal{V}$ and $\mathcal{D} \subseteq \mathcal{V}$). In an urban environment, traffic flow is typically observed between every origin-destination pairs (O-D pairs), and commonly there exist multiple paths connecting an origin and a destination. We denote the average flow from $r \in \mathcal{O}$ to $s \in \mathcal{D}$ over a certain time period (e.g., every Tuesday 6AM to 7AM) as $u_{rs}$, the paths from $r$ to $s$ as $J_{rs}$, and the flow on a single path $k \in J_{rs}$ as $u_{rs}^k$. Then, $u_{rs}$ is represented as follows:

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

Differing from the definition of a path flow, the flow of a road segment $f_e, e \in \mathcal{E}$ aggregates all path flows traversing $e$:

$$f_e = \sum_r \sum_{s} \sum_{k \in J_{rs}} \delta_{\mathcal{e}}^k u_{rs}^k, \forall r \in \mathcal{O}, s \in \mathcal{D}, \quad (2)$$
where $\delta^k_e$ is an indicator variable denoting whether $k$ contains $e$. By arranging all O-D pairs in $u = [..., u_{ij}, ...]'$ and the assignment proportions for $e$ from all O-D pairs as $P_e = [..., p_{eij}, ...]'$, we have $f_e = P_e u$. More compactly, by denoting all road-segment flows as $f = [..., f_e, ...]'$, $e \in \mathcal{E}$, and the assignment proportions for each $e$ as $P = [..., P_e, ...]'$, $e \in \mathcal{E}$, we obtain the general form of flows of road segments and O-D pairs in a network as $f = Pu$ where $P$ is termed assignment matrix.

In principle, once $u$ is appropriately assigned, we can compute travel times and flows of all road segments by solving the traffic assignment problem. One programme commonly used is the Wardrop’s principle (20) (a.k.a. user equilibrium) taking the following form:

$$\min z(f) = \sum_{e \in \mathcal{E}} \int_0^{f_e} t_e(\omega)d(\omega)$$

s.t. $u_{rs} = \sum_{k \in \mathcal{E}_{rs}} u^k_{rs}, \forall r \in \mathcal{O}, s \in \mathcal{D}$,

$f_e = \sum_r \sum_s \delta^k_e u^k_{rs}, \forall e \in \mathcal{E}$,

$u_{rs} \geq 0, \forall r \in \mathcal{O}, s \in \mathcal{D}$,

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

Traditionally, O-D pairs are obtained through a large-scale survey which is conducted infrequently due to prohibitive cost. Recent approaches have been taking traffic counts provided by monitoring sensors (e.g., loop detectors and video cameras) into account (2), as these sensors provide up-to-date traffic information. To illustrate the idea formally, we denote the target O-D pairs $\tilde{P}$ which are obtained from existing data as $\tilde{u}$ and observed flows from traffic sensors as $\tilde{f} = [..., \tilde{f}_a, ...]'$, $a \in \mathcal{A}$, where $\mathcal{A}$ denotes a set of road segments installed with traffic sensors. The estimated O-D pairs $\tilde{u}$ through estimated road-segment flows $\tilde{f}$ can be computed by solving the following formula (3):

$$\min_{\tilde{u}} F_1(\tilde{u}, \tilde{u}) + F_2(\tilde{f}, \tilde{f})$$

s.t. $\tilde{f} = M(\tilde{u}), \tilde{u} \geq 0$.

where $F_1$ and $F_2$ represent arbitrary distance functions. If $M$ takes the form of Equation (3), then Equation (4) becomes nested optimization (21). The upper Level tries to minimize differences between estimated OD pairs and traffic flows with measurements while the lower Level ensures that estimated quantities conforming observed traffic relationships, i.e. Equation (3). The key element of Equation (4) is $\tilde{f}$, as it represents up-to-date traffic monitoring results. The quantity and quality of $\tilde{f}$ greatly affect the estimation accuracy of $\tilde{u}$ and $\tilde{f}$. According to (14), regardless of the quality of $\tilde{u}$, better estimations are achieved as the number of monitored road-segments (i.e., $|\tilde{f}|$) approaches the number of O-D pairs (i.e., $|\tilde{u}|$). In reality, however, stationary traffic sensors such as loop detectors and video cameras are installed mostly on major roads and highways – which constitute only a small portion of a city. Since the number of O-D pairs is roughly proportional to the number of road segments in a city network, traditional techniques relying on the stationary sensors are insufficient to estimate traffic flows at city scale (2,22).

TRAFFIC CONDITIONS ESTIMATION
Our goal is to estimate traffic conditions of a city-scale network. We explicitly address two challenges presented by GPS data: the low-sampling rate and the spatial-temporal sparsity through three steps: coarse inference, iterative refinement, and nested estimation. Traffic is commonly assumed to be quasi-static (9) and has a weekly period (8). Based on these observations, we divide an entire week into discrete time intervals and treat the traffic within each interval as static. In this section, we focus our discussion on the three steps over a single time interval. The temporal missing values over an entire traffic period are interpolated using the technique developed in (13).

There are three key differences between our work and previous studies. First, for addressing the low-sampling-rate data, we use an iterative refinement rather than a sequential computation so that errors of the map-matching and travel-time inference processes are gradually reduced. Second, for addressing the spatial sparsity, we treat probe vehicles as mobile traffic sensors. By incorporating the sensing results generated from a large number of probe vehicles into the traffic assignment program (i.e., Equation (4)), 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 knowledge of the transportation engineering. These robust results allow our modeling of traffic one step closer to the real-world traffic.

Coarse Inference
The network travel times are intrinsically stochastic due to many factors: fluctuations in traffic demand, varying weather conditions, and heterogeneous driver behaviors. Because of these factors, we model the travel time of a road segment as a random variable which subjects to a probability distribution $Z_e$ parameterized by $\theta_e$. Our objective is to estimate $Z$, the joint distribution of travel times of road segments with GPS data coverage.

We obtain the coarse inference of $\mathbb{E}(Z)$ by solving a convex optimization program. According to Wardrop's principles (20), we treat the time difference $\Delta t$ 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 need to raise $\{t_e\}_{e \in k}$ until collectively $t_k \geq \Delta t$. Denoting $\mathbb{E}(Z)$ as $t = [t_1, t_2, \ldots, t_{|\mathcal{E}|}]'$ and corresponding indicator variables as $\xi = [\xi_1, \xi_2, \ldots, \xi_{|\mathcal{E}|}]$, where $\xi_e = 1, \forall e \in k$, and $\xi_e = 0, \forall e \notin k$, this constraint is denoted as $\xi' t \geq \Delta t$. The collection of constraints, $\mathcal{J}$, then takes the form $B't \geq \mathcal{H}$, in which $B_{|\mathcal{J}| \times |\mathcal{E}|}$ consists of $\{\xi_j\}_{j=1, \ldots, |\mathcal{J}|}$ and $\mathcal{H}_{|\mathcal{J}| \times 1}$ consists of $\{\Delta t\}_{j=1, \ldots, |\mathcal{J}|}$. To set upper and lower bounds, the travel time of a road segment $t_e$ is ceiled by the free-flow travel time, $t_{e,\text{min}}$ (taking 120% of the speed limit), and the jam-density travel time, $t_{e,\text{max}}$ (taking speed 0.5m/s). We further propose a regularization term, $\mathcal{D}$, as a 2-dimensional fused Lasso penalty (19) to model the traffic correlation of nearby road segments by enumerating all pairs of road segments at each $v \in \mathcal{V}$. The final convex optimization program is as follows:

$$\min_\mathcal{J} ||\mathcal{D}||_1$$

s. t. $B't \geq \mathcal{H}, t_{\text{min}} \leq t \leq t_{\text{max}}$.  

Iterative Refinement
We refine the initial estimated results using a nested iterative process. The outer loop alternates between map-matching and travel-time inference, and the inner loop – which resides in travel-time inference– performs an expectation-maximization (EM) algorithm. The rationale 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 on a network, we perform a simple method for map-matching: first, for a GPS points, a set of candidate mapping positions $Q$ (e.g. $|Q| = 5$) including nodes and locations on road segments is formed based on their distances to $s$; then, for any pair of candidates of consecutive GPS points $q_i \in Q_i$ and $q_{i+1} \in Q_{i+1}$, 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 $Δt(s_i, s_{i+1})$ is treated as the "true" path $\bar{k}$. The result of map-matching is a collection of aggregate measurements, denoted as $\mathcal{N} = \{(k_r, t_r)\}_{r=1,..,|\mathcal{N}|}$. In order to estimate $Z$ given $\mathcal{N}$, we set up the learning problem of $\theta = \{\theta_e\}_{e \in \mathcal{E}}$ as a maximum likelihood estimation (MLE) program:

$$\max_{\theta} \mathcal{L}(\theta | \mathcal{N}) = \sum_r \log \pi(\Delta t | k_r; \theta), \quad (6)$$

where $\pi$ represents the distribution for modeling travel time. We assume $\pi$ is a univariate distribution and travel-time distributions are pairwise independent (7). With these assumptions, we can factorize Equation (6) as follows:

$$\sum_r \log \pi(\Delta t_r | k_r; \theta) = \sum_r \log \int \pi(\Delta t_r | Z, k_r) \pi(Z; \theta) dZ =$$

$$\sum_r \log \int \pi(\Delta t_r | Z, k_r) \prod_e \pi(Z_e; \theta_e) dZ = \quad (7)$$

Equation (7) can be solved via an EM algorithm. In the E-step, for each $(k_r, \Delta t_r)$, $\Delta t_r$ is decomposed and distributed to road segments to suffice $\xi^t \Delta t^* = \Delta t_r$ where $\Delta t^*$ is a possible decomposition of $\Delta t_r$. In addition, the likelihood values, $w^*$, are computed based on current $\theta$. We call $(\Delta t^*, w^*)$ a random allocation. The realization of $\Delta t^*$ is essentially a fulfillment of the conditional expectation calculation and can be generated infinitely. In the M-step, all random allocations generated using $\mathcal{N}$ are gathered and regrouped by individual road segments. For a road segment $e$ with regrouped samples $\{(t_{e,i}, w_{e,i})\}_{i=1,..,d}$, we learn the parameter $\theta_e$ with the formula:

$$\max_{\theta_e} \sum_i w_{e,i} \log \pi(t_{e,i}; \theta_e). \quad (8)$$

To solve Equation (8), we select $\pi$ to be the Gamma distribution. The conditional sampling to suffice $\xi^t \Delta t^* = \Delta t_r$ using the shape factor $\alpha$ and the scale factor $\beta$ of $\pi$ is realized as follows (9):

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

where $\{t_{e,i}\}_{i=1,..,|\mathcal{E}|} = t^*$ and $\{\xi_{r,e}\}_{e=1,..,|\mathcal{E}|} = \xi_r$. For initializing $\theta_e = (\alpha_e, \beta_e)$, we take $E(Z)$ as the expectation and 60s as the standard deviation. After solving Equation (8) for all road segments with GPS data coverage, we have obtained the estimation of $Z$ and our framework performs map-matching again. The iteration continues until certain stop criteria are reached (e.g., 10 iteration runs). In the computation, the conversion between travel times and flows is attained by inverting the road-segment performance function proposed by the U.S. Bureau of Public Roads as:

$$t_e = t_{e,\text{min}} \left( 1 + \left( \frac{3f_e \delta}{2e_e} \right)^{\delta} \right), \forall e \in \mathcal{E}, \quad (10)$$
where $c_e$ is the capacity of a road segment $e$ computed as $c_e = 1700 + 10t_{e,\min}$ if $t_{e,\min} \leq 70\text{mph}$; $c_e = 2400$, otherwise (this formula is taken from fhwa.dot.gov).

**Nested Optimization**

The goal of this step is to estimate traffic flows and travel times on road segments that are not covered by GPS data. The estimations are obtain by solving Equation (4) with inputs $\bar{u}$ (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 Equation (3), $\bar{f}$ is obtained using estimated travel times from the previous steps by treating probe vehicles as mobile sensors.

Following (2), we select the distance functions in Equation (4) to be the generalized least squares (GLS) estimator. By further assuming $\bar{u}$ and $\bar{f}$ are results of the following stochastic system of equations:

$$\bar{u} = \bar{u} + \epsilon_1, \bar{f} = \bar{f} + \epsilon_2,$$

Equation (4) becomes:

$$\min_{\bar{u}} \eta (\bar{u} - \bar{u})'U^{-1}(\bar{u} - \bar{u}) + (1 - \eta)(\bar{f} - \bar{f})'V^{-1}(\bar{f} - \bar{f})$$

s.t. $\bar{f} = \mathcal{M}(\bar{u}), \bar{u} \geq 0, \bar{f} \geq 0, 0 \leq \eta \leq 1,$

where $U$ and $V$ are variance-covariance matrices of $\epsilon_1$ and $\epsilon_2$, respectively. We take $\mathbb{E}(\epsilon_1) = 0$ and $\mathbb{E}(\epsilon_2) = 0$ (3). $\eta$ decides the importance of $\bar{u}$ and $\bar{f}$ on the estimation results. We solve Equation (12) iteratively using quadratic programming and the Frank-Wolfe solver (11).

**EXPERIMENTS**

In order to evaluate our approach, we use the road network from downtown San Francisco (obtained from openstreetmap.org) as the benchmark. The network contains 5,407 nodes, 1,612 road segments, and 296 TAZs (obtained from data.sfgov.org). We have also generated a set of heuristic network travel times and a set of vehicle paths inferred from the Cabspotting dataset (obtained from crawdad.org) as the ground truth.

We generate one set of heuristic network travel times by solving the system optimal (SO) model (18). The SO model addresses traffic assignment by minimizing the entire travel time of a road network. The solution is a set of flows and travel times of all road segments in a network. The Cabspotting dataset, which we used to generate the travel times, only represents partial network flows, we multiply the results by 10 constants to correspond to 10 congestion levels ranging uniformly from 0.19 to 1.85. The congestion level is measured by volume over capacity (VOC) and computed as $\sum_{e \in E} \frac{f_e}{c_e}$. We generate another set of heuristic network travel times based on GPS timestamps. Using the Cabspotting dataset, we equally distribute the time difference of a pair of GPS points to all paths that connect them. For road segments that are covered by multiple GPS traces, the average travel times are used. Using this approach, we have produced 24 network travel times representing 24 hours of a typical weekday. We refer to this method of generating network travel times as the *Timestamp* model.

Using the established network travel times, we randomly simulate 20 collections of GPS traces. Each collection contains 30 sets GPS traces and all sets in a collection share the same number of traces. The sharing number for the 20 collections goes from 50 to 1,000 in increments of 50. As a result, we have produced over 10 million 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. To mimic features of the Cabspotting dataset, the sampling rate is set to be 60s, and all coordinates are perturbed by the Gaussian noise $(0,20m)$ (25).

**Evaluation and Comparison**

We evaluate our technique by comparing to Hunter et al. (9) and Rahmani et al. (17). The first method is equivalent to the inner loop of our travel-time inference process. 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 (9). The second method takes a non-parametric perspective, using a kernel-based technique to estimate travel times. The weights used to allocate travel times to individual road segments are set to be the ratio of free-flow travel times among road segments (5).

We set parameters of our nested iterative process as follows: retaining the same settings for the inner loop as in (9), we empirically set the number of iterations for the outer loop to 10. This setting is based on results shown in Figure 2, where the relationship between the normalized convergence rate (%) and the number of iterations for both types of network travel times is plotted. Each datum in the plot is the average value computed using all network travel times across all sets of sampled GPS traces of either the SO model (6,000 trials) or the Timestamp model (14,400 trials). The measurement of each trial is the mean square error (MSE) between a recovered and a ground-truth traffic condition (i.e., \( \frac{\sum (t_e - \bar{t}_e)^2}{|E|} \)). As a result, the convergence rate decreases quadratically as the number of iterations increases and tends to flatten after 10 iterations.

**FIGURE 2** The relationship between the normalized 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 flatten after 10 iterations.

In Figure 3, we demonstrate the evaluation of our technique under three measurements. The first metric is the performance gain of our technique over existing methods on travel times by considering all road segments of a network. We compute this metric based on \( MSE = \frac{\sum (t_e - \bar{t}_e)^2}{|E|} \) as \( \frac{MSE_{existing} - MSE_{our}}{MSE_{our}} \), where \( MSE_{our} \) represents the error between a recovered traffic condition using our technique and the ground-truth traffic condition, and \( MSE_{existing} \) represents the error computed using an existing method with the same ground truth. The maximum relative improvements over Hunter et al. (9) and Rahmani et al. (17) under the SO model are 78% and 97% (Figure 3 TOP-LEFT), and under the Timestamp model are 54% and 49% (Figure 3 TOP-RIGHT),
respectively. In general, with more GPS traces used in estimation, better performance gains are achieved. Such effects are more apparent on the SO model than the Timestamp model.

The second metric is the error rate of the aggregate travel time of the entire network, computed as \( \frac{\sum_{e} \bar{t}_e - \sum_{e} t_e}{\sum_{e} t_e}, \forall e \in E \), where \( \bar{t}_e \) represents an estimated travel time and \( t_e \) represents a ground-truth travel time. Figure 3 MIDDLE-LEFT shows the results by averaging the experimental outcomes of all network travel times via the SO model. The minimum error rate of our technique is 18%, of Rahmani et al. (17) is 34%, and of Hunter et al. (9) is 48%. Experimenting on network travel times generated via the Timestamp model shown in Figure 3 MIDDLE-RIGHT, the corresponding minimum error rates are 8%, 28%, and 37%, accordingly. 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 regards the map-matching accuracy by computing the success rate as

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

We sum all success rates generated using our method and an existing approach, and derive the performance gain as \( \frac{\sum SR_{\text{our}} - \sum SR_{\text{existing}}}{\sum SR_{\text{our}}} \). The maximum gain of our method over Hunter et al. (9) and Rahmani et al. (17) under the SO model are 28% and 34%, and under the Timestamp model are 19% and 25%, correspondingly. These results are shown in Figure 3 BOTTOM. Again, as the number of GPS traces used in recovering network travel times increases, gains in the improvements are observed.

**FIGURE 3** From LEFT to RIGHT, LEFT diagrams show results via the system optimal (SO) model, while RIGHT diagrams show results via the Timestamp model. TOP: The
performance gain (%) of network travel times measured in Mean Squared Error (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 sets of synthetic GPS traces. In summary, our technique achieves consistent improvements over the other two methods.

Analysis of Nested Optimization

We have shown that our approach outperforms existing methods on estimating travel times of a road network. In turn, by inverting Equation (10), we can obtain better estimations of target flows – which serve as inputs to the nested optimization.

FIGURE 4 TOP and MIDDLE: The normalized noisy levels (%) of target road-segment flows $\tilde{f}$ computed according to MSE of estimated network travel times to ground truth. In general, for both models, our technique produces lower noisy levels than other two techniques. BOTTOM: The normalized MSE of target O-D pairs $\bar{u}$ (%) under different values of the weighting factor $\eta$ and various noisy levels of $\tilde{f}$ (%). When $\eta$ is small, the error is more sensitive to perturbations on $\tilde{f}$. Overall, the error increases as the noisy level of $\tilde{f}$ increases. For all studies, the normalized noisy level of target O-D pairs $\bar{u}$ has been set to
Our method has achieved consistently lower error rates as compared to other methods. The factors affecting the program are the weighting factor η, the noisy level of target O-D pairs \( \tilde{u} \), and the noisy level of target road-segment flows \( \tilde{f} \). The noises of \( \tilde{u} \) and \( \tilde{f} \) are assumed to have zero means and diagonal variance-covariance matrices (3). In reality, the noisy level of \( \tilde{u} \) is difficult to assess because not only \( \tilde{u} \) usually comes from existing data but also the true values of \( \tilde{u} \) are seldom (3). For this reason, in the analysis of η and the noisy level of \( \tilde{f} \), we set the normalized noisy level of \( \tilde{u} \) to 50%. Subsequently, the normalized noisy levels of \( \tilde{f} \) (%) computed based on MSE of estimated travel times to ground truth are shown in Figure 4 LEFT and MIDDLE. In general, our technique produces lower noisy levels of \( \tilde{f} \) than other two techniques, especially under the SO model which is considered to be a better approximation to real-world traffic than the Timestamp model (18).

In order to evaluate how η and the noisy level of \( \tilde{f} \) affect the estimation accuracy of \( \tilde{u} \) (i.e., the estimated O-D pairs), we compute the normalized MSE of \( \tilde{u} \) (%) under different η and various noisy levels of \( \tilde{f} \) (%). The results are shown in Figure 4 RIGHT. When η takes a small value (e.g., 0.1), the impact of \( \tilde{u} \) is restricted, thus the MSE of \( \tilde{u} \) reacts actively to perturbations on \( \tilde{f} \). As we gradually increase the value of η, the impact of \( \tilde{f} \) attenuates. Nevertheless, the MSE of \( \tilde{u} \) increases as the noisy level of \( \tilde{f} \) progresses.

FIELD TESTS
We conduct field tests in two diverse cities in two continents, namely Beijing and San Francisco, to study urban traffic dynamics on a metropolitan scale. The GPS datasets used in the field tests are from the Cabspotting project and T-drive project (24), respectively. The TAZ file of San Francisco is obtained from data.sfgov.org and the area considered is from <37.7083,-122.514> to <37.812,-122.358> (in latitude and longitude). The TAZs of Beijing are constructed by dividing the area from <39.8043,116.1904> to <40.035,116.5673> – into equal-sized 1km by 1km grids. The target O-D pairs are estimated using the technique from (22) with the GPS dataset of each city.

![FIGURE 5](image_url) The estimated traffic conditions measured in average volume over capacity (VOC) throughout the two cities, San Francisco and Beijing, for various types of roads. The computation is 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 of a week indicating mid-day traffic relief. Such phenomena are not observed in Beijing which suggests the congestion tends to form throughout the day time. The similarity of traffic patterns among various road types in both cities indicates their similar usage as road infrastructure.

The results shown in Figure 5 demonstrate the estimated traffic conditions measured in average volume over capacity (VOC) throughout the two cities. All computations are conducted using the epoch time. From the result, we can see that the recovered traffic patterns show clearly periodic phenomena over the course of a week – this feature is considered as one of the hallmarks of traffic (8) – for both overall roads and decomposed road types. In addition, in San Francisco, we observe saddle shapes corresponding to mid-day traffic relief over several days of a week. Such phenomena are more evident on major roads of San Francisco (i.e., motorway and truck), but not on the rest types of roads – on which the traffic patterns are similar indicating their similar usage as transportation infrastructure. In comparison, in Beijing, we don't observe such saddle shapes appearing in the middle of a day which suggests that congestion remains severe throughout the day time. Moreover, all types of roads of Beijing share similar traffic patterns indicating their similar usage in traffic.

**FIGURE 6 Field tests in Beijing.** Different colors represent different ranges of volume over capacity (VOC). Four time periods, namely Sunday 9AM, Tuesday 9AM, Thursday Noon, and Friday 7PM are displayed as examples to illustrate weekend vs. weekday and morning vs. evening traffic.

In Figure 6, we show detailed estimation results over four time intervals: Sunday 9AM representing *weekend morning traffic*, Tuesday 9AM representing *weekday morning traffic*, Thursday Noon representing *weekday mid-day traffic*, and Friday 7PM representing *weekday evening traffic*. In Beijing, 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 9AM, especially considering the traffic between the 4th and the 5th ring roads, where more residential units are found than the region inside the 4th ring road.

**CONCLUSION AND FUTURE WORK**
We have presented a novel framework for estimating urban traffic conditions using traffic flow theory and GPS traces. First, we obtain the coarse inference of network travel times using convex optimization. Next, we take an iterative approach to jointly estimate traversed paths of probe vehicles and travel times of road segments in a network. Third, we incorporate the previous estimated results into a nested optimization program to estimate traffic conditions on road segments that are not covered by GPS data. By adopting Compressed Sensing to interpolate the temporal missing values, we have obtained citywide traffic dynamics over an entire traffic period.
Our approach has been evaluated using a real road network resulting in consistent and notable improvements over state-of-the-art methods. In order to understand 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. Examples can be found in (31).

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

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

REFERENCES
30. Li, W., D. Wolinski, and M. Lin. City-Scale Traffic Animation Using Statistical Learning and Metamodell-Based Optimization. *The 10th ACM SIGGRAPH Conference and