2018 IEEE 34th International Conference on Data Engineering

Realtime Traffic Speed Estimation with Sparse Crowdsourced Data

Zheng Liu, Lei Chen, Yongxin Tong

[#] *Department of Computer Science and Technology, HKUST, Hong Kong SAR, China*
[1] zliual,[2] leichen@cse.ust.hk

[*] *SKLSDE Lab, NSTR and IRI, Beihang University, China.*
[3] yxtong@buaa.edu.cn

Abstract—**Realtime traffic speed estimation is an important issue in urban computation. Existing approaches usually focus on exploiting the periodicity properties of the traffic speed and utilize crowdsourcing techniques to facilitate real-time estimation. The quality of such estimation is limited in real world: 1) the accuracy of existing estimation over-relies on the probed data; 2) the accidental traffic variance is ignored; 3) existing strategies incur exhaustive usage of human workers to get fine-grained estimation results. Thus, a more intelligent RTSE approach is desired. In this paper, we propose the framework of CrowdRTSE (Crowdsourcing-based Real-time Traffic Speed Estimation), which adopts a hybrid offline-online process to collaboratively exploit the historical and real-time data to produce high-quality RTSE. To accomplish such a framework, we devise effective algorithms to judiciously select the best group of human workers with a constant approximation ratio, and effectively propagate the crowdsourced data with high efficiency. Comprehensive evaluations have been conducted on both synthetic and real world datasets. The experimental results verify the effectiveness and efficiency of our proposed methods.**

I. INTRODUCTION

Realtime traffic speed estimation (RTSE) is a crucial component in many urban applications, such as traffic surveillance, route planning, accident detection, and so on. Given a set of queried roads within a traffic network, RTSE estimates the realtime traffic speed for the corresponding queries. Despite the deployment of various traffic monitoring and participatory sensing devices (e.g. inductive sensors and mobile GPS devices [1], [2]), it is still difficult to obtain high-quality realtime traffic speed estimation for the entire interested area, as a result of the limited coverage of sensor deployment and the concern of privacy.

In recent years, thanks to the availability of high-volume offline traffic data, like realtime speed record and trajectories, numerous data-driven techniques have been developed for RTSE. To improve the estimate quality, researchers mainly focus on exploring and utilizing the two statistical properties of traffic data: **periodicity** and **correlation**. On one hand, considering that there is always a normal pattern for the traffic speed on one road, the periodicity can be captured using the historical data and RTSE can be therefore inferred with the periodic property [3], [4], [5]. However, as these periodicity based methods can only predict the overall trend of traffic speed, they are incapable of predicting the accidental variations and it is difficult to produce fine-grained result for the RTSE. On the other hand, given the fact that roads are connected with each other through the traffic network, the speeds of different roads are highly correlated. By making use of the correlation, the traffic speed on one queried road can be estimated with other probed data (realtime traffic speed) from the road network [6], [7]. While the performance of the correlation based approaches highly depends on the quality of probed data: if the probed data is inadequate, or has weak correlations with the queried roads, the estimation quality will be inevitably coarsened. As such, based on the above analysis, it can be inferred that only relying on the periodicity or correlation is insufficient to produce quality estimations. However, to the best of our knowledge, currently there is no approach that makes joint usage of the periodicity and correlation to generate quality results for RTSE.

Inspired by the rapid development of crowdsourcing services, such as Field Agent, CheckPoints and OpenStreetMap, a promising option of enhancing the current RTSE techniques is to utilize the power of crowd. Given a specific realtime speed query, we can ask the human workers to report the speed measurement of certain locations and the realtime speed for the queried road can be estimated based on the crowdsourced data. Although there are some preliminary works proposed along this direction, such as [8], [9], their practical applications have been seriously hindered by some methodology defects and unrealistic assumptions. Firstly, the existing techniques on realtime traffic speed estimation cannot effectively work with the crowdsourced data. Secondly, the inherent periodicity property of the traffic road has been overlooked by these existing methods: in fact, for those roads of strong periodicity, their realtime traffic speed can be effectively estimated with little assistance from crowdsourcing. As crowdsourcing services are normally conducted with a limited budget, such a limitation on resource allocation will easily make the existing methods economically impractical to handle the RTSE tasks. Lastly, most existing methods implicitly require workers to travel physically so as to collect the desirable crowdsourced data. However, such a requirement will inevitably incur unacceptable delay and reduce workers' willingness to execute the tasks.

To deal with the above challenges, we propose CrowdRTSE (**Crowd**sourcing-based **R**ealtime **T**raffic **S**peed **E**stimation),

2375-026X/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDE.2018.00038

which produces high-quality RTSE with a modest budget. To be specific, CrowdRTSE works with a hybrid offline-online framework, as shown in Fig. 1. In the offline stage, a probabilistic graphical model, RTF (**R**ealtime **T**raffic-Speed **F**ield), is constructed in the form of Gaussian Markov Random Field (GMRF). By exploring the historical data, RTF seamlessly bridges the inherent topology of traffic network and the statistical properties (i.e., periodicity and correlation) of traffic speed, which is also the foundation for online-processing. In the online stage, two main steps are involved to produce RTSE for a presented query. In the first step, CrowdRTSE needs to select some roads from the pool of roads currently with workers distributed (referred as crowdsourced roads), aiming to allocate the crowdsourcing resource in an effective way. To optimize the selection, the OCS (**O**ptimal **C**rowdsourced Road **S**election) problem is solved to find the best group of crowdsourced roads w.r.t. the presented query and the given budget. Then the realtime traffic speed is sampled for these selected roads (referred as sampled roads). In the second step, using the data collected from sampled roads, the realtime traffic speed can be inferred for all non-sampled roads. Seeking for more reliable inferences, GSP (**G**raph-based **S**peed **P**ropagation) is developed on top of RTF, which exploits the statistical properties to obtain the most likely speed for non-sampled roads.

The proposed CrowdRTSE can handle the above difficulties properly, including the challenges to utilize the periodicity and correlation in a collative way, the limitations to incorporate with crowdsourced data, the deficiency in crowdsourcing resource allocation and the restrictive requirement of physical movement. To summarize, the following contributions are made in this work.

- We propose a novel framework, CrowdRTSE, which collaboratively exploits the historical data and the crowdsourcing data to produce high quality RTSE with a modest budget.

- A graphical model RTF is constructed to encode the topology structure of traffic network and statistical properties, i.e., periodicity and correlation, of traffic data. Quality speed estimation can be obtained based on this elegant graphical model. To allocate crowdsourcing resource in an effective way, the OCS problem is proposed, which selects the optimal set of crowdsourced roads w.r.t. the presented query and budget. The OCS problem is proved to be NP-hard, and effective algorithms are designed to find its approximate solution.

- We verify the effectiveness and efficiency of the proposed methods through extensive experiments on both real and synthetic datasets.

The rest of the paper is organized as follows. The related work is discussed in Section II and the problem overview is made in Section III. Then, the formulation of RTF is presented in Section II. After that, the optimal crowd selection and speed propagation are discussed in Section V and VI, respectively. Finally, we report the experimental results in Section VII and conclude in Section VIII.

II. Related Work

In this section, we discuss the related work under two categories: the techniques for traffic speed estimation and the crowdsourcing assisted computation.

A. Traffic Speed Estimation

To estimate the realtime traffic speed, the regression based approaches are extensively exploited in the existing works, such as [3], [10], [6], [11], [12], [13], [14], [9]. Regardless the adoption of different statistical modeling (e.g., Linear Regression, SVR, Neural Network), all the regression based approaches assume the constant correlation a fixed set of observation variables (e.g., the realtime traffic speed of a set of observation roads) and the realtime traffic speed of all the interested roads, and the model parameters can be inferred from the historical records. Because of the capability of capturing the complex correlation between the traffic speed of different roads, the regression based approaches are proved to be effective in making accurate estimation for the realtime traffic speed. However, the significant limitations about such methods are two-fold. Firstly, because of the sparse connection of the traffic network, each road may only closely correlated with a few neighboring roads (as discussed in [9]). Therefore, to make accurate estimation, the number of observation sites must be large enough, which enables the close correlation with all the interested roads. However, when the number of observation sites is small, the estimation result will be severely coarsened. Secondly, the regression based methods simply model the correlation between the interested roads and a fixed set of variables. Although it works well for scenarios where the data is collected from the deployed loop sensors or cameras (whose positions are fixed), it is not suitable for the scenario of crowdsourcing where the data is usually collected from unfixed locations (because the workers' distribution is time variant). To make of the regression based methods, [9] implicitly asks workers to travel physically to the selected observation sites. However, such operation leads to extra time cost (which is crucial in making realtime estimation) and reduces the worker's willingness to carry out the task.

An alternative way to estimate the realtime traffic speed relies on the technique of matrix completion ([15], [16]). Let each road crossing (or end) specify a unique row and column of a matrix M, the realtime traffic speed of one road can be represented by a specific entry of M. With the realtime traffic speed collected for a certain set of roads, the missing values of M can be recovered through matrix completion. And to improve the estimation quality, the Graph Laplacian factor ([17]) is usually added to enforce the spatial smoothness. Apparently, such methods is free from the requirement of fixed observation sites. However, the statistical properties of the historical data is not fully effectively captured, which significantly harms its estimation accuracy.

Different from the existing works, our proposed approach collaboratively exploits the correlational and periodic properties of the traffic speed, which is able to produce accurate estimation even with a limited size of realtime crowdsourced

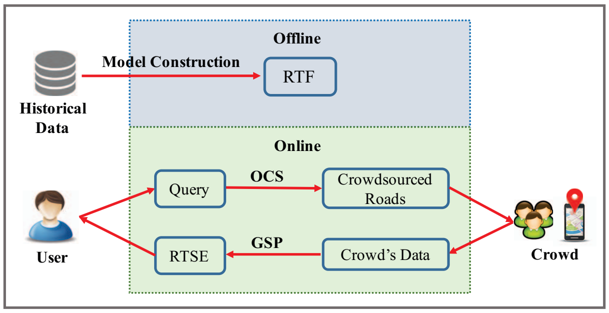


Fig. 1: System Infrastructure. The blue dash box and green dash box demonstrate the offline and online stage, respectively.

data. Besides, the proposed approach naturally works with the crowdsourced data, and it is exempted from the restriction of fixed observation sites.

B. Crowdsourcing Assisted Computation

With the quick development of crowdsourcing services, e.g., Amazon MTurk and Yahoo Answer, human intelligence is effectively employed for many challenging problems, like entity resolution [18] and data cleaning [19]. Besides, crowdsourcing is also widely applied to many spatial applications, where workers' are paid to perform spatial tasks traffic surveillance [20], [9], [8], [21]. However, the existing works cannot fully address the problem of realtime traffic estimation due to the following limitations: firstly, there is no effective realtime speed estimation approach designed to work with the crowdsourced data; secondly, the allocation of crowdsourcing resource is not optimized in practice.

III. Problem Overview

A. Preliminaries

Traffic Network A traffic network is composed of two basic components: the roads (R) and the adjacency relationship (E). In this work, each road represents a unique isolated interval of path jointing two adjacent crossings (or the end of a path). In other words, each road is an atomic unit, which contains no other roads as its sub-components. For each road, we assume there is a unique realtime traffic speed associated with it, which can be comprehended as the average traffic speed of the road for the current timestamp. The given traffic network is formulated on a undirected graph $\mathrm{N}\{R, E\}$, where the vertexes R denote the universal set of roads and the edges E represent the adjacent relationship.

For a given traffic network $\mathrm{N}(R, E)$, a realtime traffic speed query (Q) is launched to a certain set of roads R^q, whose realtime traffic speed is returned by the CrowdRTSE.

Crowdsourcing Crowdsourcing is utilized to probe the realtime traffic speed for a small fraction of roads, based on which the realtime traffic speed estimation is made to the given query. While conducting the crowdsourcing, each worker has to make a task demand to the system and provide her localization information. Once a worker is selected by the system, she will be allocated with a task, which asks her to report the realtime traffic speed of her current location. Since most modern mobile devices have the ability to detect the realtime traveling speed, the workers will submit their answers easily. If a worker's answer is successfully submitted, she will be paid with a predefined payment (or credit) as reward.

B. System Infrastructure

The architecture of CrowdRTSE is sketched in Figure 1, with major components presented as follows.

Realtime Traffic Speed Field (RTF) RTF is a probabilistic graphical model, which is constructed under the framework of Gaussian Markov Random Field. RTF shares the same topological structure of the given traffic network, and the model's parameters are inferred with the historical record of traffic data. The function of RTF is two-fold. On one hand, RTF captures the speed periodicity of each road and the speed correlation between each pair of adjacent roads. Based on such information, the best group of crowdsourced roads is selected. On the other hand, a belief-propagation based approach, namely Graph-based Speed Propagation, is devised on top of RTF, which exploits the crowdsourced data to estimate realtime traffic speeds for the whole road network.

Optimal Crowdsourced Roads Selection (OCS) OCS selects the best group of roads, whose realtime traffic speed is probed through crowdsourcing. Specifically, the selection of the crowdsourced roads is conducted with the joint consideration of both utility and feasibility. From the perspective of utility, the crowdsourced data should help to increase the quality of RTSE as much as possible. From the perspective of feasibility, the crowdsourced data has to be collected from the roads where workers are currently distributed (which makes the workers free from extra traveling effort). Besides, the selection of the crowdsourced roads is subject to a given budget because of the economic concern.

Graph-based Speed Propagation (GSP) With the data collected from the crowdsourcing workers, the realtime traffic speed is inferred for the whole traffic network though speed propagation. Such a method is inspired by the previous works on belief propagation, which has been successfully applied to various problems, such as image segmentation and topic modeling. While making the inference, GSP iteratively refines the estimated speed of each road, which finally leads to the most credible result w.r.t. the collected data and RTF. Thanks to the full exploitation of the statistical properties, GSP significantly improves the quality of RTSE, especially for the cases where the size of collected data is small.

Workflow of CrowdRTSE CrowdRTSE works in a hybrid offline-online approach. In the offline part, the graphical model, RTF, is constructed based on the historical record of traffic speed. Such a model provides the foundation for the online processing. In the online part, a presented query is answered with three steps. Firstly, the system conducts OCS, which selects the best set of crowdsourced roads. Secondly, the crowdsourcing is launched, which probes the realtime traffic speed for all the crowdsourced roads. Thirdly, with data collected from the crowd, the speed propagation is performed to infer the traffic speed for the whole traffic network, which

R	the universal set of roads
r_i	a specific road
$n(r_i)$	the adjacent roads of r_i
E	the adjacency relationship of all roads
$\mathrm{N}(R, E)$	the traffic network of roads R and adjacency relationship E
R^q	the queried road segments
R^w	the set of roads where workers are distributed
R^c	the crowdsourced roads
v_i^t	the realtime traffic speed of road r_i at time slot t

TABLE I: General Notations

can produce the RTSE for all the queried roads as well. Then the RTSE is returned to the user as the answer to her query.

To ease the presentation, the frequent used symbols are summarized in Table I.

IV. Realtime Traffic Speed Field

A. Model Construction

In this paper, a graph model $G = (R, E)$ is proposed to reflect the topology structure of the traffic network. Each road r_i ($r_i \in R$) in the traffic network is regarded as one node in the graph. If two roads r_i and r_j are adjacent, there is an edge e_{ij} ($e_{ij} \in E$) connecting them. Following the conventional way of modeling temporal-spatial data, graph G is formulated under the framework of Gaussian Markov Random Field[22], such that fundamental statistical properties of traffic speed: periodicity and correlation, can be substantially captured.

Periodicity. Similar with previous works[5], [23], each day is divided into 288 fine-grained time slots so that each 5-minutes interval becomes a unique slot. Considering the recurrent pattern of traffic speed, similar value is expected for the same time-slot of different days. As such, for each road r_i in time slot t ($t \in T$), the road speed v_i^t follows Gaussian distribution:

$$v_i^t \sim \mathcal{N}(\mu_i^t,\ {\sigma_i^t}^2), \tag{1}$$

where μ_i^t is the expectation and σ_i^t is the standard deviation.

Correlation. The correlation of traffic speeds is captured with their differences. In particular, given that v_i and v_j are Gaussian variables, the following relationship can be derived[24]:

$$v_i^t - v_j^t \sim \mathcal{N}(\mu_{ij}^t,\ {\sigma_{ij}^t}^2), \tag{2}$$

where $\mu_{ij}^t = \mu_i^t - \mu_j^t$ and $\sigma_{ij}^t = \sqrt{{\sigma_i^t}^2 + {\sigma_j^t}^2 - 2\rho_{ij}^t \sigma_i^t \sigma_j^t}$. Here, ρ_{ij}^t quantifies the correlation between v_i^t and v_j^t, which acts as the edge weight in the graph ($\rho_{ij} \in [0, 1]$).

Based on Eq. (1) and (2), the conditional log likelihood of v_i^t given the traffic speed of $R \setminus \{r_i\}$ can be derived as:

$$\begin{aligned}
&\mathcal{L}(v_i^t \mid V_{R\setminus\{r_i\}}^t) \\
&= \log(\mathrm{P}(v_i^t \mid V_{R\setminus\{r_i\}}^t) / \int_{v_i^t=0}^{\infty} \mathrm{P}(v_i^t \mid V_{R\setminus\{r_i\}}^t)) \\
&= -\frac{(v_i^t - \mu_i^t)^2}{{\sigma_i^t}^2} - \sum\nolimits_{V_{R\setminus\{r_i\}}^t} \frac{[(v_i^t - v_j^t) - \mu_{ij}^t]^2}{{\sigma_{ij}^t}^2},
\end{aligned} \tag{3}$$

where the first item shows the speed's periodicity, and the second item indicates the correlation with other roads.

Notice that the road network is a flow system, where traffic condition of each road is determined by the status of itself and its direct-adjacent neighbors. Thus, given traffic speeds of r_i's direct neighbors, v_i^t is conditionally independent with v_j^t iff. r_i and r_j are non-adjacent, which makes Eq. (3) simplified as:

$$\begin{aligned}
&\mathcal{L}(v_i^t \mid V_{R\setminus\{r_i\}}^t) = \mathcal{L}(v_i^t \mid V_{n(r_i)}^t) \\
&= -\frac{(v_i^t - \mu_i^t)^2}{{\sigma_i^t}^2} - \sum\nolimits_{V_{n(r_i)}^t} \frac{[(v_i^t - v_j^t) - \mu_{ij}^t]^2}{{\sigma_{ij}^t}^2},
\end{aligned} \tag{4}$$

where $n(r_i)$ denotes the adjacent roads of r_i. In fact, similar operations are common in semi-supervised learning[25], [26], where structural information (i.e., correlations between vertexes) are utilized without introducing unnecessary parameters.

As each day has been divided into T time slots, a series of graphical models G^t ($t \in 1, \cdots, T$) are combined together to represent the realtime traffic speed field (RTF). There are two attributes associated with G^t: V_R^t and P_E^t, where $V_R^t = \{v_i^t \mid r_i \in R\}$ denotes the realtime traffic speeds for all roads and $P_E^t = \{\rho_{ij}^t \mid e_{ij} \in E\}$ represents the correlation coefficients for all pairs of adjacent roads. In addition, there are two auxiliary variables for each v_i^t: μ_i^t and σ_i^t, which represents the expectation and variance of the realtime traffic speed, respectively. Based on Eq. (4), the joint likelihood of G^t can be presented as follows:

$$\mathcal{L}_{G^t} = -\sum\nolimits_{V_R^t} \left(\frac{(v_i^t - \mu_i^t)^2}{{\sigma_i^t}^2} + \sum\nolimits_{V_{n(r_i)}^t} \frac{[(v_i^t - v_j^t) - \mu_{ij}^t]^2}{{\sigma_{ij}^t}^2} \right). \tag{5}$$

Remark 1. (Physical Meaning of RTF) The physic meaning of RTF is two-fold. On one hand, the graph-based RTF follows the same topological structure of the traffic network, as each vertex (or edge) of RTF corresponds to one unique road (or roads-adjacency) of the traffic network. On the other hand, both periodicity and correlation of the traffic speed are encoded within RTF. Firstly, the parameter μ_i^t captures the expected speed of road r_i within time slot t and the "intensity" of periodicity is reflected in the parameter σ_i^t. If the value of σ_i^t is small, it indicates that the traffic speed of r_i is stable within the time slot t, thus, the expected speed μ_i^t will be an effective approximation of v_i^t. Otherwise, it won't be appropriate to approximate v_i^t with μ_i^t. Secondly, the value of ρ_{ij}^t represents the "strength" of correlation: a larger value of ρ_{ij}^t means a closer correlation between the realtime traffic speed of road r_i and r_j. Therefore, the speed of one road can be effectively inferred with the knowledge of the other one.

B. Parameter Inference

In RTF, there are three sets of parameters to be inferred for the graph G, the expectations of V_R^t: $\mathrm{M} = \{\mu_i^t | r_i \in R, t \in T\}$, the standard variances of V_R^t: $\Omega = \{\sigma_i^t \mid r_i \in R,\ t \in T\}$, and the correlation coefficients between V_R^t: $\mathrm{P} = \{\rho_{ij}^t \mid e_{ij} \in E, t \in T\}$. Given the historical record of traffic speed (denoted as H), the parameter inference is conducted so that the joint likelihood $\mathcal{L}_G$ can be maximized accordingly:

$$\max_{\mathrm{M},\Omega,\mathrm{P}} \mathcal{L}_G(\mathrm{M}, \Omega, \mathrm{P} | \mathrm{H}). \tag{6}$$

Algorithm 1: Parameter Inference

```
input : H, $\lambda$
output: M, $\Omega$, P
begin
    Initialize: M, $\Omega$, P $\leftarrow$ small random values;
    while Not covnverge do
        for $\mu_i^t \in$ M do
            $\mu_i^t \leftarrow \mu_i^t + \lambda \frac{\partial \mathcal{L}_G}{\partial \mu_i^t}$;
        for $\sigma_i^t \in \Omega$ do
            $\sigma_i^t \leftarrow \sigma_i^t + \lambda \frac{\partial \mathcal{L}_G}{\partial \sigma_i^t}$;
        for $\rho_{ij}^t \in$ P do
            $\rho_{ij}^t \leftarrow \rho_{ij}^t + \lambda \frac{\partial \mathcal{L}_G}{\partial \rho_{ij}^t}$;
Return M, $\Omega$, P;
```

To achieve the maximization, partial derivatives of Eq. (6), i.e., $\frac{\partial \mathcal{L}_G}{\partial \mu_i^t}$, $\frac{\partial \mathcal{L}_G}{\partial \sigma_i^t}$ and $\frac{\partial \mathcal{L}_G}{\partial \rho_{ij}^t}$, are firstly calculated w.r.t. each of the parameters, based on which cyclic coordinate descent (CCD) [27] approach is adopted for optimization. To be specific, the parameters of M, Ω, and P are updated in a sequential manner: for each $x \in \mathcal{X}$, where $\mathcal{X} = \mathrm{M} \cup \Omega \cup \mathrm{P}$, the following gradient ascension is conducted: $x \leftarrow x + \lambda \frac{\partial \mathcal{L}_G}{\partial x}$, where λ is the step size. During each iteration, only one parameter is selected to be updated, with the rest parameters ($\mathcal{X}/x$) remained unchanged. The updating process is repetitively carried out until the convergence threshold (or the maximum number of iterations) is reached. The whole process of the parameter inference is summarized as Alg. 1.

Time Efficiency of Parameter Inference. The convergence of Alg. 1 has been discussed in [27], and it is clear that the time complexity of each update iteration is $O(|R|^2)$, where $|R|$ is the total number of roads within the traffic network. Denoting the maximum converging iteration with a constant $\mathcal{C}_v$, the overall complexity of Alg. 1 turns out to be $O(\mathcal{C}_v|R|^2)$.

V. Optimal Crowdsourced Roads Selection

In this section, we will first discuss the criteria for crowd (crowdsourced roads) selection and then provide the formal formulation of optimal crowd selection (OCS). A hybrid greedy-based algorithm is further developed, which can find the near-optimal solution for OCS within polynomial running time and the approximation ratio is strictly above $(1 - \frac{1}{e})/2$.

A. Formulation of OCS

Given a set of queried roads, we need to consider two significant factors during the selection of crowdsourced roads: the **periodicity** of queried roads and the **correlation** between the crowdsourced and queried roads. As explained in **Remark 1**, the road speed with strong periodicity can be effectively predicted with historical data, while those with weak periodicity need additional assistances and should be emphasized during OCS; as a closer correlation can benefit the speed inference between two roads, it is preferable that the correlation between the crowdsourced and queried roads can be maximized. Apart from these two desirable factors, the formulation of OCS also needs to meet the following two constraints: regarding to the limited budget, the selection of crowdsourced roads should satisfy the **feasibility** requirement; the **redundancy** among crowdsourced roads (the internal correlation between the crowdsourced roads) should be restricted for the sake of efficiency. In the following, we will give the detailed formulations of these factors and constraints.

Correlation. Here a general correlation is defined for three different scenarios: **road-road**, **road-set** and **set-set**.

The road-road correlation covers two cases: two roads are adjacent or non-adjacent. As mentioned above, given the RTF model, the road-road correlation between two adjacent roads, r_i and $r_{i'}$, can be measured with the edge weight. Specifically,

$$corr^t(r_i, r_{i'}) = \rho_{ii'}^t, \text{ iff. } e_{ii'} \in E. \quad (7)$$

For two non-adjacent roads r_i and r_j, the road-road correlation between them is measured with the maximal cumulative product of the edge weights along their joining paths:

$$corr^t(r_i, r_j) = \max_{\phi_{ij} \in \Phi_{ij}} \{\prod\nolimits_{e_{kl} \in \phi_{ij}} \rho_{kl}^t\}, \quad (8)$$

where Φ_{ij} is the universal set of joining paths between r_i and r_j. Let ϕ_{ij}^* denotes the path which leads to the optimization of Eq. (8), and it is straightforward to verify that ϕ_{ij}^* also satisfies the following relationship:

$$\phi_{ij}^* = \text{argmin}\{\sum\nolimits_{e_{kl} \in \phi_{ij}} 1/\rho_{kl}^t \mid \forall\ \phi_{ij} \in \Phi_{ij}\}. \quad (9)$$

In other words, ϕ_{ij}^* gives rise to the shortest path between r_i and r_j by converting the original edge weights to their reciprocals (i.e., $1/\rho_{ij}^t$). With ϕ_{ij}^* found using Dijkstra's Algorithm, Eq. (8) can be re-written as:

$$corr^t(r_i, r_j) = \prod\nolimits_{e_{kl} \in \phi_{ij}^*} \rho_{kl}^t. \quad (10)$$

The correlation calculation between each pair of roads is performed offline, whose result Γ_R $(\{corr^t(r_i, r_j) \mid \forall\ r_i, r_j \in R,\ t \in T\})$ can be directly accessed when necessary.

The road-set correlation, which measures the correlation between a road (e.g., r_i) and a set of roads (e.g., R^c), is defined as the maximum road-road correlation between them:

$$corr^t(r_i, R^c) = \max\{corr^t(r_i, r_j) \mid r_j \in R^c\}. \quad (11)$$

The set-set correlation is proposed to indicate relationship between the queried roads R^q and the crowdsourced roads R^c. Specifically, it is defined as the summation of road-set correlation for roads in R^q given R^c:

$$corr^t(R^q, R^c) = \sum\nolimits_{r_i \in R^q} corr^t(r_i, R^c). \quad (12)$$

Periodicity-weighted Correlation. As discussed in last section, the intensity of periodicity for each road is different and reflected in the parameter of standard variance σ_i^t. For a road with weak periodicity (i.e., the value of σ_i^t is large), the expected speed μ_i^t cannot approximate the ground-truth of realtime traffic speed well. In this case, we need to rely more on the crowdsourced data to make a quality inference for the

Algorithm 2: Ratio Greedy

```
input : $R^q$, $R^w$, $\Gamma_R$, $\Omega$, $K$, $\theta$
output: $R^c$
begin
    Initialize: $R^c \leftarrow \emptyset$, $budget \leftarrow K$,
      feasible_set $\leftarrow \{r_i | c_i \leq K, r_i \in R^w\}$;
    while $feasible\_set \neq \emptyset$ do
        $r^* = \text{argmax}\{ratio(r, R^c)|$ feasible_set $\}$;
        $R^c \leftarrow R^c + r^*$, budget $\leftarrow$ budget $- c_{r^*}$,
          feasible_set $\leftarrow \{r_i | c_i \leq$ budget,
          $r_i \in R^w / R^c,\ corr^t(r_i, R^c) \leq \theta\}$;
Return $R^c$;
```

Algorithm 3: Objective Greedy

```
input : $R^q$, $R^w$, $\Gamma_R$, $\Omega$, $K$, $\theta$
output: $R^c$
begin
    Initialize: $R^c \leftarrow \emptyset$, $budget \leftarrow K$,
      feasible_set $\leftarrow \{r_i | c_i \leq K, r_i \in R^w\}$;
    while $feasible\_set \neq \emptyset$ do
        $r^* = \text{argmax}\{\text{ocs}(R^c + r) - \text{ocs}(R^c)|$ feasible_set $\}$;
        $R^c \leftarrow R^c + r^*$, budget $\leftarrow$ budget $- c_{r^*}$,
          feasible_set $\leftarrow \{r_i | c_i \leq$ budget,
          $r_i \in R^w / R^c,\ corr^t(r_i, R^c) \leq \theta\}$;
Return $R^c$;
```

traffic speed. Considering that the crowdsourcing resource is limited by the given budget, higher priority should be placed to those roads with weak periodicity during crowdsourced roads selection. As such, we propose to incorporate the intensity of periodicity and the strength of correlation simultaneously during OCS, by introducing the periodicity-weighted correlation, which is defined as:

$$\widehat{corr}(R^q, R^c) = \sum\nolimits_{r_i \in R^q} \sigma_i^t * corr^t(r_i, R^c), \qquad (13)$$

where σ_i^t is the intensity of periodicity for r_i and $corr^t(r_i, R^c)$ refers to the correlation with crowdsourced roads. In this work, to maximize $\widehat{corr}(R^q, R^c)$ becomes the objective of OCS. Apparently, the maximization of $\widehat{corr}(R^q, R^c)$ will not only maximize the correlation between R^q and R^c, but emphasize more on the queried roads with weaker periodicity as well.

Feasibility. During the crowdsourced roads selection, there are two feasibility constraints that must be satisfied. Firstly, the crowdsourced roads have to be selected from the roads where workers are currently distributed (denoted as R^w), i.e., $R^c \subseteq R^w$; Secondly, the scale of selected crowdsourced roads must be restricted to the limited budget. In real world applications, one single answer may not reflect the ground-truth of realtime traffic speed on the corresponding road. To obtain a more accurate result, multiple answers are required to be collected and integrated for each crowdsourced road. In this work, for each candidate road, the minimum number of its required answers is referred as cost. Suppose that each answer will be rewarded with one unit of payment. Given the maximum payment K, the budget constraint can be presented as : $\sum_{r_i \in R^c} c_i \leq K$, where c_i is the cost of road r_i and K denotes the total budget.

It is notable that the road cost value can be distinct with each other. For example, vehicles on the highway are normally traveling with a constant speed, while those on the secondary road may experience more significant speed fluctuations. As a result, the crowd's answers for the highway tend to be more stable and accurate, which leads to a smaller road cost. Moreover, many existing approaches (e.g. [28], [29]) can be adopted to determine the cost of each road, which estimate the exact value from the historical answers of crowd.

Redundancy. The redundancy is introduced to evaluate the internal correlation within R^c. Specifically, given two roads r_i and r_j ($r_i,\ r_j \in R^c$), the redundancy between them is measured with road-road correlation $corr^t(r_i, r_j)$. Clearly, it is unnecessary to select two highly-correlated roads from R^c at the same time, since one road's realtime traffic speed can be well inferred with the knowledge of the other's. To prevent the kind of ineffective selection, the following constraint is enforced for each pair of crowdsourced roads:

$$corr^t(r_i, r_j) \leq \theta,\ \forall\ r_i, r_j \in R^c, \qquad (14)$$

where θ (0< θ <1) is the threshold of redundancy. A smaller value of θ can bring a stronger restriction on the redundancy, while this may narrow down the feasible candidates of R^c and lead to some adverse impacts on the objective optimization. As such, an optimal setting of θ is necessary for the crowdsourced road selection, which can be appropriately tuned through the exploration of historical data [30].

Optimal Crowdsourced Roads Selection. As the desirable factors (periodicity and correlation) have been encoded into the periodicity-weighted correlation, the objective of optimal crowd selection is to maximize $\widehat{corr}(R^q, R^c)$. Together with the feasibility and redundancy constraints, the formulation of OCS is given as follows:

$$\begin{aligned} \max \quad & \widehat{corr}(R^q, R^c) \\ s.t. \quad & R^c \subseteq R^w,\ \sum\nolimits_{r_i \in R^c} c_i \leq K, \\ & corr^t(r_i, r_j) \leq \theta,\ \forall\ r_i, r_j \in R^c. \end{aligned} \qquad (15)$$

Remark 2. (Trivial Cases of OCS) Notice that when $\theta = 1$ and $c_r = 1, \forall r \in R$, there will be two cases where the optimal solution of OCS is trivial to get. The first case happens when $|R^w| < K$, which means the budget is over-adequate so that all candidate roads R^w can be selected. In this case, the optimal solution turns out to be: $R^{c*} = R^w$. The second trivial case appears when $|R^q| < K$. In this case, it's easy to verify that the optimal solution is produced by selecting the highest correlated roads for each of the queried ones, which is presented as the following expression:

$$R^{c*} = \bigcup \{\text{argmax}_{R^w}\{corr^t(r_i, r_j) \mid r_i \in R^q\}\}.$$

While solving OCS, we will exclude the trivial cases from discussion. In this situation, the hardness of OCS is presented by the following theorem.

Theorem 1: The OCS problem is NP-hard.

Proof: The NP-hardness of OCS is demonstrated by its reduction to the Maximum k-Coverage (MKC) problem. The

Algorithm 4: Hybrid Greedy

```
input : $R^q$, $R^w$, $\Gamma_R$, $\Omega$, $K$, $\theta$
output: $R^c$
begin
    $R^c_{ratio} \leftarrow$ answer of Ratio-Greedy;
    $R^c_{obj} \leftarrow$ answer of Ratio-Objective;
    $R^c = \text{argmax}\{\text{ocs}(R^c_x) | R^c_x \in \{R^c_{ratio}, R^c_{obj}\}\}$;
Return $R^c$;
```

MKC is NP-hard, whose formulation is presented as follows. Suppose we a set of elements: $X = \{x_1, ..., x_n\}$, and the set coverage relationship: $S = \{s_1, ..., s_m\}$, which is defined over the domain of X (i.e. $x_i \in s_j$ iff. x_i is covered by s_j). The MKC problem is to find the optimal subset S' ($S' \subset S$), which covers the maximum number of elements subject to the condition that $|S'| \le k$. Now, we specify the following instance of OCS: 1) let $\theta = 1$, $k = K$ and $c_r = 1, \forall r \in R$; 2) let $corr^t(x_i, s_j) = 1$ iff. s_j covers x_i, otherwise $corr^t(x_i, s_j) = 0$; 3) let $\sigma_i^t = 1$, $\forall x_i{\in}X$ and $t{\in}T$. Clearly, the optimal solution of the above OCS problem also leads to the optimization of the corresponding MKC problem. Hence, OCS can be regarded as a generalization of MKC, which justifies the claim of Theorem 1. ■

B. Solution of OCS

In this part, we firstly propose a greedy-based approach, namely Ratio-Greedy, which solves OCS in linear time, but performs poorly for the worst case. Then, Ratio-Greedy is adapted to a hybrid solution, a.k.a. Hybrid-Greedy, which not only preserves the property of linear time complexity, but achieves a constant approximation ratio ($(1 - \frac{1}{e})/2$) as well.

Ratio-Greedy. The Ratio-Greedy solves OCS in an iterative manner,and initially, R^c is set to be an empty set: $R^c \leftarrow \emptyset$. To select the crowdsourced roads which produce large objective value of OCS, both the objective increment induced by a candidate road and the corresponding cost have to be considered. To capture both aspects, the objective-cost ratio of a non-included road is defined as:

$$ratio(r_i, R^c) = (\text{ocs}(R^c + r_i) - \text{ocs}(R^c))/c_i,\ r_i \in R^w/R^c,$$

where R^c is the currently selected crowdsourced roads and $\text{ocs}(R^c)$ is the objective value of R^c. With such a definition, Ratio-Greedy selects the next crowdsourced road from the non-included ones (i.e., R^w/R^c), which maximizes the objective-cost ratio and satisfies the feasibility requirement simultaneously in each iteration:

$$r^* = \text{argmax}\{ratio(r, R^c) | (R^c + r) \text{ is feasible}\}.$$

The selected road r^* will be added to the current solution: $R^c \leftarrow R^c + r^*$, and such an operation will be repetitively conducted until no more feasible candidates can be included. The whole process of Ratio-Greedy is summarized as Alg. 2.

It is clear that Alg. 2 converges within no more than K iterations, and each iteration takes O(|feasible-set|) (which is less than O($|R^w|$)) time to find the best candidate to include. Hence, the overall time complexity is O($K|R^w|$). The major space cost of Alg. 2 is induced by feasible-set, which is bounded by O($|R^w|$). Both aspects are linear in terms of $|R^w|$.

Although Ratio-Greedy solves OCS with economic running time, its solution can be arbitrary bad for the worst case. Such a property is demonstrated by the following example.

Example 1: (Worst Case Analysis of Ratio-Greedy) Suppose there are two candidate roads: r_1 and r_2 (i.e., R^w= $\{r_1, r_2\}$), whose cost equal to 1 and K (suppose K>1), respectively. Besides, there is one queried road r_3 (i.e., R^Q=$\{r_3\}$), and periodicity-weighted correlation are set to be: $\widehat{corr}(\{r_3\}, \{r_1\})$=1, $\widehat{corr}(\{r_3\}, \{r_2\})$=$K$-1. Assume that θ=1 (which means the redundancy requirement will always be satisfied), and the total budget is set to be K. For such a problem, Ratio-Greedy will produces the solution: R^c=$\{r_1\}$, which leads to an objective value of 1. However, the optimal solution should be R^c=$\{r_2\}$, which gives a objective value of K (since K>1). As such, the approximation ratio of Ratio-Greedy is $1/K$, which will be arbitrary small when K is large.

Hybrid-Greedy. To achieve constant approximation ratio, the Hybrid-Greedy algorithm is developed, which incorporates two different greedy solutions: the Ratio-Greedy and Objective-Greedy to produce its answer. The process of Ratio-Greedy is the same as our previous discussion, while the Objective-Greedy is presented as follows.

Similar with Ratio-Greedy, Objective-Greedy also employs an iterative approach to generate its solution. However, for each iteration, Objective-Greedy picks the non-included crowdsourced road, which maximizes the objective increment, into the current selection of R^c. Specifically,

$$r^* = \text{argmax}\{\text{ocs}(R^c + r) - \text{ocs}(r) | (R^c + r) \text{ is feasible}\}.$$

The selected r^* is included to R^c, and the selection is repetitively conducted until no feasible roads can be added. The process of Ratio-Greedy is summarized as Alg. 3.

Finally, the Hybrid-Greedy conducts both Ratio-Greedy and Objective-Greedy separately, and the solution with higher objective value is selected to be its answer.

It is obvious that both Ratio-Objective and Objective-Greedy take the same time complexity: O($K|R^w|$), thus, the overall time complexity of Hybrid-Greedy is O($K|R^w|$) as well. Besides, it's easy to verify that Objective-Greedy takes the same space complexity as Ratio-Greedy, hence, the space complexity of Hybrid-Greedy is O(R^w).

The solution of Hybrid-Greedy achieves a constant approximation ratio w.r.t. the optimal value. Such a property is presented as the following theorem.

Theorem 2: Hybrid-Greedy achieves an approximation ratio of $(1 - \frac{1}{e})/2$.

Proof: The performance of Greedy-Ratio is analyzed in the first place. Suppose R^c is selected in l iterations, and the temporary result of the kth iteration is denoted as: R^c_k (k=1,...,l). Because R^c is generated through Ratio-Greedy selection, the following inequality can be easily deduced:

$$\begin{aligned} & ocs(R^{c*}) - ocs(R^c_{k-1}) \le ocs(R^{c*}/R^c_{k-1}) \\ \le\ & K * (ocs(R^c_k) - ocs(R^c_{k-1}))/c_k \end{aligned}$$

Algorithm 5: Graph-based Speed Propagation

```
input : $\hat{V}_{R^c}$, $G^t$, $\epsilon$
output: $V_R^t$
begin
    Initialize: $V_{R^c}^t \leftarrow \hat{V}_{R^s}$;
    Sorting: $\{V_1, ..., V_L\} \leftarrow$ BFT($V_{R \setminus R^c}^t, V_{R^c}^t$);
    while Not converge do
        for $l$ = 1, ... , $L$ do
            for $v_i^t \in V_l$ do
                $v_i^t \leftarrow v_i^{t*}$;
    Return $V_R^t$;
```

where R^{c*} is the optimal solution and c_k is the cost of the road included in the kth iteration. Based on such an inequality, the following conclusion can be verified through induction:

$$ocs(R_l^c) \geq [1 - \prod\nolimits_{k=1}^{l}(1 - c_k/K)] * ocs(R^{c*}).$$

Now assume it's possible to take one more road r_{l+1}^c into R^c through Ratio-Greedy (which produces $R_{l+1}^c \leftarrow R_l^c + r_{l+1}^c$), the following inequality will be induced:

$$\begin{aligned} ocs(R_{l+1}^c) &\geq [1 - \prod\nolimits_{k=1}^{l+1}(1 - c_k/K)] * ocs(R^{c*}) \\ &\geq [1 - \prod\nolimits_{k=1}^{l+1}(1 - c_k/c(R_{l+1}^c))] * ocs(R^{c*}), \end{aligned}$$

where $c(R_{l+1}^c) = \sum_k c_k$. Clearly, the minimum value for last expression of the above inequality is achieved when c_k=$\lfloor K/l \rfloor$, k=$1, ..., l$, thus leading to the following inequality:

$$(1 - 1/e)ocs(R^{c*}) \leq ocs(R_{l+1}^c) \leq ocs(R^c) + ocs(\{r_{l+1}^c\}).$$

Now we turn our focus back to Hybrid-Greedy. Because Objective-Greedy generates its answer $R^{c'}$ by selecting the candidate roads w.r.t. the descending order of the objective growth, and Hybrid-Greedy picks the winner of R_c and R_c' as its final answer, the following relationship can be deduced:

$$\begin{aligned} &max\{ocs(R^c), ocs(R^{c'})\} \geq (ocs(R^c) + ocs(R^{c'})/2 \\ &\geq ocs(R^c) + ocs(\{r_{l+1}^c\})/2 \geq ocs(R^{c*}) * (1 - 1/e)/2, \end{aligned}$$

which justifies the claim of Theorem 2. ■

VI. Graph-based Speed Propagation

With the realtime traffic speed probed for the crowdsourced roads, the speed propagation is conducted on top of RTF, which infers the realtime traffic speed of the whole traffic network. As such, the RTSE is produced for the queried roads. The speed propagation consists of two steps: the initialization and the iterative update, which are presented as follows.

Initialization. Given the crowdsourced realtime traffic speed (denoted as $\hat{V}_{R^c}$), the variables of $V_{R^c}^t$ are updated firstly according to the crowdsourced data, and at the same time, the variables of $V_{R \setminus R^c}^t$ are initialized with their mean values:

$$v_i^t = \hat{v}_i,\ \forall r_i \in R^c;\ v_j^t = \mu_j^t, \forall r_j \in R \setminus R^c.$$

The crowdsourced data reveals the current status of the traffic network, based on which the realtime traffic speed of all the rest of roads (i.e. $R \setminus R^c$) is adjusted accordingly. To find the most likely speed for these roads, the following likelihood maximization is derived w.r.t. the RTF model (G^t):

$$\max \mathcal{L}_{G^t}(V_{R \setminus R^c}^t \mid V_{R^c}^t = \hat{V}_{R^c}). \tag{16}$$

	|R^w|	**|R^q|**	**road cost**	**K**	**θ**
Semi-syn	607	33, 51	1~5, 1~10	30~150	0.92, 1
gMission	30	50	1~10	10~50	0.92

TABLE II: Datasets' Statics

Iterative Update. According to Eq. (4), it's easy to verify that Eq. (16) is a non-convex function w.r.t. the variables of V_R^t. Therefore, an EM-based approach: iterative update, is employed to achieve the maximization of Eq. (16). To be specific, the maximization is carried out iteratively, and for each iteration one single variable of $V_{R^c}^t$ (e.g., v_i^t) is selected, which maximizes $\mathcal{L}_{G^t}$ with all other variables being fixed:

$$\max \mathcal{L}_{G^t}(v_i^t | V_R^t / v_i^t) \tag{17}$$

Let the partial derivative of $\mathcal{L}_{G^t}$ equal to 0: $\partial \mathcal{L}_{G^t} / \partial v_i^t = 0$, the value of v_i^t can be optimal updated as:

$$v_i^{t*} \leftarrow (\frac{\mu_i^t}{{\sigma_i^t}^2} + \sum\nolimits_{V_{n(r_i)}^t} \frac{(v_j^t + \mu_{ij}^t)}{{\sigma_{ij}^t}^2}) / (\frac{1}{{\sigma_i^t}^2} + \sum\nolimits_{V_{n(r_i)}^t} \frac{1}{{\sigma_{ij}^t}^2}). \tag{18}$$

Based on Eq. (18), the update of v_i^t ($r_i \in R \setminus R^c$) is only triggered by the value changes of its adjacent variables. In other words, if v_i^t has no adjacent variables change their values, it will remain its initialized value μ_i^t (which can be easily verified with Eq. (18)). Therefore, the iterative update should be launched from the adjacent variables of R_c (denoted as $n(R^c)$), and then progressively propagated to the rest. To incorporate such a property, the update sequence of the all the variables is scheduled w.r.t. the hop-count towards $V_{R^c}^t$. Specifically, the variables of $V_{R \setminus R^c}^t$ are sorted with the ascending order of their minimum hop-count towards $V_{R^c}^t$, and the update process is carried out according to the sorted order. The iterative update has to be repetitively conducted, and the convergence is reached iff. the value changes of all the variables are lower than a predefined small number "ϵ".

The workflow of speed propagation is presented as Alg 5. Before the iterative update is launched, the variables of $V_{R^c}^t$ are partitioned into $\{V_1, ..., V_L\}$ through BFS (breadth-first-search) w.r.t. $V_{R^c}^t$. Obviously, the variables within the same partitioned group have the same hop-count towards $V_{R^c}^t$, therefore, they are put to the same update loop.

Time Efficiency of GSP. Given a fixed threshold ϵ, it can be proved that Alg. 5 converges with constant number (denoted as Λ) of iterations. As such, the overall time complexity for Alg. 5 is $O(\Lambda |R \setminus R^s|)$. Besides, Alg. 5 can be easily parallelized. According to Eq. (18), the calculation of v_i^{t*} is only determined by three types of parameters: the parameters of v_i^t: μ_i^t, σ_i^t, the values and parameters of v_i^t's adjacent variables: v_j^t, μ_{ij}^t, σ_{ij}^t, and the weights of v_i^t's adjacent edges: ρ_{ij}^t. Based on the conclusion of [31], the variables of v_i^t and v_j^t can be updated in a concurrent manner iff. 1) both v_i^t and v_j^t are within the same partitioned group V_l, 2) v_i^t and v_j^t are not adjacent to each other. Such a parallelization will lead to further reduction of running time, which contributes to the realtime computation of RTSE.

VII. Experimental Study

A. *Experiment Setup*

The experiments are conducted for the traffic network of Hong Kong, whose realtime traffic speed is published by the Public Sector Information Portal of Hong Kong SAR[1]. The realtime traffic speed is published every 5 minutes for the road network in Hong Kong, where a total of 607 roads are monitored. The data is continuously crawled for 3 consecutive months, and 5244480 pieces of speed records are collected in total. Such data is used for constructing the graphical model RTF and provides the ground the truth for the evaluations.

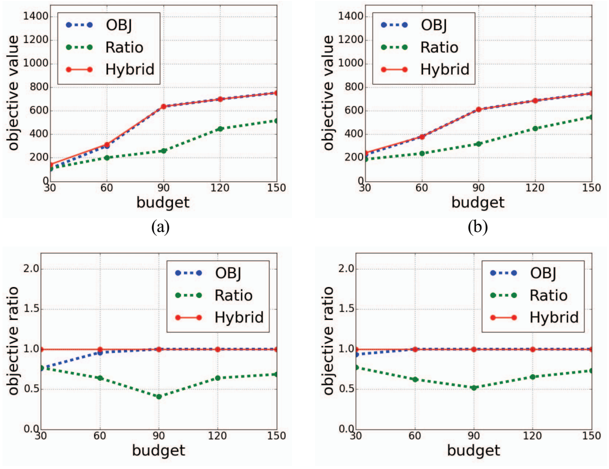
Two datasets are tested for the evaluations. One is the **semi-synthesized** dataset, where queried roads R^q are randomly selected from R with uniform distribution, and crowd's answers are generated with the ground-truth speeds. In addition, workers' are assumed to cover all the tested roads, i.e., $R^w = R$. The other one is the **gMission dataset**, which is collected from gMission[2] gMission is research-based general spatial crowdsourcing system, which is able to provide the workers' localization information. To perform realtime traffic speed detection, a worker simply needs to activate the localization option and the traveling speed can be calculated within a short period of time. To ensure correlation between the queried and crowdsourced roads, a mutually connected subcomponent of R is selected as R^q, and workers are asked to travel along such roads; as such, $R^w \subset R^q$. Finally, the roads' costs are generated synthetically for both datasets with uniform distributions. Although road-length or travel cost would be more meaningful choices in practice, such kinds of auxiliary information are not included in our experimental data. Thus, we adopt the randomized costs which will not affect the correctness of evaluation.

Statistics for both semi-synthesized and gMission datasets are shown in Table II.

B. *Evaluation of OCS*

1) Evaluated Methods and Metrics: We jointly evaluate three candidate solutions for OCS: Hybrid-Greedy (Hybrid), Ratio-Greedy (Ratio) and Objective-Greedy (OBJ). The comparison is carried out w.r.t. two aspects: the objective value of OCS (VO), and the overall running time (ORT).

2) Experiment Results and Analysis: Figure 2 (a) and (b) demonstrate the relationship between VO and the varying budget. To highlight the different performance of each algorithm, the VO's ratios of Ratio-Greedy/Hybrid-Greedy and Objective-Greedy/Hybrid-Greedy are shown in Figure 2 (c) and (d). In (a) and (c), the roads' costs are randomly generated within C_1, while in (b) and (d), the roads' costs are randomized within C_2. There is no obvious difference between different setting of θ, so the results are only reported for $\theta = 0.92$ due to the space limitation.



(c)

(d)

Fig. 2: Evaluation of OCS

Effect of budget. In both Figure 2 (a) and (b), VO monotonously grows with the incrementation of budget, and the monotonicity can easily be verified from the process of Alg. 2, 3 and 4. Besides, in every testing case, Hybrid-Greedy produces the highest VO, as Hybrid-Greedy takes the winner of Ratio-Greedy and Objective-Greedy as its output. Moreover, the VO of both Ratio-Greedy and Objective-Greedy are lower than that of Hybrid-Greedy when budget is small, but Ratio-Greedy produces the same VO as Hybrid-Greedy when budget is large enough. Such a phenomenon can be explained as follows. Suppose R^c_{rat} and R^c_{obj} are the roads selected by Ratio-Greedy and Objective-Greedy, respectively. If $ocs(R^c_{rat}) < ocs(R^c_{obj})$, then there must exist two elements r_x ($r_x \in R^c_{rat}$) and r_y ($r_y \in R^c_{obj}$), which satisfy: $ocs(R^c_{rat}/r_x + r_y) < ocs(R^c_{rat})$, and it can be further inferred that $cost(R^c_{rat}/r_x) + c_y > K$. Whereas, if budget K is increased to $cost(R^c_{rat}/r_x) + c_y$, r_x can be replaced by r_y: $R^c_{rat} \leftarrow R^c_{rat}/r_x + r_y$. When K is large enough, it can be guaranteed that no elements $r_x \in R^c_{rat}$ and $r_y \in R^c_{obj}$ will satisfy $ocs(R^c_{rat}/r_x + r_y) < ocs(R^c_{rat})$.

Effect of roads' costs. Comparatively speaking, the difference between Hybrid-Greedy and Ratio-Greedy is larger when the roads' costs are generated from C_1. Such a phenomenon can be further explained based on our discussion in the last paragraph. If $ocs(R^c_{rat}) < ocs(R^c_{obj})$, the elements r_x and r_y must satisfy the conditions that: 1) $r_x \in R^c_{rat}$ and $r_y \in R^c_{obj}$, 2) $ocs(R^c_{rat}/r_x + r_y) < ocs(R^c_{rat})$, 3) $cost(R^c_{rat}/r_x) + c_y > K$. Given the fact that $cost(R^c_{rat}/r_x) + c_x \leq K$, if costs of different roads are similar, the probability for occurrence of such elements (r_x and r_y) will be small. On the contrary, when costs of roads are randomized in a larger range C_1, costs between different roads will be less similar, which makes the phenomenon more obvious.

Scalability. The running time of the evaluated algorithms is demonstrated in Figure 4 (a), where the roads' costs are generated within C_1. (Similar observations can be found for the setting of C_2.) According to the results, the running time for all the algorithms grows linearly with the incrementation of budget. When budget reaches the maximum number, the most

[1] https://data.gov.hk/en/

[2] http://www.gmissionhkust.com/.

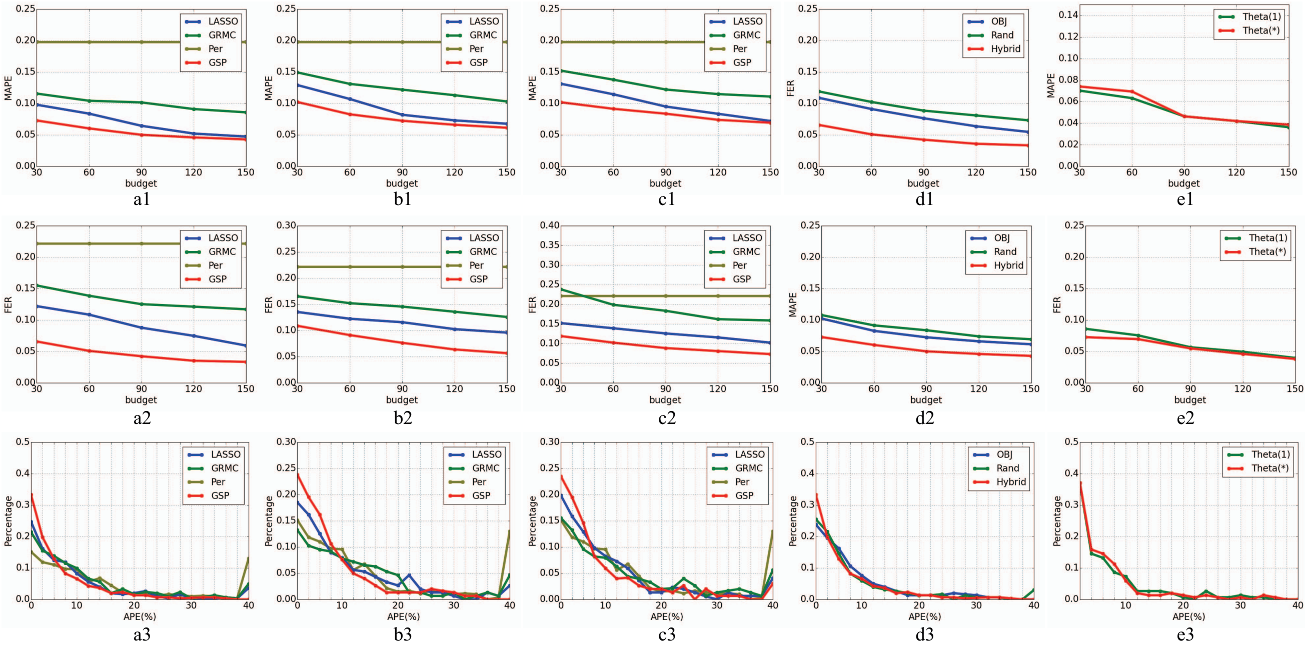


Fig. 3: Evaluation of GSP

time-consuming algorithm, Hybrid-Greedy, can still return the answer within one second, which verifies its scalability and efficiency in terms of running time.

C. Evaluation of GSP

1) Evaluated Methods and Metrics: Three baseline methods are selected to compare with the Graph-based Speed Propagation (GSP), which is proposed in this work. In particular, LASSO Regression (LASSO, [32]) and Graph Regularized Matrix Completion (GRMC, [33], [16]) are employed[3], which purely depend on traffic speeds' correlation to make their estimation. Both methods are popular representatives of realtime traffic speed estimation, thanks to their effectiveness of dealing with over-fitting and data sparsity. Besides, Per is adopted, which purely relies on the periodicity and provides the periodic traffic speeds as its estimation.

The comparison is conducted from four aspects: the mean absolute percentage error (MAPE), the false estimation rate (FER), the distribution of absolute percentage error (DAPE), and overall running time. Specifically, the absolute percentage error (APE) is defined as ratio between the absolute estimation error and the ground truth: $|\hat{y}-y|/y$, where $\hat{y}$ is the estimated value and y is the ground truth. The MAPE shows the average absolute percentage error of all the testing cases, and the DAPE demonstrates the testing cases' distribution over APE. In addition, a testing case is determined to be a false estimation iff. its APE exceeds a predefined threshold φ (φ=0.2 in our experiments), and FER calculates the rate of false estimation for all the testing cases.

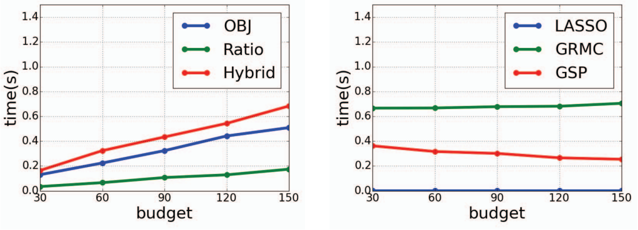
2) Experiment Results and Analysis: The experiment results are demonstrated in Figure 4, where the 1st, 2nd and 3r rows show the results of MAPE, FER, and DAPE, respectively (due to the limitation of space, DAPE is only demonstrated for cases whose budgets equal to 30). Columns a, b and c show the results whose crowdsourced roads are selected by Hybrid-Greedy, Objective-Greedy and Randomization; column d demonstrates comparisons of different crowdsourced roads selections, and column e indicates the effect of different θ.

Effect of different estimation methods. From the results of first three columns, the following phenomenons can be observed: in most of the testing cases, the best performances are generated by GSP; besides, the advantage of GSP is most clear when budget equals to the minimum value 30, as both of its MAPE and FER are significantly lower than the others, and most of its estimation errors (APE) are closely distributed to zero (which is captured by DAPE). Although the performance of LASSO is comparable to GSP in terms of MAPE when budget is large enough, there still exists clear gap in the aspect of FER. The above phenomenons can be summarized into the following points: firstly, GSP is able to produce high quality speed estimation, which is comparatively more effective with small budget. Secondly, GSP is more effective in preventing the false estimation from happening. Such findings are in accord with our previous discussion about GSP, and can be clearly justified by the following explanations. First, GSP jointly takes advantage of the periodicity and correlation of traffic speeds, while others simply uses one of such properties. When the realtime probed data is sparse, GSP is more capable of producing accurate estimation because of its sufficient exploitation of historical data. Second, the road network is sparsely connected and one road may be only closely correlated to very few number of other roads. In this situation, it is almost impossible to eliminate the occurrence over-fitting ([34], [35]), which inevitably leads to frequent

[3] LASSO and GRMC's parameters, L1-regularization and latent-dimension, are tuned within 0∼0.5 and 5∼20, respectively; and set to be 0.1 and 10 for the best performances.

	30	60	90	120	150
OBJ	24 / 38	39 / 58	65 / 79	73 / 99	91 / 106
Rand	20 / 32	33 / 51	48 / 72	62 / 92	72 / 108
Hybrid	33 / 43	49 / 68	76 / 94	93 / 109	115 / 126

TABLE III: 1-hop and 2-hop Coverages of the Queried Roads.



(a) (b)

Fig. 4: Overall Running Time

false estimation even the budget is large.

Effect of budget size. In columns a, b and c of Figure 3, it can also be observed that stepwise improvement of GSP's estimation quality is large when budget is small, and it becomes increasingly smaller with the growth of budget. Such a tendency becomes more obvious when the crowdsourced roads are selected by Hybrid-Greedy. A possible explanation about this phenomenon is made as follows. As discussed previously, some roads may have weak periodicity and it is hard to estimate their realtime traffic speed without the help of crowdsourced data, while others may have strong periodicity whose traffic speed can be accurately estimated with little help of crowdsourced data. (As such, it is reasonable to assume that most of the estimation errors are resulted from the weak-periodic roads.) Since the number of weak-periodic roads is limited, and in Hybrid-Greedy, the crowdsourced roads are selected for them with higher priority, the estimation errors caused by such roads will be quickly reduced in the early stage. That is why the reduction rates of MAPE and FER are comparatively larger for the smaller budgets.

Effect of crowdsourced roads selection. The performance of GSP is further studied with comparisons between different crowdsourced roads selections, whose results are shown in Figure 3 d1, d2 and d3. In addition, 1-hop and 2-hop Coverages of the queried roads, i.e., roads of R^q covered by 1-hop and 2-hop neighborhoods of R^c, are explicitly shown in Table III. It is observed that the adoption of Hybrid-Greedy clearly improves the estimation quality; meanwhile, more queried roads can be covered by neighbors of the crowdsourced roads. Such a finding is consistent with our previous evaluation of OCS in Figure 2, which reflects that Hybrid-Greedy effectively identifies the roads contributing the most to realtime traffic speed estimation; whereby, the estimation quality is significantly improved especially when budget is limited.

Effect of redundancy threshold. The GSP performance with different settings of redundancy threshold θ is demonstrated in Figure 3 (e1), (e2) and (e3), where Theta(1) stands for the setting of $\theta = 1$ while Theta($*$) represents the fine-tuned setting $\theta = 0.92$. From the results, it can be observed that the fine-tuned θ is able to improve the estimation quality when the budget is small. However, there is no clear difference between both settings when budget is large enough. The following explanation is given for such a phenomenon. When budget is small, an appropriate setting of θ will resist the selection of two highly correlated roads whose probed data is redundant. In other words, the crowdsourced roads will be able to provide more diversified information about the current traffic status, which help to make more accurate estimation. However, when budget is large, the crowdsourced roads can be adequately selected for all the queried roads, and the diversity of information is less crucial, thereby resulting in the similar performances of both settings.

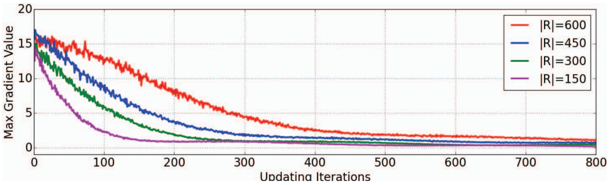


Fig. 5: Convergence of RTF Training.

Scalability. The running time of LASSO, GRMC and GSP is demonstrated in Figure 4 (b) (Per is omitted as its result can be directed accessed from RTF). According to the results, LASSO utilizes the smallest running time, as it simply requires one step of matrix multiplication to get the result. Both GRMC and GSP work iteratively, thus their overall running time is clearly higher. However, the running time of GSP is almost independent of the budget size (which is in accord with the process of Alg. 5), and the estimation can always be made within half a second. As such, the time efficiency and scalability of GSP are justified. In addition, we also demonstrate the scalability RTF's offline training. Specifically, subcomponents of 150 to 600 roads are selected from the whole road network, with training convergences measured in terms of $\{\mu\}_R$'s maximum gradient[4]. According to tested result in Figure 5, it takes more iterations to train RTF for a larger network. However, the growth of convergence is roughly linear to the network size, which means training cost for large networks will be tolerable in practice.

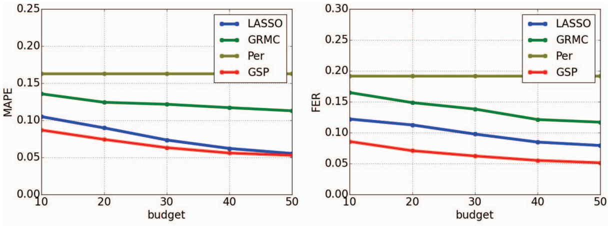
gMission Dataset. Results on gMission dataset is demonstrated in Figure 6, where GSP, LASSO, GRMC and Per are compared in terms of MAPE and FER, with crowdsourced roads selected by Hybrid-Greedy. Despite comparative smaller testing scales (compared with the semi-synthesized data), the results show similar patterns to Figure 3 a1 and a2, where the semi-synthesized data is tested, thus further verifying our conclusions for the semi-synthesized data.

D. Experiment Summary

The findings of the experiments can be summarized with the following points.

- The adoption of Hybrid-Greedy guarantees that the near-optimal solution of OCS can always be obtained, and it is especially crucial for the cases where the budget is small and the roads' costs are largely varied. Although the running

[4] vanilla gradient descent is employed, where λ is fixed to be 0.1; and each iteration takes less than 100 ms on the testing machine.



(a) (b)

Fig. 6: Evaluation of gMission data.

time of Hybrid-Grid is comparatively higher, the efficiency and scalability are still feasible for real applications.

• The estimation quality is clearly improved by GSP, especially for the situations of small budgets. Besides, the effective solution of OCS is equally important to estimation quality. Moreover, with proper tuning of the redundancy threshold, the estimation quality can be further improved.

VIII. Conclusion

In this paper, we propose a novel framework CrowdRTSE to estimate the realtime traffic speed within sparse crowdsourced data. To produce high-quality estimation, CrowdRTSE works with a hybrid offline-online manner so that the historical data and realtime crowdsourced data are jointly exploited. In the offline stage, a graphical model RTF is constructed based on the historical data, which effectively captures both periodicity and correlation of the traffic speeds. In the online stage, the crowdsourcing resources are judiciously allocated with OCS and fine-grained realtime traffic speeds are produced with GSP. Extensive experiments are conducted on both real and synthetic datasets, which verify the effectiveness and efficiency of our proposed methods.

IX. Acknowledgment

The work is partially supported by the Hong Kong RGC GRF Project 16207617, National Grand Fundamental Research 973 Program of China under Grant 2014CB340303, the National Science Foundation of China (NSFC) under Grant No. 61729201, Science and Technology Planning Project of Guangdong Province, China, No. 2015B010110006, Webank Collaboration Research Project, and Microsoft Research Asia Collaborative Research Grant.

References

[1] L. A. Klein, M. K. Mills, and D. R. Gibson, "Traffic detector handbook: -volume ii," Tech. Rep., 2006.

[2] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *ICMDM'11*.

[3] W.-C. Hong, "Application of seasonal svr with chaotic immune algorithm in traffic flow forecasting," *Neural Computing and Applications*, 2012.

[4] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering*, 2003.

[5] J. Zheng and L. M.-S. Ni, "Time-dependent trajectory regression on road networks via multi-task learning," in *AAAI'13*.

[6] S. Clark, "Traffic prediction using multivariate nonparametric regression," *Journal of transportation engineering*, 2003.

[7] H. Tan, G. Feng, J. Feng, W. Wang, Y.-J. Zhang, and F. Li, "A tensor-based method for missing traffic data completion," *Transportation Research Part C: Emerging Technologies*, 2013.

[8] A. Artikis, M. Weidlich, F. Schnitzler, I. Boutsis, T. Liebig, N. Piatkowski, C. Bockermann, K. Morik, V. Kalogeraki, J. Marecek *et al.*, "Heterogeneous stream processing and crowdsourcing for urban traffic management." in *EDBT'14*.

[9] H. Huiqi, L. Guoliang, B. Zhifeng, C. Yan, and F. Jianhua, "Crowdsourcing-based real-time urban traffic speed estimation: From trends to speeds." in *ICDE'16*.

[10] G. Gopi, J. Dauwels, M. T. Asif, S. Ashwin, N. Mitrovic, U. Rasheed, and P. Jaillet, "Bayesian support vector regression for traffic speed prediction with error bars," in *ITSC'13*.

[11] Z. Shan, D. Zhao, and Y. Xia, "Urban road traffic speed estimation for missing probe vehicle data based on multiple linear regression model," in *ITSC'13*.

[12] N. Polson and V. Sokolov, "Deep learning predictors for traffic flows," *arXiv 2016*.

[13] Y. Tian and L. Pan, "Predicting short-term traffic flow by long short-term memory recurrent neural network," in *IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 2015.

[14] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, 2015.

[15] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *KDD'14*.

[16] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, "Latent space model for road networks to predict time-varying traffic," *arXiv 2016*.

[17] K. Q. Weinberger, F. Sha, Q. Zhu, and L. K. Saul, "Graph laplacian regularization for large-scale semidefinite programming," in *NIPS'06*.

[18] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "Crowder: Crowdsourcing entity resolution," in *PVLDB 2012*.

[19] Y. Tong, C. C. Cao, C. J. Zhang, Y. Li, and L. Chen, "Crowdcleaner: Data cleaning for multi-version data on the web via crowdsourcing," in *ICDE'14*.

[20] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang, "gmission: A general spatial crowdsourcing platform," *PVLDB 2014*.

[21] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in real-time spatial data: experiments and analysis," *PVLDB 2016*.

[22] C. E. Rasmussen and C. K. Williams, *Gaussian processes for machine learning*. MIT press Cambridge, 2006, vol. 1.

[23] B. Yang, C. Guo, and C. S. Jensen, "Travel cost inference from sparse, spatio temporally correlated time series using markov models," in *PVLDB 2013*.

[24] C. M. Grinstead and J. L. Snell, *Introduction to probability*. American Mathematical Soc., 2012.

[25] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[26] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[27] S. J. Wright, "Coordinate descent algorithms," *Mathematical Programming*, 2015.

[28] R. W. Ouyang, L. Kaplan, P. Martin, A. Toniolo, M. Srivastava, and T. J. Norman, "Debiasing crowdsourced quantitative characteristics in local businesses and services," in *IPSN'15*.

[29] R. W. Ouyang, L. M. Kaplan, A. Toniolo, M. Srivastava, and T. J. Norman, "Aggregating crowdsourced quantitative claims: Additive and multiplicative models," *TKDE'16*.

[30] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *IJCAI'95*.

[31] P. Richtárik and M. Takáč, "Parallel coordinate descent methods for big data optimization," *Mathematical Programming*, 2015.

[32] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1996.

[33] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon, "Collaborative filtering with graph information: Consistency and scalable methods," in *NIPS'15*.

[34] D. M. Hawkins, "The problem of overfitting," *Journal of chemical information and computer sciences*, 2004.

[35] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, 2003.