Crowdsourcing-Based Real-Time Urban Traffic Speed Estimation: From Trends to Speeds

Huiqi Hu†, Guoliang Li†, Zhifeng Bao‡, Yan Cui†, Jianhua Feng†
†Department of Computer Science, Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing, China
‡Computer Science and Information Technology, RMIT University, Melbourne, Australia
{hhq11, cuiy12}@mails.tsinghua.edu.cn; {liguoliang,fengjh}@tsinghua.edu.cn; zhifeng.bao@rmit.edu.au

***Abstract*—Real-time urban traffic speed estimation provides significant benefits in many real-world applications. However, existing traffic information acquisition systems only obtain coarse-grained traffic information on a small number of roads but cannot acquire fine-grained traffic information on every road. To address this problem, in this paper we study the traffic speed estimation problem, which, given a budget K, identifies K roads (called seeds) where the real traffic speeds on these seeds can be obtained using crowdsourcing, and infers the speeds of other roads (called non-seed roads) based on the speeds of these seeds. This problem includes two sub-problems: (1) Speed Inference – How to accurately infer the speeds of the non-seed roads; (2) Seed Selection – How to effectively select high-quality seeds. It is rather challenging to estimate the traffic speed accurately, because the traffic changes dynamically and the changes are hard to be predicted as many possible factors can affect the traffic. To address these challenges, we propose effective algorithms to judiciously select high-quality seeds and devise inference models to infer the speeds of the non-seed roads. On the one hand, we observe that roads have correlations and correlated roads have similar traffic trend: the speeds of correlated roads rise or fall compared with their historical average speed simultaneously. We utilize this property and propose a two-step model to estimate the traffic speed. The first step adopts a graphical model to infer the traffic trend and the second step devises a hierarchical linear model to estimate the traffic speed based on the traffic trend. On the other hand, we formulate the seed selection problem, prove that it is NP-hard, and propose several greedy algorithms with approximation guarantees. Experimental results on two large real datasets show that our method outperforms baselines by 2 orders of magnitude in efficiency and 40% in estimation accuracy.**

I. INTRODUCTION

Real-time urban traffic speed estimation plays an important role in many real-world applications, e.g., navigation systems and online map services. For example, on-the-fly route planning with real-time traffic information can guide users to avoid the traffic jams, which not only shortens the travel time, but also saves energy and reduces the air pollution. Existing traffic information acquisition systems rely on the static traffic sensors (e.g., surveillance cameras and inductive loops) or vehicle GPS records (e.g., taxi trajectories) to detect the real-time traffic information [9]. However, the coverage of existing traffic speed information is not sufficient due to the expensive maintenance cost of traffic sensors [1]. For example, there are more than two millions road segments in Beijing (for simplicity, we use roads to replace road segments), but there are only 22 thousand traffic sensors in Beijing. Thus the traditional systems only get coarse-grained traffic information on a small number of roads (e.g., expressways and main roads). To increase the coverage of the real-time traffic information, it calls for an effective method to obtain the fine-grained traffic information on every road. Such a demanding requirement has inspired both industry and academic communities to leverage crowdsourcing to improve traffic management [2], [15], by collecting traffic information through user-generated GPS data from crowdsourced drivers. Google Traffic and MIT CarTel[1] are real applications that have established the crowdsourcing approach to benefit our daily life.

In this paper, we study the traffic speed estimation problem, which, given a budget K, identifies K roads (called seeds) where we assume that we can obtain the real traffic speeds on these seeds via crowdsourcing acquisition methods and use them to infer the speeds of other roads (called non-seed roads). It further reduces to two sub-problems: (1) *Speed Inference* – How to accurately infer the speeds of the non-seed roads based on the speeds of seeds; (2) *Seed Selection* – How to select K high-quality seeds in order to improve the quality of speed inference. For example, we first select K seeds to collect the traffic speeds, then we ask the drivers on those seeds to report their speeds by paying them certain monetary awards.

It is rather challenging to estimate the traffic speeds accurately. First, the traffic changes dynamically and the changes are hard to be predicted, because many possible factors can affect the traffic, e.g., incidents, road maintenance and weather. To infer the speeds of non-seed roads, existing studies [24], [14] assumed that adjacent roads have similar speeds, and they utilized the speeds on the adjacent roads to infer the speeds of the non-seed roads. However, we observe that this assumption is too strict and usually not applicable to real traffic. For example, the entrance road to the main road usually has lower speed than the main road. The road to the downtown usually has lower speed than the exit road from the downtown. Second, many factors can influence traffic speeds, and it is hard to effectively model the real traffic and select high-quality seeds.

To address these challenges, we propose effective algorithms to judiciously select high-quality seeds and devise inference models to infer the speeds of non-seed roads. In particular, we make the following contributions. (1) We observe that roads have correlations and correlated roads usually have similar traffic trends: the speeds of correlated roads rise or fall compared with their historical average speed simultaneously. We utilize this property to propose a traffic trend correlation model. (2) We propose a two-step model to infer the traffic speed. In step 1 we adopt a graphical model to infer the traffic trend of a road v and in step 2 we devise a hierarchical linear model to estimate the traffic speed of v based on its traffic

[1] https://en.wikipedia.org/wiki/Cartel

trend. (3) We formulate the seed selection problem, prove that it is NP-hard, and propose several greedy algorithms with approximation guarantees. (4) We have conducted experiments on real datasets and the experimental results show that our method significantly outperforms state-of-the-art approaches.

The rest of this paper is organized as follows. We formulate the problem and review related work in Section II. Section III presents our observation. We propose the speed inference method in Section IV. Section V discusses the seed selection strategy. Experimental results are reported in Section VI. We conclude the paper in Section VII.

II. Preliminary

A. Problem Formulation

Road Network. We model a road network as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is a set of vertices (e.g., crossroads) and $\mathcal{E}$ is a set of roads. We denote $\mathcal{E} = \{x^1, x^2, \cdots, x^{|\mathcal{E}|}\}$, where $|\mathcal{E}|$ is the number of roads and x^i is a specific road.

Traffic Speed. At time t, road x^i has a traffic speed v_t^i. We normalize the speed to a number in $[0, 1]$[2]. For simplicity, we simplify the symbol v_t^i by omitting the superscript i and the subscript t if there is no ambiguity, i.e., x denotes a specific road and v is the traffic speed on road x at time t.

Problem Definition. Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the traffic estimation problem selects a K-size subset of roads $\mathcal{S} \subset \mathcal{E}$. $\mathcal{S}$ is called the seed set and each road in $\mathcal{S}$ is called a seed. The speed of each seed is known and we want to estimate the speeds of roads in $\mathcal{E} - \mathcal{S}$ (called non-seed roads). Given a non-seed road x, suppose its real speed is v and its estimated speed is $\hat{v}$. We use the well-known mean absolute percentage error (**MAPE**) to evaluate the quality of estimated speeds, which is defined as below:

$$\mathsf{MAPE} = \frac{1}{|\mathcal{E} - \mathcal{S}|} \sum_{x \in \mathcal{E} - \mathcal{S}} \frac{|\hat{v} - v|}{v}.$$

Definition 1: Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, the traffic speed estimation problem is to select a K-size subset of roads, $\mathcal{S} \subset \mathcal{E}$, such that the MAPE for all the rest roads $x \in \mathcal{E} - \mathcal{S}$ (i.e., $\sum_{x \in \mathcal{E} - \mathcal{S}} \frac{|\hat{v} - v|}{v}$) can be minimized when $\mathcal{S}$ is used to estimate the speed of $x \in \mathcal{E} - \mathcal{S}$.

The traffic speed estimation problem includes two sub-problems. (1) How to effectively estimate the speeds of the non-seed roads given the speeds of the selected seeds. (2) How to judiciously select k high-quality seeds. To address these problems, we first introduce a traffic correlation model in Section III and then discuss how to address these two problems in Sections IV and V respectively.

B. Related Work

Traffic Estimation. Existing traffic estimation methods can be broadly classified into two categories: (1) future traffic estimation [16], [20], [21], [5], [7], [18], [11], [10], [13], [19]; (2) current traffic estimation [24], [14], [25].

For future traffic estimation, the problem has been widely studied by transportation field and data mining field. A general method was based on some time series model, e.g. Bayesian network models [7], historic average (HA) models [18], the hidden Markov model [20] or the ARMA (Auto-Regressive-Moving-Average) model [19], [13], which considered both the temporal information and the road features (e.g., the road structures and the traffic signals) that affected the traffic evolvement over time. They focused on predicting short or long term future traffic and most of them were based on an assumption that they were clearly aware of the current traffic. Unfortunately, they had only limited traffic coverage over the large-scale urban road networks in reality [17], making it non-trivial to obtain the current traffic for each road.

For current traffic estimation, with limited amount of observed data by using probing data and traffic sensors, existing methods [24], [25], [23] utilized KNN methods to infer the speeds of unknown roads simply based on their spatial neighbors with known speeds. In recent studies, Zheng et. al. [24], [17], [14] modeled the traffic on a road network with a road-time matrix and proposed a matrix factorization based method by incorporating other traffic related features which include the location of roads and the distribution of nearby points of interest. They assumed that adjacent roads had similar traffic speeds and collaboratively factorized the road-time matrix (or the driver-road-time cube), the road-feature matrix and the time-grid matrix. The traffic speed was estimated by filling in the missing values in the road-time matrix.

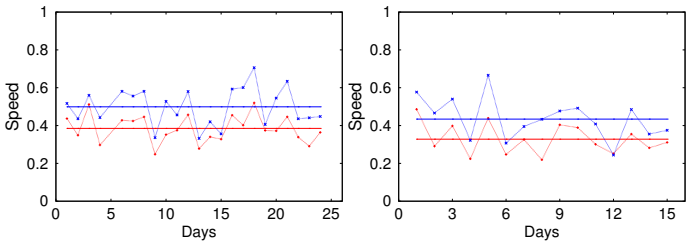
Our work differs from existing works in two aspects, thereby enabling a more accurate estimation of the traffic speed: (1) we drop out a common intuition adopted by existing work, i.e., the correlated roads usually have *similar speeds*, which is not statistically significant according to our observation (see Section III); instead we utilize a more reasonable observation: the correlated roads usually have *similar trends*. (2) We study how to judiciously select seeds to further improve the accuracy, while none of existing works is aware of it.

Semi-Supervised Learning with Graph Regularization. Inferring the values of unknown edges given some known edges in a graph was related to the semi-supervised learning on graphs. Graph regularization with laplacian matrix [8], [3] was the state-of-the-art technique. The general idea was that if two edges were connected, their estimation would be similar. Based on this assumption, these methods can be used to estimate the speeds of unknown roads by exploiting the speeds of the known neighboring roads. However, as we can see from Section VI later, the speed correlation is not as useful as our proposed traffic trend correlation in Section III.

III. Traffic Correlations

We observe that the traffic speed of a road dynamically changes around its average speed in the history and the speeds of adjacent roads (sharing a common vertex) have high correlations. We use a real taxi dataset of Beijing to illustrate the historical traffic (see Section VI for details of the dataset). Figure 1(a) illustrates the traffic speeds of two randomly selected adjacent roads at 5pm in 25 workdays and Figure 1(b) shows their speeds in 15 weekends, where the straight lines are the average speeds of the two roads and the two curves are the speeds of the two roads on 5pm in every day. We can see that the speeds of the two roads dynamically change around their average speeds. Furthermore, if the speed of a road is larger than the average speed, the speed of its adjacent road is also larger than the average speed, and vice

[2]We divide the speed by a maximum speed (e.g., $100km/h$).



(a) Workday (b) Weekend

Fig. 1. Traffic Correlations of 2 Adjacent Roads.

versa. Thus *the speeds of the two roads are highly correlated on both workdays and weekends*.

We further exploit this observation. Let $\bar{v}$ denote the average speed of road x at time t on every workday or weekend, which can be computed based on the historical data. If two roads are highly correlated, their speeds should have similar trends, i.e., rising or falling almost simultaneously compared with the average speed. Thus, the larger the percentage that the speeds of x^i and x^j both rise or fall is, the higher the correlation between x^i and x^j is. We then define the correlation score between roads x^i and x^j as

$$\textsc{Cor}(x^i, x^j) = \frac{\textsc{Cnt}(v^i \geq \bar{v}^i, v^j \geq \bar{v}^j) + \textsc{Cnt}(v^i < \bar{v}^i, v^j < \bar{v}^j)}{\textsc{TotalCnt}}, \tag{1}$$

where $\textsc{Cnt}(v^i \geq \bar{v}^i, v^j \geq \bar{v}^j)$ ($\textsc{Cnt}(v^i < \bar{v}^i, v^j < \bar{v}^j)$) denotes the number of times that the speeds of x^i and x^j both rise (fall). TOTALCNT is the total number of times that v^i and v^j are both observed in the historical data. Intuitively, the closer the roads are, the larger their correlation is.

Next, we quantify the distance between two roads in the following way: we transform the graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ to a *reverse graph* $\mathcal{G}' = \{\mathcal{V}', \mathcal{E}'\}$, where each vertex in $\mathcal{V}'$ corresponds to a road in $\mathcal{E}$ and there is an edge between two vertexes in $\mathcal{V}'$, if their corresponding roads share a common vertex in $\mathcal{G}$. Then we define the *hop (distance)* between two roads, which is the length of the shortest path of their corresponding vertexes in the reverse graph $\mathcal{G}'$.

Definition 2 (h-hop Neighbor): A road is an h-hop neighbor of x, if its distance to x is exactly h.

Next we use an experiment to show the correlation scores between roads with different hops. We first randomly select $10,000$ roads and then pick their h-hop neighbors for $1 \leq h \leq 6$. For each road and its h-hop neighbors, we compute their correlation scores. Table I(a) shows the distributions of correlation scores. For example, for 1-hop road pairs, 32.9% pairs have correlation scores in [0.8, 1.0].

We have the following observations. First, the roads with small distance have fairly strong correlation. For the roads and their 1-hop neighbors, the percentage of pairs with correlation scores larger than 0.7 is 67%; for the roads and their 2-hop neighbors, the percentage of pairs with correlation scores larger than 0.7 is 51.7%. Second, the correlation becomes weaker if their distance is larger, i.e., the smaller the distance between two roads is, the larger correlation their traffic trend is. For example, for $2/3$-hop neighbors, many pairs have correlation scores over 0.7, while few pairs for 4/5/6-hop neighbors.

For comparison, we also investigate the distribution of the relative speed differences between the $10,000$ roads and their h-hop neighbors (for road x and its neighbor x^j, the relative speed difference is $\frac{|v^j - v|}{v}$), as shown in Table I(b). Only 29.5%

(a) Correlation Scores on h-hop Roads.

	$[0, 0.5)$	$[0.5, 0.6)$	$[0.6, 0.7)$	$[0.7, 0.8)$	$[0.8, 1.0]$
1 hop	5.7%	8.6%	18.7%	36.1%	30.9%
2 hop	8.7%	11.6%	28.0%	32.0%	19.7%
3 hop	10.4%	15.5%	31.8%	28.7%	13.6%
4 hop	29.7%	25.2%	20.2%	19.3%	5.6%
5 hop	43.2%	34.1%	17.0%	4.1%	1.6%
6 hop	48.0%	33.5%	14.1%	3.6%	0.8%

(b) Relative Speed Differences on h-hop Roads.

	$[0, 0.2)$	$[0.2, 0.3)$	$[0.3, 0.4)$	$[0.4, 0.5)$	$[0.5, +\infty]$
1 hop	29.5%	29.4%	15.1%	12.8%	13.2%
2 hop	21.3%	30.5%	18.8%	15.3%	14.1%
3 hop	17.4%	27.1%	20.7%	17.1%	17.7%
4 hop	16.1%	26.5%	23.8%	16.4%	17.2%
5 hop	15.6%	24.7%	24.0%	17.4%	18.3%
6 hop	15.4%	22.8%	24.2%	17.9%	19.7%

TABLE I. TRAFFIC CORRELATIONS

1-hop neighbors have relative difference within 20%, while about 26% 1-hop neighbors have relative difference over 0.4. According to the statistics, if we directly use these 1-hop neighbors to estimate the traffic speed, the estimation difference can be over 30%. The speed difference also increases with larger hops. For instance, the percentage within 0.2 for the 2-hop neighbors decreases to 21.3%.

From the results, we have the following observations. *First, we cannot directly utilize the speeds of the h-hop neighbors to effectively estimate the traffic speed. Second, close roads usually have strong traffic correlations. Specifically, the roads with small distances have fairly large correlation on traffic trend and the correlation becomes weak if the distance is large.* These observations motivate us to use another option: we first use the trend of the h-hop neighbors of a road v to infer the traffic trend of v, based on which, we further estimate the traffic speed of v. Next we introduce how to infer the traffic trend and the speed of v based on those correlated neighbors.

IV. TRAFFIC SPEED INFERENCE

In this section, we study how to infer the traffic speeds of non-seed roads based on the traffic speeds of seeds. Based on the traffic correlation, we propose a two-step model. The first step is a *traffic trend inference model*, which infers the traffic trend: the speed rises or falls compared with the average speed (Section IV-A). The second step is a *traffic speed learning model*, which learns the speed based on the traffic trend inference model (Section IV-B).

A. Traffic Trend Inference

We utilize a Markov random field [11] to infer the traffic trend. For each road x, we construct a probability graphical model based on the roads that may affect the trend of x. Each node in the graphical model corresponds to a road and represents the traffic trend of the road, which is a binary random variable within domain $\{+1, -1\}$, where $+1$ denotes that the traffic speed of x has a rising trend at time t and -1 denotes that the speed of x has a falling trend at time t compared with the average speed $\bar{v}$. Formally, we use Δv to represent the traffic trend of x, which is defined as

$$\Delta v = \begin{cases} +1 & \text{if } v - \bar{v} > 0 \\ -1 & \text{if } v - \bar{v} < 0 \end{cases} \tag{2}$$

Before we discuss which roads may influence the traffic trend of x, we first introduce several concepts.

Definition 3 (Correlated Roads): A road is called a correlated road of x, if its correlation score to x is larger than a threshold τ (e.g., $\tau = 0.7$). We denote the set of correlated roads of x by $\mathcal{C}(x)$.

Definition 4 (Correlated Seeds): A seed is called a correlated seed of x if it is a correlated road of x.

Definition 5 (H-hop Correlated Roads/Seeds): A road (seed) is called an h-hop correlated road (seed) of x, if it is not only a correlated road (seed) of x, but an h-hop neighbor of x. The set of h-hop correlated roads of x is denoted by $\mathcal{C}_h(x)$.

To infer the trend of x based on its observed correlated seeds, a naive approach is to utilize a voting strategy, i.e. if most of the seeds in $\mathcal{C}(x)$ have rising trend, then Δv is estimated as $+1$ and vice versa. However, this solution is not very accurate as it does not consider the connections between roads. Consider x and its 1-hop neighbor x^i and 2-hop neighbor x^j expanded from x^i. Suppose x^j is only connected to x through x^i and both x^i and x^j are correlated to x, then using both x^i and x^j (e.g. counting twice for voting) to infer x is biased as x^j can only affect x through x^i and their trend correlations are redundantly used.

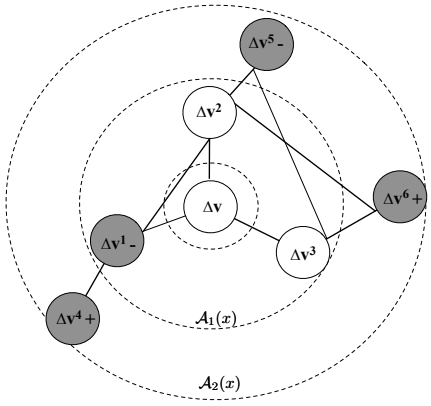
To address this problem, we build a two-layer Markov network. First, only the 1-hop correlated roads of x can directly affect the traffic trend of x and they separate x from other roads. We call such roads as *layer-1 impact nodes*, denoted as $\mathcal{A}_1(x)$. Second, other roads that can indirectly affect the traffic trend of x must be through the 1-hop correlated roads of x and are the correlated roads of $\mathcal{C}_1(x)$, i.e., $\cup_{x' \in \mathcal{C}_1(x)} \mathcal{C}(x')$. However, if a road in the set is not a seed, then its traffic trend is unknown, and using its estimated traffic trend may lead to inaccurate estimations of the traffic trend of x. Therefore, only roads in $\cup_{x' \in \mathcal{C}_1(x)} \mathcal{C}(x') \cap \mathcal{S}$ can indirectly influence the trend of x, namely the *layer-2 impact nodes* of x, denoted as $\mathcal{A}_2(x)$.

In our graphical model, $\mathcal{A}_1(x) = \mathcal{C}_1(x)$ and $\mathcal{A}_2(x) = \cup_{x' \in \mathcal{C}_1(x)} \mathcal{C}(x') \cap \mathcal{S}$; the roads in $\mathcal{A}_1(x)$ and $\mathcal{A}_2(x)$ are used to infer the traffic trend of x. We will further discuss why we choose 2-layer model at the end of Section IV-A. Next we formally introduce how to construct the graphical model.

Definition 6 (Graphical Model for Trend Inference): For each road x, we construct a two-layer graphical model $\mathcal{U}(\mathcal{U}^n, \mathcal{U}^e)$, where nodes $\mathcal{U}^n$ include (1) $\{x\}$, (2) layer-1 impact nodes $\mathcal{A}_1(x)$, (3) layer-2 impact nodes $\mathcal{A}_2(x)$. Each node represents the trend of its corresponding road; And edges $\mathcal{U}^e$ include (1) edges between x and its layer-1 impact nodes, i.e., $\{(x, x_1) | x_1 \in \mathcal{A}_1(x)\}$, (2) edges among layer-1 impact nodes, i.e., $\{(x_1, x_1') | x_1 \in \mathcal{A}_1(x), x_1' \in \mathcal{A}_1(x), x_1$ is a correlated road of $x_1'\}$, (3) edges between layer-1 and layer-2 nodes, i.e., $\{(x_1, x_2) | x_1 \in \mathcal{A}_1(x), x_2 \in \mathcal{A}_2(x)\}$, x_2 is a correlated road of $x_1\}$.

Figure 2 illustrates an example of the graphical model of x, where the solid nodes are seeds. For example, suppose x has three 1-hop correlated roads x^1(x^1 *is also a seed*), x^2, x^3. x^1 has a correlated seed x^4 and a correlated road x^2. x^2 has two correlated seeds x^5 and x^6. x^3 also has two correlated seeds x^5 and x^6.

In our graphical model, if x and a road in layer-2 are separated by a seed in layer-1, then according to the *Local Markov property* of the Markov random field [11], they are conditionally independent. For example, Δv^4 and Δv are separated by seed Δv^1, thus Δv and Δv^4 are independent given Δv^1.



Assignment	Probability
$\Delta v = +1, \Delta v^2 = +1, \Delta v^3 = +1$	0.0421
$\Delta v = +1, \Delta v^2 = +1, \Delta v^3 = -1$	0.0105
$\Delta v = +1, \Delta v^2 = -1, \Delta v^3 = +1$	0.0421
$\Delta v = +1, \Delta v^2 = -1, \Delta v^3 = -1$	0.0105
$\Delta v = -1, \Delta v^2 = +1, \Delta v^3 = +1$	0.0105
$\Delta v = -1, \Delta v^2 = +1, \Delta v^3 = -1$	0.0421
$\Delta v = -1, \Delta v^2 = -1, \Delta v^3 = +1$	0.1684
$\Delta v = -1, \Delta v^2 = -1, \Delta v^3 = -1$	0.6737

Fig. 2. Example of Graphical Model.

Given a graphical model $\mathcal{U}$ of road x, we model the Markov random field as a product of all edge potentials [11][3] with

$$\mathcal{P}(\mathcal{U}) = \frac{1}{\mathrm{Z}} \prod_{<x^i, x^j> \in \mathcal{U}^e} \psi_{x^i, x^j}(\Delta v^i, \Delta v^j), \tag{3}$$

where $\mathcal{P}(\mathcal{U})$ is the distribution of an assignment over nodes in $\mathcal{U}$ (in an assignment, each node is assigned with a traffic trend +1 or -1), Z is the partition function to normalize the probability of all the assignments, and $\psi_{x^i, x^j}(\Delta v^i, \Delta v^j)$ is the edge clique potential function between roads x^i and x^j, which reflects the relevance of traffic trends Δv^i and Δv^j. We utilize the correlation score to define the potential function:

$$\psi_{x^i, x^j}(\Delta v^i, \Delta v^j) = \begin{cases} \textsc{Cor}(x^i, x^j) & (\Delta v^i = \Delta v^j) \\ 1 - \textsc{Cor}(x^i, x^j) & (\Delta v^i \neq \Delta v^j) \end{cases} \tag{4}$$

In the above function, if two nodes have the same traffic trend, the potential function assigns a high relevance value based on their correlation score (see Table I); otherwise, the function assigns a low relevance value.

Next, given the constructed graphical model and the seeds, we utilize the MAP (Maximum a Posterior [11]) inference to infer the traffic trend of the non-seed roads. The MAP inference aims to find an assignment that maximizes the posterior probability of the traffic trend on non-seed roads given those on the seeds, as formally defined below.

Definition 7 (Traffic Trend Inference): Given a road x, its graphical model $\mathcal{U}$, and a seed set $\mathcal{S}$, let $\mathcal{U}^n - \mathcal{S}$ denote the set of non-seed roads in $\mathcal{U}$ and $\mathcal{U}^n \cap \mathcal{S}$ be the set of seeds

[3]A Markov random field is usually factorized over its clique potentials. In our model, a clique potential is considered as the product of all its edge potentials. Thus, our graphical model is simply expressed as the product of all edge potentials.

Algorithm 1: TRAFFICTRENDINFERENCE()

Input: x: any non-seed road.
$\mathcal{U}$: the graphical model for road x.
Output: Δv: the inferred trend of x

```
1  p_max = 0;
2  Δv_max = −1;
3  foreach assigment {Δv, ··· , Δv^i, ···} of U^n − S do
4      p = 0;
5      foreach < x^i, x^j > ∈ U^e do
6          p = p + log(ψ_{x^i,x^j}(Δv^i, Δv^j));
7      if p > p_max then
8          p_max = p;
9          Δv_max = Δv;
10 return Δv_max;
```

in $\mathcal{U}$. It aims to find a uniform distribution over all the nodes by maximizing the probability of the traffic trend of roads in $\mathcal{U}^n - \mathcal{S}$, i.e.,

$$\arg\max_{(\mathcal{U}^n-\mathcal{S})} \mathcal{P}\big((\mathcal{U}^n - \mathcal{S})|(\mathcal{U}^n \cap \mathcal{S})\big). \tag{5}$$

Since $x \in \mathcal{U}^n - \mathcal{S}$, we can infer Δv for road x. Specifically, if there is no seed in the graphical model $\mathcal{U}$, i.e., $\mathcal{U}^n \cap \mathcal{S} = \emptyset$, we cannot estimate the traffic trend of x and thus we simply return the average speed $\bar{v}$ as the estimation speed.

Example 1: Consider the Markov random field in Figure 2. Suppose all the correlation scores in the graph are 0.8. We want to maximize the probability $\mathcal{P}(\Delta v, \Delta v^2, \Delta v^3 | \Delta v^1 = -1, \Delta v^4 = +1, \Delta v^5 = -1, \Delta v^6 = +1)$. We can get the best assignment by enumerating Δv =+1/-1, Δv^2 =+1/-1 and Δv^3 =+1/-1, and the maximum assignment is $\Delta v = -1, \Delta v^2 = -1$ and $\Delta v^3 = -1$. Therefore, the trend of v in this example is inferred as -1 (i.e., a falling trend).

The problem of finding the maximum assignment in a Markov network is NP-hard [11]. Fortunately, recall Definition 6 that the non-seed roads in our graphical model are only limited to the 1-hop neighbors of a road, and the maximum number is usually within 10. As the computational complexity of the inference model is $\mathcal{O}(2^n)$, if $n \leq 10$, we can simply enumerate the possible assignments or use the variable elimination [11], [4] to do inference.

Algorithm 1 presents how to infer the trend of a road x by enumerating all possible assignments for roads in $\mathcal{U} - \mathcal{S}$(lines 3-9). According to Bayesian equation, maximizing $\mathcal{P}\big((\mathcal{U}^n - \mathcal{S})|(\mathcal{U}^n \cap \mathcal{S})\big)$ is equivalent to maximizing $\mathcal{P}\big((\mathcal{U}^n - \mathcal{S}), (\mathcal{U}^n \cap \mathcal{S})\big) = \mathcal{P}(\mathcal{U})$ in Equation 3. Therefore, we compute the log likelihood of the function by using the log-sum form instead of the product of probability to prevent float underflow (lines 5 and 6). Finally, we return Δv with the maximum assignment (line 10).

Discussion on choice of number of layers. We utilize a two-layer graphical model to identify the correlated roads of x and use the MAP inference to infer its trend. An alternative but more complex method is to construct a multiple layer graph by utilizing k-hop roads as $\mathcal{A}_1(x) = \mathcal{C}_1(x), \mathcal{A}_2(x) = \cup_{x' \in \mathcal{A}_1(x)} \mathcal{C}_1(x'), \cdots \mathcal{A}_k(x) = \cup_{x' \in \mathcal{A}_{k-1}(x)} \mathcal{C}_1(x'), \mathcal{A}_{k+1}(x) = \cup_{x' \in \mathcal{A}_k(x)} \mathcal{C}(x')$. However, it will involve many unobserved roads and the number of neighbors grows exponentially with the expansion of hops. Also for a road x, it can only be assigned to a few seeds (see Section VI). Therefore, this complex inference model will be ineffective when there are more hidden nodes than observed nodes in the Markov network, especially when the hidden nodes are centered at x. Moreover, the inference efficiency will be low, as the inference complexity is $O(2^n)$, which is controlled by the number of hidden nodes. To this end, we only use $\mathcal{C}_1(x)$ in the model.

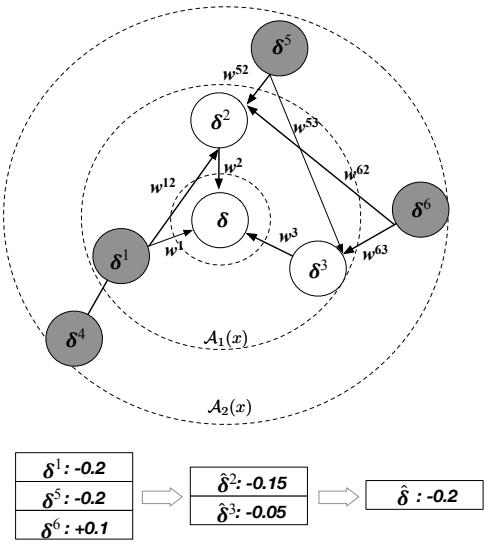


Fig. 3. Example of Speed Estimation.

B. Traffic Speed Estimation

We discuss how to use the traffic trend Δv to estimate the speed v. Let $\delta = |v - \bar{v}|$ denote the difference between the speed v and the historical average speed $\bar{v}$. We aim to estimate δ as accurately as possible. Suppose the estimated difference is $\hat{\delta}$ and we compute the estimated speed $\hat{v}$ by

$$\hat{v} = \begin{cases} \bar{v} + \hat{\delta} & (\Delta v > 0) \\ \bar{v} - \hat{\delta} & (\Delta v < 0) \end{cases} \tag{6}$$

To estimate $\hat{\delta}$ of a road x, we learn a hierarchical linear model for δ offline based on the two-layer graphical model and historical data by considering two cases: $\Delta v > 0$ and $\Delta v < 0$. As the methodology is the same for the two cases, here we only discuss the case of $\Delta v < 0$ and similar techniques can be used for $\Delta v > 0$.

Estimation of $\hat{\delta}$. Note that the speed difference δ actually depends on the differences of the correlated roads of x: if the traffic speeds of x's correlated roads decrease heavily, δ_i will be large and vice versa. To capture such dependence, we estimate $\hat{\delta}$ as a linear integration of the speed difference of its correlated roads. Given the two-layer graphical model $\mathcal{U}(\mathcal{U}^n, \mathcal{U}^e)$, the correlated roads of x are in $\mathcal{A}_1(x) \subset \mathcal{U}^n$. Therefore, we estimate $\hat{\delta}$ as

$$\hat{\delta} = \sum_{x^j \in \mathcal{A}_1(x)} w^j \cdot \hat{\delta}^j \tag{7}$$

where x^j is a correlated road of x, $\hat{\delta}^j$ is the speed difference of x^j, w^j is the weight of $\hat{\delta}^j$ (we will introduce how to learn the weights later) and $\hat{\delta}$ is the linear combination of $\hat{\delta}^j$ by considering the respective weights. Note that we use $\hat{\delta}^j$ rather than the real difference δ^j, as δ^j may not be fully observed. If x^j is a seed, then δ^j can be observed, otherwise if $x^j \notin \mathcal{S}$,

Algorithm 2: OFFLINEWEIGHTLEARNING

Input: x: a non-seed road;
$\mathcal{U}$: the graphical model for road x;
$\{\delta_1, \cdots, \delta_N\}$: the historical data for x;
$\{\delta_1^l, \cdots, \delta_N^l\}$: the historical data for $x^l \in \mathcal{U}^n \cap \mathcal{S}$.
Output: w^j, w^{lj}: learned weights to optimize $\mathcal{J}(w)$

foreach $x^j \in \mathcal{A}_1(x)$ **do**
  $w^j = \text{RANDOM}(0, 1)$;
foreach $(x^l, x^j) \in \mathcal{U}^e$, $x^l \in \mathcal{U}^n \cap \mathcal{S}$, $x^j \in \mathcal{A}_1(x)$ **do**
  $w^{lj} = \text{RANDOM}(0, 1)$;
while TRUE **do**
  for $i = 1$ *to* N **do**
    $\hat{\delta}_i$=SPEEDDIFFERENCEESTIMATION$(w, \delta_i^{1:|\mathcal{U}^n \cap \mathcal{S}|})$;
  $\hat{\delta}^{org} = \frac{1}{N}\sum_{i=1}^{N} \hat{\delta}_i$;
  foreach w^j **do**
    Compute $\frac{\partial \mathcal{J}(w)}{\partial w^j}$ by Equations 8, 13;
    $w^j = w^j - \alpha \cdot \frac{\partial \mathcal{J}(w)}{\partial w^j}$;
  foreach w^{lj} **do**
    Compute $\frac{\partial \mathcal{J}(w)}{\partial w^{lj}}$ by Equation 8;
    $w^{lj} = w^{lj} - \alpha \cdot \frac{\partial \mathcal{J}(w)}{\partial w^{lj}}$;
  for $i = 1$ *to* N **do**
    $\hat{\delta}_i$=SPEEDDIFFERENCEESTIMATION$(w, \delta_i^{1:|\mathcal{U}^n \cap \mathcal{S}|})$;
  $\hat{\delta}^{new} = \frac{1}{N}\sum_{i=1}^{N} \hat{\delta}_i$;
  if $|\hat{\delta}^{new} - \hat{\delta}^{org}| < \tau_{con}$ **then** break;

Function SPEEDDIFFERENCEESTIMATION$(w, \delta^{1:|\mathcal{U}^n \cap \mathcal{S}|})$

Input: w: weights including all w^j and w^{lj};
$\delta^{1:|\mathcal{U}^n \cap \mathcal{S}|}$: observed speed difference for $x^l \in \mathcal{U}^n \cap \mathcal{S}$.
Output: $\hat{\delta}$: estimation of speed difference.

foreach $x^j \in \mathcal{A}_1(x)$ **do**
  if $x^j \in \mathcal{S}$ **then** $\hat{\delta}^j = \delta^j$;
  else $\hat{\delta}^j = \sum_{w^{lj}} w^{lj} \cdot \delta^l$;
$\hat{\delta} = \sum_{w^j} w^j \cdot \hat{\delta}^j$;
return $\hat{\delta}$;

we should further estimate it as the linear integration of its correlated roads, which is formally computed as

$$\hat{\delta}^j = \begin{cases} \delta^j & (x^j \in \mathcal{S}) \\ \sum_{x^l \in \mathcal{U}^n \cap \mathcal{S} \ \&\ <x^l, x^j> \in \mathcal{U}^e} w^{lj} \cdot \delta^l & (x^j \notin \mathcal{S}) \end{cases} \quad (8)$$

where x^l is a correlated seed of x^j and δ^l is its observed speed difference, w^{lj} is the weight from x^l towards x^j. Given the graphical model of x, we first compute $\hat{\delta}^j$ with Equation 8, then estimate $\hat{\delta}$ with Equation 7.

Example 2: Consider the example in Figure 3, to estimate the speed difference for δ, we use three seeds in the graphical model, i.e. $\{\delta^1 = -0.2$ (i.e., the speed decrease by 0.2), $\delta^5 = -0.2, \delta^6 = 0.1\}$. Then it computes $\hat{\delta}^2$ and $\hat{\delta}^3$ with the weights $\{w^{12}, w^{52}, w^{62}, w^{53}, w^{63}\}$. Suppose all the weights in the example are learned as 0.5, then $\hat{\delta}^2$ and $\hat{\delta}^3$ are computed as $-0.2\times0.5-0.2\times0.5+0.1\times0.5=-0.15$ and $-0.2\times0.5 + 0.1\times0.5 = -0.05$ respectively. Finally, it aggregates $\delta^1, \hat{\delta}^2, \hat{\delta}^3$ by weights w^1, w^2, w^3 to calculate δ as -0.2.

Algorithm 3: ONLINETRAFFICSPEEDESTIMATION

Input: x: a non-seed road.
Output: $\hat{v}$: traffic speed of x.

$\mathcal{U}$=CONSTRUCTINFERENCEGRAPH$(\mathcal{A}_1(x), \mathcal{A}_2(x))$;
if $\mathcal{U}^n \cap \mathcal{S} = \emptyset$ **then return** $\hat{v} = \bar{v}$;
Δv=TRAFFICTRENDINFERENCE$(\mathcal{U})$; //Equation 5
$\hat{\delta}$ =SPEEDDIFFERENCEESTIMATION(Δv);
if $\Delta v > 0$ **then return** $\hat{v} = \bar{v} + \hat{\delta}$;
else return $\hat{v} = \bar{v} - \hat{\delta}$;

Speed Learning Model. We learn the weights in the model based on the regression of the historical data. We denote the parameters in the model by $\{w^j | x^j \in \mathcal{A}_1(x)\}$ and $\{w^{lj} | x^j \in \mathcal{A}_1(x) \cap \mathcal{S}, x^l \in \mathcal{U}^n \cap \mathcal{S}, (x^l, x^j) \in \mathcal{U}^e\}$. To simplify the description, we omit the domain of parameters when enumerating them, i.e. computing $\sum_{w^j} w^j$ denotes that we sum up all the w^j in its domain set.

Suppose we have N historical data records with falling trends on road x at time t, denoted by $\{\delta_1, \delta_2, \cdots, \delta_N\}$, where $\delta_i = \bar{v} - v^i$. Meanwhile, given the graphical model $\mathcal{U}$ of x, for any observed seed $x^l \in \mathcal{U}^n \cap \mathcal{S}$, we also have N historical observations $\{\delta_1^l, \delta_2^l, \cdots, \delta_N^l\}$. We learn w^j and w^{lj} by minimizing the square loss of the historical data, which is

$$\mathcal{J}(w) = \frac{1}{2N}\sum_{i=1}^{N}(\hat{\delta}_i - \delta_i)^2 + \frac{\lambda}{2}(\sum_{w^j}(w^j)^2 + \sum_{w^{lj}}(w^{lj})^2), \quad (9)$$

where w represents all the parameters of w^j and w^{lj}. $(\hat{\delta}_i - \delta_i)^2$ is the square loss between δ_i and its estimation $\hat{\delta}_i$. $\sum_{w^j}(w^j)^2 + \sum_{w^{lj}}(w^{lj})^2$ is the regularization item for w^j and w^{lj} to prevent overfitting and λ is the parameter to tune its importance. To optimize $\mathcal{J}(w)$, we use the classic back propagation algorithm [4] (which is widely used to train neural-networks) to learn w^j and w^{lj}. It iteratively updates $\hat{\delta}_i$, $\hat{\delta}_i^j$, w^j and w^{lj} based on equation 7, 8 and the derivation of $\mathcal{J}(w)$ on w^j and w^{lj}, which are computed by

$$\begin{aligned} w^j &= w^j - \alpha \cdot \frac{\partial \mathcal{J}(w)}{\partial w^j} \\ w^{lj} &= w^{lj} - \alpha \cdot \frac{\partial \mathcal{J}(w)}{\partial w^{lj}} \end{aligned} \quad (10)$$

where α is the learning rate for update (usually set by a very small value, e.g., 0.05). The derivations for w^i and w^{lj} can be computed separately. For a single record δ_i, we have

$$\begin{aligned} \frac{1}{2}\frac{\partial(\hat{\delta}_i - \delta_i)^2}{\partial w^j} &= (\hat{\delta}_i - \delta_i) \cdot \frac{\partial \hat{\delta}_i}{\partial w^j} = (\hat{\delta}_i - \delta_i) \cdot \frac{\partial \sum_{w^k} w^k \hat{\delta}_i^k}{\partial w^j} \\ &= (\hat{\delta}_i - \delta_i) \cdot \hat{\delta}_i^j \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{1}{2}\frac{\partial(\hat{\delta}_i - \delta_i)^2}{\partial w^{lj}} &= (\hat{\delta}_i - \delta_i) \cdot \frac{\partial \hat{\delta}_j}{\partial w^{lj}} = (\hat{\delta}_i - \delta_i) \cdot \frac{\partial \sum_{w^k} w^k \hat{\delta}_i^k}{\partial w^{lj}} \\ &= (\hat{\delta}_i - \delta_i) \cdot w^j \cdot \frac{\partial \sum_{x^{kj}} w^{kj} \delta_i^k}{\partial w^{lj}} \\ &= (\hat{\delta}_i - \delta_i) \cdot w^j \cdot \delta_i^l. \end{aligned} \quad (12)$$

$\frac{\partial \mathcal{J}(w)}{\partial w^j}$ and $\frac{\partial \mathcal{J}(w)}{\partial w^{lj}}$ are computed by

$$\begin{aligned}\frac{\partial \mathcal{J}(w)}{\partial w^j} &= \frac{1}{N}\sum_{i=1}^{N}((\hat{\delta}_i - \delta_i)) \cdot \hat{\delta}_i^j + \lambda w^j \\ \frac{\partial \mathcal{J}(w)}{\partial w^{lj}} &= \frac{1}{N}\sum_{i=1}^{N}(\hat{\delta}_i - \delta_i) \cdot w^j \cdot \delta_i^l + \lambda w^{lj}\end{aligned} \tag{13}$$

Algorithm 2 presents the training part of our speed estimation model. Given a road x, its graphical model $\mathcal{U}$ and the respective historical data, it first initializes all w^j and w^{lj} with random weights (lines 1-3), and then iteratively updates $\hat{\delta}_i$, w^j and w^{lj} until convergence (lines 5-18). In each iteration, it first computes the original $\hat{\delta}_i$ (lines 6 to 8) with the current weights using function SPEEDDIFFERENCEESTIMATION, which estimates $\hat{\delta}$ based on w^j , w^{lj} and the observed speed differences of the seeds. Then it updates w^j and w^{lj} respectively (lines 9-14). Next it computes a new $\hat{\delta}_i$ with the updated weights (lines 15-17) and checks whether it is converged (line 18).

Lastly, we propose our online traffic speed estimation in Algorithm 3. For each non-seed node x, we first construct the graphical model (line 1). If $\mathcal{U}^n \cap \mathcal{S} = \emptyset$, there is no seed in the graphical model and we simply estimate $\hat{v}$ by its average speed $\bar{v}$ (line 2); otherwise, we infer the trend Δv using Equation 5 (line 3) and estimate the speed distance $\hat{\delta}$ using function SPEEDDIFFERENCEESTIMATIONEQUATION based on Equations 7 and 8 (line 4). Finally we estimate the speed $\hat{v}$ based on $\bar{v}$ and $\hat{\delta}$ with Equation 6 (lines 5-6).

V. SEEDS SELECTION

In this section, we study how to judiciously select high-quality seeds. There are two desired features in seed selection: (1) *Large Coverage*; and (2) *High Support*. On the one hand, for each seed, the larger the number of its correlated roads is, the larger the number of roads that can be inferred by the seed is (called *coverage* of this seed). Thus we want to select the seeds with large overall coverage. On the other hand, for each non-seed road, the larger the number of its correlated seeds is, the larger the number of seeds that can be used to infer its speed is (called *support* of this road). Thus we want to select the seeds that can provide high support for each non-seed road.

However, there is a budget constraint that we can only select K seeds. This constraint makes the two factors contradict to each other. A large coverage will lead to a small support of some roads (because it selects the seeds that are correlated to as many roads as possible, and thus each road will have a small number of correlated seeds). On the contrary, high supports of some roads will result in a small coverage (because the selected seeds are only correlated to a small number of roads, while the rest have no correlated seeds). Threreby it calls for an effective selection method that makes a good balance between coverage and support. Next we will formally define these two factors and propose several selection strategies.

Recall the graphical model in Section IV-A, given a road x, we infer its trend Δv based on $\mathcal{A}_1(x) = \mathcal{C}_1(x)$ and $\mathcal{A}_2(x) = \cup_{x' \in \mathcal{C}_1(x)} \mathcal{C}(x') \cap \mathcal{S}$. Thus roads in $\mathcal{C}_1(x) \bigcup \cup_{x' \in \mathcal{C}_1(x)} \mathcal{C}(x')$ can be used to infer x. We then define the inference set and the support set of x that can be used to infer the speed of x.

Definition 8 (Inference Set): The inference set of road x is

$$\mathcal{I}(x) = \mathcal{C}_1(x) \bigcup \cup_{x' \in \mathcal{C}_1(x)} \mathcal{C}(x'). \tag{14}$$

Definition 9 (Support Set): The support set of a road x is $\mathcal{I}(x) \cap \mathcal{S}$. The support of x is the size of its support set, i.e., $\text{SUP}(x) = |\mathcal{I}(x) \cap \mathcal{S}|$.

Definition 10 (Coverage Set): The coverage set of a road x is the set of roads that have x in their inference sets, i.e., $\mathcal{I}^{-1}(x) = \{x^i | x \in \mathcal{I}(x^i)\}$. The coverage set of the seed set $\mathcal{S}$ is the set of roads that can be inferred from $x \in \mathcal{S}$, i.e.,

$$\mathcal{I}^{-1}(\mathcal{S}) = \bigcup_{x \in \mathcal{S}} \mathcal{I}^{-1}(x). \tag{15}$$

The coverage of $\mathcal{S}$ is the size of its coverage set, i.e., $\text{COV}(\mathcal{S}) = |\mathcal{I}^{-1}(\mathcal{S})|$.

Obviously, the larger the support $\text{SUP}(x)$ is, the more the seeds that can be used to infer the speed of x are; the larger $\text{COV}(\mathcal{S})$ is, the more the roads that can be inferred from $\mathcal{S}$ are. Our goal is to maximize both $\text{SUP}(x)$ and $\text{COV}(\mathcal{S})$.

First, we consider the problem of optimizing the overall support $\sum_x \text{SUP}(x)$ for all roads, and propose the SUPGREEDY algorithm to maximize $\text{SUP}(x)$.

SUPGREEDY. It is a greedy algorithm which iteratively selects K roads into $\mathcal{S}$. In each iteration, it selects the road with the largest inference set in $\mathcal{E} - \mathcal{S}$, i.e. x^i which maximizes $|\{x | x^i \in \mathcal{I}(x) \ \& \ x \in \mathcal{E} - \mathcal{S}\}|$. The algorithm can optimize the overall support since it maximizes the increase of $\sum_x \text{SUP}(x)$ for each iteration and $\sum_x \text{SUP}(x)$ will reduce if we replace the selected seeds with any other roads.

Next, we consider the problem of maximizing the coverage $\text{COV}(\mathcal{S})$. We find that the problem of maximizing $\text{COV}(\mathcal{S})$ is NP-hard as proved in Theorem 1.

Theorem 1: Given a budget K, the problem of selecting a K-size seed set $\mathcal{S}$ to maximize $\text{COV}(\mathcal{S})$ is NP-hard.

Proof: We first prove that the decision problem is NP-complete: given a budget K and an integer m, whether there exists a seed set $\mathcal{S}$ with $|\mathcal{S}| = K$ and $|\bigcup_{x \in \mathcal{S}} \mathcal{I}^{-1}(x)| \geq m$. Next we prove this decision problem by a reduction from an existing set-cover problem [6]. For an arbitrary set-cover instance with elements $\{x^1, x^2, \cdots, x^m\}$ and n sets $\{S_1, S_2, \cdots, S_n\}$, which asks whether there exist K subsets that contain all the m elements, we can construct a road network with $|\mathcal{E}| = m$ roads. If $n = m$, we set $\mathcal{I}^{-1}(x^i) = S_i$ $(1 \leq i \leq m)$; If $n < m$, we set $\mathcal{I}^{-1}(x^i) = S_i$ $(1 \leq i \leq n)$, $\mathcal{I}^{-1}(x^i) = \emptyset$ $(n+1 \leq i \leq m)$; if $n > m$, we set $\mathcal{I}^{-1}(x^i) = S_i$ $(1 \leq i \leq m-1)$ and $\mathcal{I}^{-1}(x^m) = \cup_{i=m}^{n} S_i$. Therefore, we can transform an arbitrary set-cover instance into an instance of our problem. Thus the decision problem is NP-complete. As this is an optimization problem, it is NP-hard. ■

With a linear combination of $\text{SUP}(x)$ and $\text{COV}(x)$, we formally define the seed selection problem as below.

Definition 11 (Seed Selection Problem): The seed selection problem is to maximize

$$\text{COV}(\mathcal{S}) + \alpha \sum_{x \in \mathcal{I}^{-1}(\mathcal{S})} \text{SUP}(x). \tag{16}$$

where α is a tuning parameter to balance the coverage and support. The large α is, the more important the support is.

We can prove that the seed selection problem is also NP-hard as formalized in Theorem 2.

Theorem 2: The seed selection problem is NP-hard.

Proof: Based on Theorem 1, we consider the special instance of the problem where $\alpha = 0$, which becomes the problem of maximizing $\text{COV}(\mathcal{S})$. Thus we can reduce the problem of maximizing $\text{COV}(\mathcal{S})$ to this problem. ■

Since the seed selection problem is NP-hard, we next discuss four approximation algorithms.

RANDOM. It randomly selects a set $\mathcal{S} \subset \mathcal{E}$ with $|\mathcal{S}| = K$, which neither maximizes $\text{SUP}(x)$ nor maximizes $\text{COV}(\mathcal{S})$.

MAXCOV. It is a greedy algorithm to maximize $\text{COV}(\mathcal{S}) = |\mathcal{I}^{-1}(\mathcal{S})|$, by selecting top-$K$ roads with the largest $|\mathcal{I}^{-1}(x)|$. But it neglects that the coverage between different roads have overlaps and thus cannot achieve high overall coverage.

COVGREEDY. It is also a greedy algorithm to maximize $\text{COV}(\mathcal{S})$. It iteratively selects K roads into $\mathcal{S}$, and in each iteration, it selects the road to maximize the increase of the coverage compared with the previous iteration. For example, in the i-th iteration, it selects x^i that maximizes

$$|\bigcup_{x\in(\mathcal{S}\cup x^i)} \mathcal{I}^{-1}(x)| - |\bigcup_{x\in\mathcal{S}} \mathcal{I}^{-1}(x)|. \tag{17}$$

HYBRIDGREEDY. This is a greedy algorithm to maximize $\text{COV}(\mathcal{S}) + \alpha \sum_{x\in\mathcal{I}^{-1}(\mathcal{S})} \text{SUP}(x)$. It iteratively selects K roads into $\mathcal{S}$, and in the i-th iteration, it selects x^i that maximizes

$$\Big(|\mathcal{I}^{-1}(\mathcal{S}\cup x^i)| + \alpha \sum_{x\in\mathcal{I}^{-1}(\mathcal{S}\cup x^i)} |\mathcal{I}(x)\cap(\mathcal{S}\cup x^i)|\Big) - \Big(|\mathcal{I}^{-1}(\mathcal{S})| + \alpha \sum_{x\in\mathcal{I}^{-1}(\mathcal{S})} |\mathcal{I}(x)\cap\mathcal{S}|\Big). \tag{18}$$

Theoretical Analyses on Greedy Algorithms. We can prove that the coverage and support functions satisfy the submodularity [22]: for any two seed sets $\mathcal{S}_1 \subset \mathcal{S}_2$, if we add an arbitrary seed x^i into $\mathcal{S}_1$ and $\mathcal{S}_2$, the increase of coverage and support on $\mathcal{S}_1$ must be larger than those on $\mathcal{S}_2$, i.e.,

$$\text{COV}(\mathcal{S}_1 \cup x^i) - \text{COV}(\mathcal{S}_1) \geq \text{COV}(\mathcal{S}_2 \cup x^i) - \text{COV}(\mathcal{S}_2),$$
$$\text{SUP}(\mathcal{S}_1 \cup x^i) - \text{SUP}(\mathcal{S}_1) \geq \text{SUP}(\mathcal{S}_2 \cup x^i) - \text{SUP}(\mathcal{S}_2).$$

where $\text{SUP}(\mathcal{S}_1 \cup x^i) = \sum_{x\in\mathcal{I}^{-1}(\mathcal{S}_1\cup x^i)} \text{SUP}(x)$.

Since the hybrid function (Equation 16) is a linear combination of the coverage and support, it is also submodular. Furthermore, the three functions are monotone. Thus, COVGREEDY (SUPGREEDY) has an approximation ratio of $1-1/e$ to maximize COV (SUP), and HYBRIDGREEDY has an approximation ratio $1-1/e$ to the seed selection problem [22].

The time complexity of the greedy algorithm is $\mathcal{O}(K*|\mathcal{E}|*|\mathcal{I}^{-1}(x)|)$, where $|\mathcal{I}^{-1}(x)|$ is the average size of the coverage set of x. The algorithm iteratively selects K seeds, and in each selection it takes $|\mathcal{E}|*|\mathcal{I}^{-1}(x)|$ times in selecting the best seeds to maximize the greedy functions.

VI. EXPERIMENTS

We evaluated our proposed techniques. Our experimental goal was to evaluate the effectiveness and efficiency of traffic trend estimation model and traffic speed estimation model.

A. Experimental Setup

1) Dataset and Evaluation Metrics: We used real datasets to evaluate our techniques.

Road Networks. We used two real road network datasets: (1) The road network of Beijing, which had $2,690,296$ roads and $1,282,156$ vertices. (2) The road network of Nanjing, which had $1,425,048$ roads and $631,200$ vertices.

Historical GPS Records. We used two real taxi datasets of Beijing and Nanjing[4]. The first contained 3.05 billions GPS records of taxi trajectories from Oct. 1, 2012 to Dec. 31, 2012 with $12,745$ taxies in Beijing and the other had 0.65 billions GPS records from Jan. 1, 2011 to Jan. 31, 2011 with $8,257$ taxies in Nanjing. Each GPS record included the taxi ID, the taxi location (i.e., longitude and latitude) and the taxi speed. Each taxi reported a record every two seconds. We projected the GPS records onto the road network using map-matching algorithms [12]. To measure the traffic in different time, we partitioned each day into $T = 288$ time intervals, i.e., taking every 5 minutes as a time interval t. Thus each GPS record belonged to a specific time interval. The traffic speed of a road was computed as the average speed of historical GPS records on the road at time t.

Test Data. To evaluate our method, we randomly selected $520K$ $(170K)$ speeds on non-seed roads in workdays and $230K$ $(170K)$ speeds in weekends as the test data for Beijing (Nanjing) dataset, and used the rest as training data.

Evaluation Metrics. We used MAPE in problem definition to evaluate the speed estimation accuracy.

2) Baselines: We compared four baseline approaches.

(1) Linear Regression (LR). It utilized the traffic speeds of the seeds as the training data to learn a linear model, considering two types of features: (i) roads, including the length, location, number of nearby points of interest; and (ii) the historical data of the corresponding workday/weekend. LR minimized the error between the estimation speed and the real speed:

$$\frac{1}{|\mathcal{S}|}\sum_{i=1}^{|\mathcal{S}|}(v^i - \theta^T x^i)^2 + \frac{\lambda_1}{2}||\theta||_2,$$

where x^i denoted the feature set of a seed, θ denoted the parameters for features, and $\frac{\lambda_1}{2}||\theta||_2$ was the norm regularization for parameters to avoid overfitting. We tuned λ_1 and selected the best value $\lambda_1 = 0.02$.

(2) Linear Regression with Graph Regularization (LR+GR). It modeled the speed estimation problem as a semi-supervised leaning problem on graphs [3]. By assuming that adjacent roads have similar speeds, it learned the following model:

$$\frac{1}{|S|}\sum_{i=1}^{|\mathcal{S}|}(v^i-\theta^T x^i)^2+\frac{\lambda_0}{2}\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\sum_{j=1}^{|\mathcal{E}|}(\theta^T x^i-\theta^T x^j)^2+\frac{\lambda_1}{2}||\theta||_2,$$

where x^i and x^j denoted the feature sets of two seeds and θ denoted the parameter for features. The equation included three parts. The first part was a linear regression function, the second part was the graph regularization function which tried to estimate the speeds of x^i and x^j as close as possible if they were adjacent, and the third part was the norm regularization for parameters to avoid overfitting. λ_0 and λ_1 were two

[4]http://www.datatang.com/data/45888

TABLE II. EVALUATION ON SEED SELECTION STRATEGIES.

(a) Coverage (%)

Seed Ratio	3%	6%	9%	12%	15%
RANDOM	10.2	20.1	28.3	35.6	39.5
MAXCOV	19.7	30.5	39.6	47.2	53.2
COVGREEDY	**63.1**	**76.3**	**85.9**	**91.4**	**95.7**
SUPGREEDY	31.1	41.5	48.5	54.9	60.8
HYBRIDGREEDY	55.1	68.6	79.9	89.7	92.6

(b) Average Support ($\frac{\sum_x \text{SUP}(x)}{\text{COV}(\mathcal{S})}$)

Seed Ratio	3%	6%	9%	12%	15%
RANDOM	1.76	1.82	1.84	1.87	1.92
MAXCOV	2.40	2.41	2.24	2.08	2.03
COVGREEDY	1.84	2.22	2.36	2.51	2.63
SUPGREEDY	**2.48**	**3.01**	**3.38**	**3.56**	**3.66**
HYBRIDGREEDY	2.23	2.74	2.98	3.07	3.16

(c) Coverage and Support ($\text{COV}(\mathcal{S}) + \sum_x \text{SUP}(x)$)

Seed Ratio	3%	6%	9%	12%	15%
RANDOM	0.07	0.14	0.20	0.26	0.29
MAXCOV	0.17	0.26	0.32	0.37	0.41
COVGREEDY	0.45	0.62	0.73	0.81	0.88
SUPGREEDY	0.28	0.42	0.54	0.64	0.72
HYBRIDGREEDY	**0.45**	**0.65**	**0.81**	**0.93**	**1**

parameters to balance the three parts. We tuned the parameters and set the best values as $\lambda_0 = 1$, $\lambda_1 = 0.02$.

(3) Collaborative Matrix Factorization Based Method [14] (TSE). TSE assumed that close roads had similar traffic speeds and utilized a matrix factorization based method.

(4) Average Speed in The History (AVG). This method directly utilized the historical average traffic speed to estimate the speed, i.e., $\hat{v}$ was estimated as $\bar{v}$.

B. Evaluation of Our Methods

We first evaluated our seed selection strategies, then tested our traffic trend inference method, and lastly evaluated our speed learning model. Here we only showed the results on Beijing dataset due to space limitation.

We needed to select an appropriate threshold τ for the correlation. If τ was small (e.g., less than 0.6), we got many loosely correlated roads to infer the speed; if τ was large (e.g., 0.8), we would get few highly correlated roads. We tuned the threshold and used the best value $\tau = 0.7$ after varying τ.

1) Evaluation of Seed Selection Strategies: We evaluated our seed selection strategies and compared the five algorithms proposed in Section V: RANDOM, MAXCOV, COVGREEDY, SUPGREEDY, HYBRIDGREEDY. We compared their coverage, support, and combined coverage and support, where coverage is the percentage of the number of covered roads and seeds ($|\text{COV}(\mathcal{S}) \cup \mathcal{S}|$) over the total number of roads which have co-occurrences with other roads in the historical data. Table II(a) showed the percent of the coverage of selected seeds to the total number of roads, i.e., $\frac{\text{COV}(\mathcal{S})}{|\mathcal{V}|}$, Table II(b) showed the average support for each road that had at least one correlated seed, i.e, $\frac{\sum_x \text{SUP}(x)}{\text{COV}(\mathcal{S})}$, and Table II(c) showed the combination of coverage and support, i.e., $\text{COV}(\mathcal{S}) + \alpha \sum_x \text{SUP}(x)$, where the values were normalized by dividing the maximum value (the value in the bottom right cell). Here, α was set to 1 and we evaluated how α affected the estimation quality in Section VI-B3.

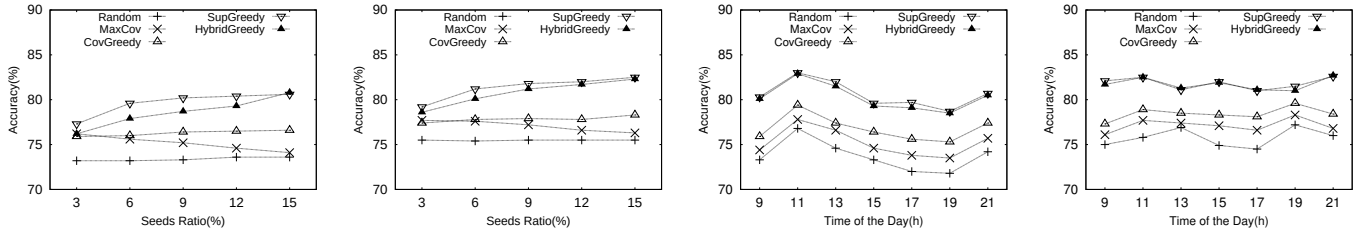
We had the following observations. (1) HYBRIDGREEDY achieved high performance on both coverage and support, and had the largest combination score of coverage and support, because its objective was to combine the two factors. In contrast, RANDOM had the lowest coverage and support as it randomly selected the seeds and did not optimize them at all. (2) COVGREEDY had the largest coverage among the five strategies as it was designed to maximize the number of correlated roads. For example, with 9% seeds, COVGREEDY can cover 85.9% roads. However, COVGREEDY had smaller support than SUPGREEDY as it focused on maximizing the coverage and did not consider the support. (3) SUPGREEDY achieved the largest support as it iteratively selected the seeds with the largest correlation for inference. (4) Both SUPGREEDY and MAXCOV had limited coverage, as they did not consider the overlap issue among the correlated roads of their selected seeds. (5) With more seeds selected, all methods achieved higher coverage and support except for the support of MAXCOV, probably because many roads with small supports are included when more seeds are selected.

2) Evaluation of Traffic Trend Inference Methods: We evaluated the accuracy of our traffic trend inference method (proposed in Section IV-A), equipped with each of the above five seed selection strategies. The accuracy was the ratio of correctly estimated trends to the total number of trends tested.

We first varied the seeds ratio from 3% to 15%, and the inference accuracy for workdays and weekends are shown in Figure 4(a) and 4(b). By linking them to Table II(b), we find: (1) The *support* was an important factor to achieve high trend inference accuracies. Specifically, SUPGREEDY and HYBRIDGREEDY outperformed other methods as they considered the support in the optimization function; SUPGREEDY was slightly better than HYBRIDGREEDY because HYBRIDGREEDY also considered the coverage which might decrease the accuracy (as some roads had small numbers of correlated seeds). (2) With the increase of seeds ratio, only the accuracy of MAXCOV did not increase as its support decreased slightly.

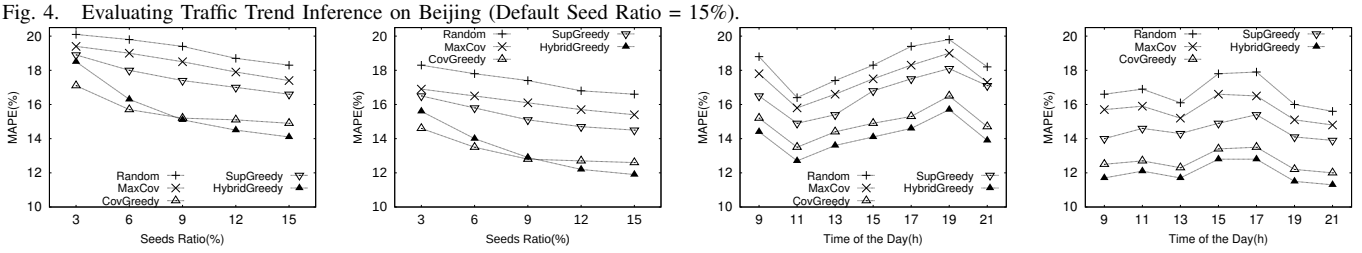
We further studied the inference accuracy by varying the time in a day from 9am to 9pm, as shown in Figures 4(c) and 4(d) (the default seeds ratio was 15%). We find: (1) Our inference method achieved high accuracy ($70\% - 85\%$). (2) High support of roads derived more correlated seeds, which in turn led to a high inference accuracy. Specifically, SUPGREEDY and HYBRIDGREEDY had higher accuracy than the rest as their support was larger. E.g., at 11am on workdays, HYBRIDGREEDY had an accuracy of 83% while the accuracy of RANDOM was 76%, because the roads of HYBRIDGREEDY had about 3.16 correlated seeds in average while RANDOM had only 1.92 correlated seeds (see Table II(b)). (3) At rush hours (9am, 5pm, 7pm), the accuracy was low as the traffic changed more dynamically then.

3) Evaluation of Traffic Speed Estimation Models: First, we compared the quality of the speed estimation models equipped with each of the five seed selection strategies in term of MAPE. Figure 5(a) and Figure 5(b) showed the results by varying the seeds ratio, where $\alpha = 1$ for HYBRIDGREEDY. We had three observations. (1) Recall Table II(b), a larger coverage led to a smaller MAPE and thus a better estimation. Thus COVGREEDY and HYBRIDGREEDY outperformed RANDOM, MAXCOV and SUPGREEDY. For example, with 15% seeds, HYBRIDGREEDY had a MAPE of 14.1% on workdays while the MAPE of RANDOM, MAXCOV SUPGREEDY, COVGREEDY were 18.3%, 17.4%, 16.6% and 14.9% respectively. Because COVGREEDY and HYBRIDGREEDY had larger coverage than other strategies, which can cover more roads



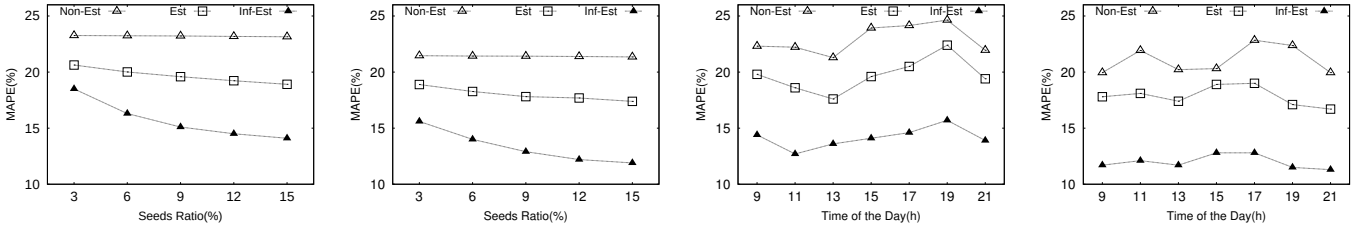
(a) Workdays (b) Weekends (c) Workdays (d) Weekends

Fig. 4. Evaluating Traffic Trend Inference on Beijing (Default Seed Ratio = 15%).



(a) Workdays (b) Weekends (c) Workdays (d) Weekends

Fig. 5. Evaluating Traffic Speed Estimation on Beijing (Seed Selection Strategies, Default Ratio = 15%).



(a) Workdays (b) Weekends (c) Workdays (d) Weekends

Fig. 6. Evaluating Traffic Speed Estimation on Beijing (Inference vs Estimation, Default Ratio = 15%).

so as to improve the inference quality. (2) The quality of CovGreedy and HybridGreedy was determined by the seeds ratio. If the seeds ratio was small, e.g. 3% or 6%, coverage was more important and CovGreedy performed better than HybridGreedy, because (i) CovGreedy covered more roads as it maximized the coverage of correlated roads; (ii) although HybridGreedy had better accuracy for traffic inference, it might lose the coverage and thus led to a larger MAPE. However, when we selected more seeds, e.g., 12% or 15%, HybridGreedy outperformed CovGreedy because they already covered many roads, and the support became important and HybridGreedy had higher support than CovGreedy. (3) MAPE became smaller when the seeds ratio increases, because more seeds were selected to do inference and improved the coverage.

Figure 5(c) and Figure 5(d) showed the results by the varying time at a day. By linking the MAPE to the accuracy in Figure 4, we find: (1) The larger the accuracy was, the smaller the MAPE was; because a larger accuracy can improve the quality to estimate the speed (i.e., smaller MAPE). For example, in Figure 5(c) and 5(d), when we varied the time of a day from 9am to 9pm, MAPE and the accuracy (in Figure 4(c) & 4(d)) had the opposite trend. Second, the coverage was more important than the support for MAPE, that explains why CovGreedy and HybridGreedy still outperformed other methods as they had a larger coverage.

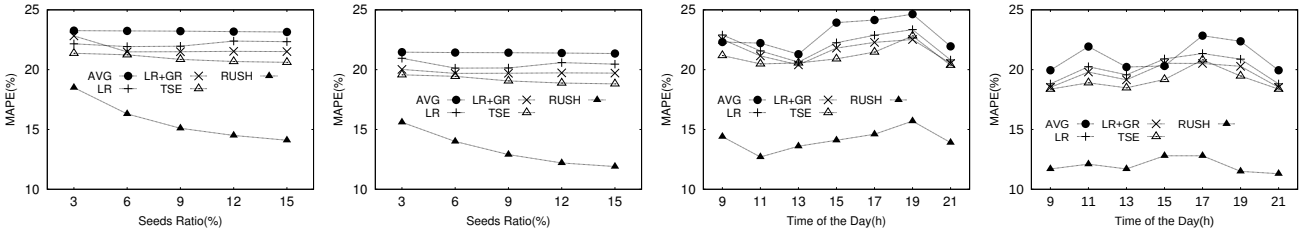
Second, we evaluated the impact of the choice of α on the coverage, traffic trend estimation accuracy and the MAPE of speed estimation respectively. Since HybridGreedy had the best MAPE, we chose it as the default seed selection strategy, and Table III showed its MAPE w.r.t. a varying α. In the combined function (Equation 16), the larger α was, the more important the support was; the smaller α was, the more important the coverage was. Thus for a small seeds ratio, the smaller α, the better, because the coverage played a significant role; for a large seeds ratio, we needed to increase α to balance the support and coverage. Our method achieved the best overall performance at $\alpha = 1$. Thus we set it as the default value.

TABLE III. Varying α for HybridGreedy (MAPE).

Seed Ratio/α	0.01	0.1	1	10
3%	**16.8%**	16.9%	18.5%	18.9%
6%	**15.8%**	16.0%	16.3%	18.2%
9%	15.3%	15.2%	**15.1%**	17.6%
12%	15.1%	15.0%	**14.4%**	17.1%
15%	15.1%	14.9%	**14.1%**	16.4%

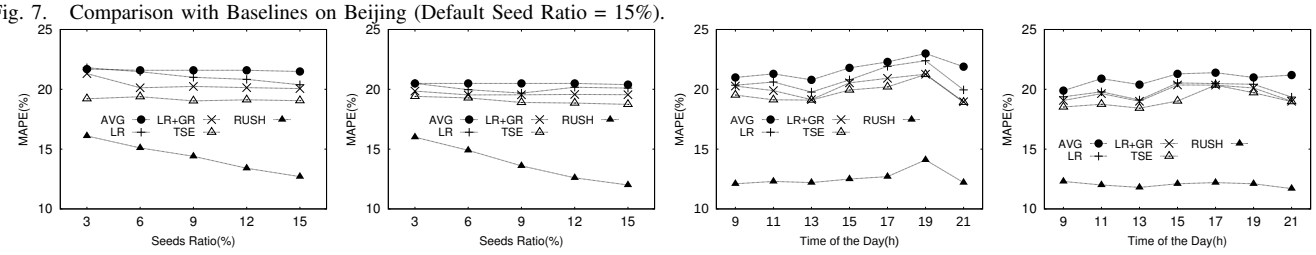
Last, we evaluated our estimation methods. In particular, we compared three methods: (1) our traffic speed estimation method based on trend inference (Inf-Est), (2) a non-estimation method (Non-Est), which utilized the average speed to estimate the real speed (i.e., $\hat{v} = \bar{v}$); (3) an estimation method that only used our hierarchical linear model to learn weights of $\delta = v - \bar{v}$ and estimated $\hat{v} = \bar{v} + \hat{\delta}$ (proposed in Section IV-B), without using the traffic trend inference model, denoted by Est. We used HybridGreedy to select seeds and evaluated the performance of the three methods. As shown in Figure 6, we find: (1) both Inf-Est and Est outperformed Non-Est as they utilized the correlated roads to improve the estimation quality. (2) Inf-Est was better than Est, as the correlated roads only had similar traffic trends but had no similar speeds. This also confirmed that our two-step framework worked well for the speed estimation problem. Furthermore, accurate trend inference can improve the MAPE, because it helped find a correct direction to reduce the estimation error. (3) With an increased seeds ratio, MAPE decreased as we can utilize more seeds to achieve a more accurate inference. (4) In term of the overall MAPE, Inf-Est achieved the best performance, because it utilized both the trend inference model and the speed estimation model to estimate the speed.

Summary. For traffic speed inference, both the traffic trend



(a) Workdays (b) Weekends (c) Workdays (d) Weekends

Fig. 7. Comparison with Baselines on Beijing (Default Seed Ratio = 15%).



(a) Workdays (b) Weekends (c) Workdays (d) Weekends

Fig. 8. Comparison with Baselines on Nanjing (Default Seed Ratio = 15%).

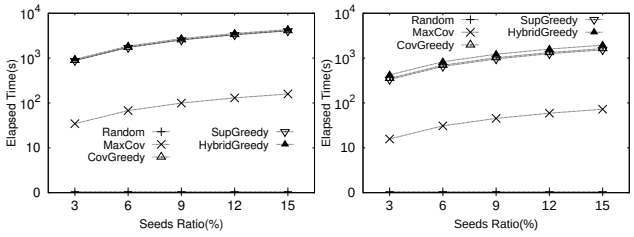
estimation model and the speed learning model were important to estimate the speed, and our two-step model was very effective. For seed selection, the support can improve the traffic trend estimation accuracy while the coverage can improve the speed estimation quality (MAPE). For a small seeds ratio, COVGREEDY and HYBRIDGREEDY outperformed the other methods; for a large seeds ratio, HYBRIDGREEDY achieved the best performance. Thus, we recommended HYBRIDGREEDY for seed selection by using an appropriate parameter α: for a small seeds ratio, we set a small α, e.g., 0.01; for a large seeds ratio, we set a large α, e.g., 1.

C. Comparison with Baselines

We compared our method (RUSH: Realtime Urban Traffic Speed Estimation with Historical Data) with the four baselines. We adopted HYBRIDGREEDY for seed selection. All the methods used the same seeds and historical data.

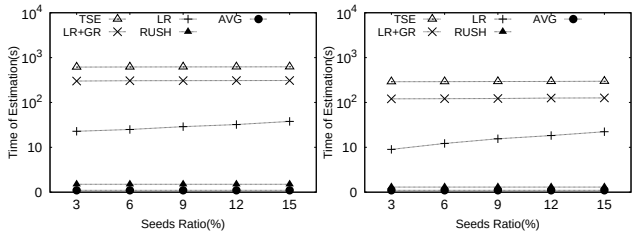
1) Comparison on Quality – MAPE: Figure 7(a) and 7(b) showed the results on Beijing Data by varying the seeds ratio. We had two observations. (1) RUSH achieved the best performance for any seeds ratio and outperformed baselines by 8%-10%, because RUSH used the observation that correlated roads had similar traffic trends but others assumed that correlated roads had similar speeds which was not true in real traffic. (2) With an increased seeds ratio, RUSH kept reducing the MAPE while the the MAPE of the rest almost remained unchanged. This was because AVG did not use the seeds; LR, LR+GR and TSE utilized a strict assumption that the correlated roads had similar speeds and their estimated speeds were rather similar to the average speed.

Figure 7(c) and 7(d) showed the results by varying the time at a day with a seeds ratio of 15%. As we can see, RUSH outperformed the baselines at any time. The MAPE of RUSH was about 10%-15% while those of the baselines were 20%-25%. For example, RUSH had average MAPE of 14.1%, while the MAPE of AVG, LR, LR+GR and TSE were 22.7%, 21.9%, 21.1% and 20.2% resp. This was because our two-step model can better model the real traffic and utilized the traffic trend to improve the speed estimation quality. In contrast, existing methods did not utilize the traffic trend. In particular, AVG utilized the average speed and cannot utilize the seed information; for LR and LR+GR, the historical information was used as a main feature to learn the traffic speed but they cannot capture the realtime traffic from the seeds; for TSE, it learned the matrix based on the historical information to infer the current traffic speeds of unknown roads but cannot utilize the traffic trend. The performance of all five methods were related to the average speed, e.g., when AVG reached the largest MAPE at rush hour 7pm, LR, LR+GR and TSE reach their largest MAPE as well. We had similar findings on the Nanjing data, as shown in Figure 8.



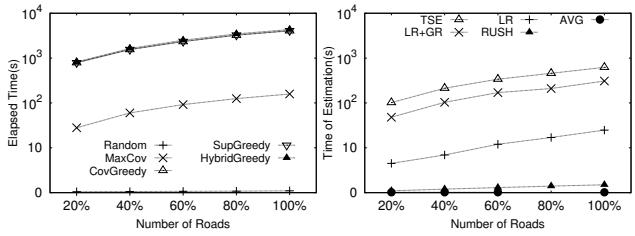
(a) Beijing (b) Nanjing

Fig. 9. Elapsed Time of Seed Selections.



(a) Beijing (b) Nanjing

Fig. 10. Time of Traffic Speed Estimation.



(a) Seed Selection (b) Traffic Speed Estimation

Fig. 11. Scalability on Beijing

2) Comparision on Efficiency & Scalability: First, we tested the elapsed time of seed selection by varying the seeds

ratio. The results were shown in Figure 9. We find: (1) CovGreedy, SupGreedy and HybridGreedy cost more time than Random and MaxCov, because they greedily picked seeds to optimize coverage and support. (2) The elapsed time increased linearly w.r.t. the seeds ratio as we needed to pick more roads into the seed set. (3) The time costs of seed selection on Beijing data was larger than Nanjing data, because Beijing had a larger-scale road network and it took more time for each greedy selection.

Next, we reported the average traffic estimation time (including both trend inference and speed estimation for Rush) on the test data by varying the seeds ratio in Figure 10. We observed that Rush and AVG were rather efficient, e.g., within 1 second, so our method Rush can meet the real-time requirement for online traffic speed estimation. However LR, LR+GR, and TSE took rather long time. For example on Beijing data, LR took nearly 25 seconds, while LR+GR and TSE took more than 300 seconds. This was because they were online learning algorithms which utilized the observed data as training data. AVG was efficient as it simply retrieved the historical data. With an increased seeds ratio, LR needed more time spent on training the model; in contrast, LR+GR and TSE were modeled based on the road network and the number of seeds had weak influence on the training time.

Lastly, we evaluated the scalability of our method. We set the seeds ratio as 15% and varied the number of roads from 20% to 100% by expanding the area of Beijing. In Figure 11(a), we can see that the elapsed time of seed selection increased w.r.t. the number of roads, as it took more time for each greedy selection on larger road networks. In Figure 11(b), the estimation time also increased with the expanding of road networks, because for LR, LR+GR and TSE more training data were involved to learn the model, and for Rush it needed to estimate more roads.

VII. Conclusion

In this paper, we studied the crowdsourcing-based urban traffic speed estimation problem. Inspired by an observation on real traffic data that, correlated roads usually had similar traffic trends, we proposed a two-step model to estimate the traffic speed: (1) we first utilized a graphical model to infer the traffic trend and then (2) adopted a probabilistic model to learn the traffic speed based on the traffic trend. We formulated the seed selection problem, proved its NP-hardness and proposed several greedy algorithms with approximation guarantees. Experiment results showed that our method significantly outperformed baselines in both accuracy and efficiency.

VIII. Acknowledgement

This work was supported by the National Grand Fundamental Research 973 Program of China (2015CB358700), and the National Natural Science Foundation of China (61272090, 61422205, 61472198), Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology, "NExT Research Center", Singapore (WBS:R-252-300-001-490), Huawei, Shenzhou, FDCT/116/2013/A3, MYRG105(Y1-L3)-FST13-GZ, National 863 Program of China (2012AA012600), Chinese Special Project of Science and Technology (2013zx01039-002-002), and the National Center for International Joint Research on E-Business Information Processing (2013B01035).

References

[1] *Traffic Detector Handbook: Third Edition*. U.S. Department of Transportation.

[2] A. Artikis, M. Weidlich, et al. Heterogeneous stream processing and crowdsourcing for urban traffic management. In *EDBT*, pages 712–723, 2014.

[3] M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, pages 624–638, 2004.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[5] L. Chen and C. L. P. Chen. Ensemble learning approach for freeway short-term traffic flow prediction. In *SoSE*, pages 1–6, 2007.

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1989.

[7] X. Fei, C.-C. Lu, and K. Liu. A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. *Transportation Research Part C: Emerging Technologies*, pages 1306–1318, 2011.

[8] J. Gui, R. Hu, Z. Zhao, and W. Jia. Semi-supervised learning with local and global consistency. *J. Comput. Math.*, 91(11):2389–2402, 2014.

[9] J. C. Herrera and A. M. Bayen. Traffic flow reconstruction using mobile sensors and loop detector data. *University of California Transportation Center*, 2007.

[10] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *IEEE Conference on Intelligent Transportation Systems*, pages 929–936, 2010.

[11] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[12] K. Liu, Y. Li, F. He, J. Xu, and Z. Ding. Effective map-matching on the most simplified road network. In *SIGSPATIAL*, pages 609–612, 2012.

[13] W. Min and L. Wynter. Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*, 19(4):606–616, 2011.

[14] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *KDD*, pages 1027–1036, 2014.

[15] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, and X. Zhou. Crowdplanner: A crowd-based route recommendation system. In *ICDE*, pages 1144–1155. IEEE, 2014.

[16] E. Vlahogianni, M. Karlaftis, and J. Golias. Short-term traffic forecasting: Where we are and where weâĂŹre going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014.

[17] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *KDD*, pages 25–34, 2014.

[18] Z.-W. WANG and Z.-X. HUANG. An analysis and discussion on short-term traffic flow forecasting [j]. *Systems Engineering*, 6:019, 2003.

[19] B. Williams, P. Durvasula, and D. Brown. Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record: Journal of Transportation Research Board*, pages 132–141, 1998.

[20] B. Yang, C. Guo, and C. S. Jensen. Travel cost inference from sparse, spatio-temporally correlated time series using markov models. *VLDB*, 6(9):769–780, 2013.

[21] B. Yang, C. Guo, C. S. Jensen, M. Kaul, and S. Shang. Multi-cost optimal route planning under time-varying uncertainty. In *ICDE*, 2014.

[22] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *KDD*, pages 1306–1315, 2014.

[23] R. Zhong, G. Li, K. Tan, L. Zhou, and Z. Gong. G-tree: An efficient and scalable index for spatial search on road networks. *IEEE Trans. Knowl. Data Eng.*, 27(8):2175–2189, 2015.

[24] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang. A compressive sensing approach to urban traffic estimation with probe vehicles. *IEEE Trans. Mob. Comput.*, 12(11):2289–2302, 2013.

[25] H.-X. Zou, Y. Yang, Q.-Q. Li, and A. G.-O. Yeh. Traffic data interpolation method of road link based on kriging interpolation. *Journal of Traffic and Transportation Engineering*, 11(3):118–126, 2011.