TransInfo
Tier I University Transportation Center

FINAL REPORT

City-Wide Transportation Information Estimation with Heterogeneous Urban Data

Date of report: March, 2019

Jing Gao, PhD, Associate Professor, Department of Computer Science and Engineering

Prepared by:
Department of Computer Science and Engineering, University at Buffalo
338 Davis Hall
Buffalo, New York 14260-2500

Prepared for:
Transportation Informatics Tier I University Transportation Center
204 Ketter Hall
University at Buffalo
Buffalo, NY 14260

1. Report No.	**2. Government Accession No.**	**3. Recipient’s Catalog No.**

4. Title and Subtitle City-Wide Transportation Information Estimation with Heterogeneous Urban Data	**5. Report Date** March 2019
	6. Performing Organization Code
7. Author(s) Jing Gao	**8. Performing Organization Report No.**
9. Performing Organization Name and Address Department of Computer Science and Engineering, University at Buffalo 338 Davis Hall Buffalo, New York 14260-2500	**10. Work Unit No. (TRAIS**
	11. Contract or Grant No. DTRT13-G-UTC48
12. Sponsoring Agency Name and Address US Department of Transportation Office of the UTC Program, RDT-30 1200 New Jersey Ave., SE Washington, DC 20590	**13. Type of Report and Period Covered** September 2017 – December 2018
	14. Sponsoring Agency Code
15. Supplementary Notes	

16. Abstract

In the same way that Big Data has transformed numerous industries by providing insights into many aspects, it has also been offering travelers and transportation authorities chances to understand transportation performance and issues better. Recently, the analysis of transportation data has attracted much attention, and many analytics methods have been proposed to extract insights from transportation data. The success of these methods is witnessed on capturing the performance on the major arterial roads because these roads are heavily equipped with sensors, traversed by many travelers, and the volume of information associated with these roads is high. However, in a city, most local roads do not have fixed sensors, and are not covered by enough floating sensors carried by travelers neither. The scarcity of sensory data brings great challenges in the accurate estimation of transportation information on these roads. As these roads occupy most areas in a city, it would be difficult to understand the city-wide transportation conditions with all these missing links. To tackle the challenging problem of estimating transportation information for data-scarce roads, this project develops novel data mining methodologies that integrate heterogeneous urban data for the estimation of city-wide transportation information. Tasks include the inference of traffic speed, volume and emission, which are all critical components that contribute to the understanding of the overall transportation conditions in a city. The project also aims to detect traffic anomalies and derive a confidence measurement together with each estimate.

17. Key Words Transportation Information Estimation, Heterogeneous Urban Data	**18. Distribution Statement** No restrictions. This document is available from the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 19	**22. Price**

City-Wide Transportation Information Estimation with Heterogeneous Urban Data

Acknowledgements

Include collaborators here

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Table of Contents

1 INTRODUCTION

During the rapid urbanization process, cities are facing numerous challenges in meeting the needs of their growing populations, especially in the transportation system. Fortunately, urban computing methods [28] can help us tackle these issues by using the large amount of data generated in cities. Obviously, being aware of the city-wide traffic volume, i.e. the number of vehicles traversing on every road segment, is the first step to solve the urban transportation problems. It can bring great convenience to the residents by helping them avoid congested roads. In addition, the traffic volume information is also very important for the government in city planning, such as detecting bottle-neck roads, intervening on congestions, and estimating vehicles' pollution emissions.

Existing volume monitoring methods use loop detectors, surveil- lance cameras or taxi trajectory data, but they all fail to provide a city-wide volume estimation. Specifically, loop detectors (certain sensors buried under the pavements) and surveillance cameras are expensive and thus difficult to be scaled up to cover the whole city area. As for the taxi trajectories, their occurrence on the streets is only a biased sample of the entire traffic volume, because their operating time and behaviors are quite different from the general population.

In this work, we propose to infer the city-wide volume information with both **loop detector** and **taxi trajectory** data. By making the best use of both data sets, we can achieve a better volume estimation on a city scale. However, this is a nontrivial problem faced with great challenges. Firstly, although loop detectors can provide accurate volume information, the data collected is very sparse and incomplete because very few roads are affordable to be equipped with loop detectors. In order to infer the volume information for those road segments without loop detectors, we can first construct an affinity graph to characterize the similarities among roads, then propagate the knowledge of the road segments with loop detector to other segments. However, constructing the affinity graph is also a difficult task, because we do not possess the knowledge of road similarities with respect to volume values. To solve this problem, we first infer the city-wide travel speed from the taxi trajectory data, and learn the daily speed patterns for each road segment. Based on our observations in the real world data, the speed patterns co- evolve with volume patterns, and nearby roads with similar speed patterns are likely to follow similar volume patterns. This allows us to learn the similarity among road segments with respect to the speed pattern, then utilize it to form the spatial affinity graph. In addition to the spatial domains, we also build connections among recent and periodical time slots in the affinity graph. With the learned affinity graph, we design a graph-based semi-supervised learning method which captures the spatio-temporal properties of traffic volume, thus achieving a city-wide volume estimation. Furthermore, we propose practical confidence measurements with spatio-temporal smoothness property and graph-structure property. These measurements will provide end users a better interpretation on the inference results.

In summary, we propose a novel framework to infer the city-wide volume information with both loop detector and taxi trajectory data. The contributions are summarized as follows.

- We propose a spatio-temporal semi-supervised learning model to tackle the data sparsity problem associated with loop detectors and taxi trajectories. Daily speed patterns are learned from taxi trajectories, and it is utilized to construct the affinity graph among roads. As a result, the knowledge from two different domains are fused in this framework to infer a city-wide volume information.
- We propose practical confidence estimation measurements with the spatio-temporal smoothness property and graph-structure property.
- We conduct extensive experiments on real-world data which was collected with 155 loop detectors and 6,918 taxis over a period of 17 days. The results demonstrate the advantages of the proposed framework on correctly inferring the city-wide traffic volume.

The rest of the paper is organized as follows. We describe the overview of the framework in Section 2. Components of the frame- work are detailed in Section 3, and experiments are shown in Section 4. We review the related work in Section 5 then conclude the paper in Section 6.

2 OVERVIEW

In this section, we first introduce several preliminary concepts that will be used throughout the work, formally define the problem that we aim to solve, then describe the proposed framework in Figure 1.

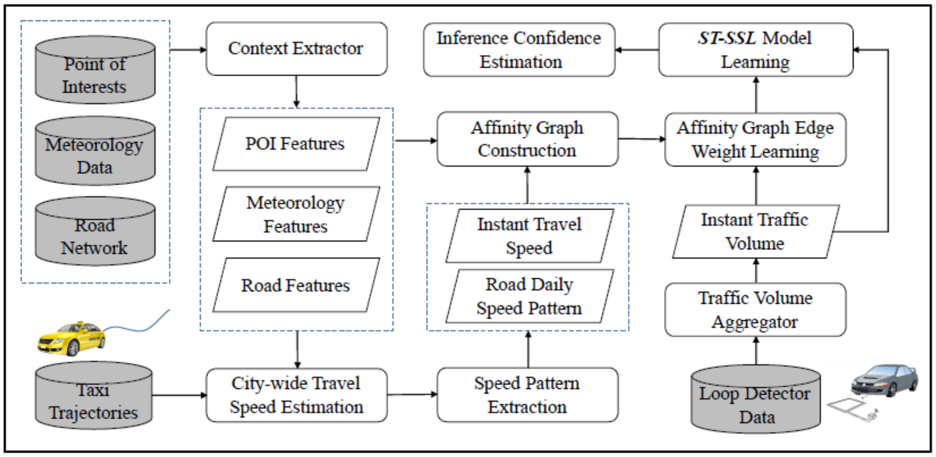


Figure 1: Proposed Framework

2.1 Preliminary

Definition 2.1. Traffic Volume of a road segment is the total number of vehicles traversing through it during a fixed time window.

The total traffic volume of each road is determined by two factors, the number of lanes and the average volume on each lane. In this paper, we aim to infer the per-lane volume for road segments, then the corresponding total traffic volume can be easily derived by multiplying with the number of lanes.

Definition 2.2. Loop Detectors, or "inductive-loop traffic detectors", are sensors buried under the pavements. They can detect vehicles passing or arriving at a certain point, thus the traffic volume information can be derived by aggregating the loop detector's readings over a certain time window.

Note that, although loop detectors can generate accurate volume information, they are so expensive that the government can implement only on a small number of road segments. For example, only 155 road segments are equipped with loop detectors, while there are totally 19165 road segments in the urban area of Guiyang. This is far from sufficient in terms of acquiring the city-wide volume information.

Definition 2.3. Trajectory (Tr) of a taxicab is a sequence of time- ordered spatial points, Tr : p1 → p2 → · · · → pn where each point p has a geo-spatial coordination set l and a time stamp t , i.e., p = (l, t).

Definition 2.4. Point of Interest (POI) is a specific location that someone may find useful. In this work, they represent venues in the physical world, e.g., banks and shopping malls. Each POI is associated with properties such as name, address, coordinates, category and etc.

Definition 2.5. Road Network is a graph-based data structure where road segments are connected to each other. Each road segment is represented as a directed edge with two terminal points.

2.2 Problem Formulation

Given the loop detector data collected from a small portion of roads, the taxi trajectory data, and urban context data sets (i.e., point of interests, road network, and weather conditions), the **Objective** is to estimate the traffic volume of any road within the city area at any time slot.

2.3 Framework Overview

Figure 1 presents the proposed framework. The major components are: city-wide travel speed estimation, volume affinity graph construction, affinity graph edge weight learning, Spatio-Temporal Semi-Supervised Learning (ST-SSL) and inference confidence estimation. The details of each component will be elaborated in the Section 3.

Briefly speaking, we first map the GPS trajectories onto a road network with map matching algorithm [14], and infer the average travel speed on all road segments with travel speed estimation method [21]. Then daily speed patterns of each road are learned, and the volume affinity graph can be constructed with those pat- terns together with features extracted from POI and road network. Once we construct the affinity graph and learn the edge weights, the proposed spatio-temporal semi-supervised learning algorithm is performed to infer the city-wide traffic volume for each road segment. Finally, in order to provide a better interpretation on the inference results, we provide confidence estimation measurements with spatio-temporal smoothness and graph-structure properties.

In the following section, we will detail every component of the proposed framework and demonstrate the reasons behind our design.

3 METHODOLOGY

3.1 City-wide Travel Speed Estimation

The taxi trajectories by themselves cannot generate the city-wide traffic volume information directly. Although their occurrences on the road surface can be recognized as a sample of the total volume, it is only a very small and biased sample. The reason is obvious, their operating time and behaviors are quite different from the general population. Fortunately, we can derive a city-wide travel speed on each road segment from the trajectories with existing methods, and the travel speed is a good factor that correlates with the volume.

In our framework, GPS coordinates are mapped onto the road network with map matching method [14] and the average travel speed of roads covered by the trajectory can be calculated. However, the taxis can only cover limited road segments for any given time period. In order to estimate the travel speed on all road segments, we perform a city-wide travel speed estimation method [21] with the context based collaborative filtering. By this means, we can derive an accurate estimation of the travel speed on each road segment of the urban area.

3.2 Urban Context and Speed Pattern Extraction

Many urban context information has correlations or impacts on the traffic volume, thus can help us with the inference task. For example, POIs reveal the functionalities of an area, and obviously a shopping mall would attract more visits than other places. Similarly, properties of road networks and weather condition also affect the traffic volume.

With the collected data, we can extract three sets of urban features: geographic features of road segments, speed pattern features, and weather features.

Geographic features of a road segment. For each road segment, we can extract features from the road network and the nearby POI data.

- Road network features: Road length, level, direction, number of lanes, speed constraint, number of connections and tortuosity., the ratio between road length and Euclidean distance between the two terminals).

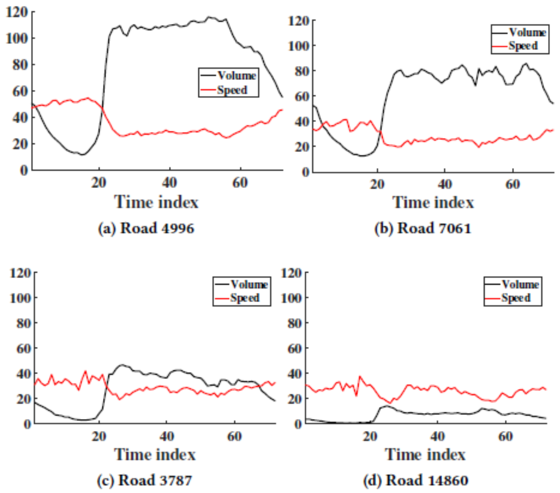
- POI features: For any given road segment, within a distance of 200 meters around its two terminals, we can calculate the distribution of POIs across ten categories. These categories are: Schools, Companies and Offices, Banks and ATMs, Malls and Shopping, Restaurants, Gas stations and Vehicle, Scenic spot, Hotels and Residences, Transportations, Entertainments and Living.

Speed pattern features. For each road segment, we can derive two features from its speed pattern:

- Daily speed pattern. Given the historical record of travel speed on each road segment, we can derive a daily speed pattern by averaging it by day.

- Daily average speed. Calculated by further average over the daily speed pattern.

The speed pattern features play important roles when inferring the traffic volume. Although we have many urban features extracted, unfortunately, none of them has direct relationship with the volume value of a road segment. However, the speed patterns usually co- evolve with the volume patterns, and this means that roads with similar average speed and speed patterns are more likely to follow similar volume patterns. Several typical speed patterns are shown in Figure 2 together with their volume patterns. As a result, the speed patterns can help us determine the similarities among roads with respect to the volume. More specifically, the Pearson coefficient between any two series of daily speed patterns can measure the similarity between these two roads under this context.



(a) Road 4996

(b) Road 7061

(c) Road 3787

(d) Road 14860

Figure 2: Typical Speed and Volume Patterns

Weather features. For any given time slot, we can collect the weather features from monitoring stations, such as temperature, humidity, weather conditions (categorized as cloudy, foggy, rainy, sunny and snowy) and etc. Among these, the weather conditions have the most direct impact on the traffic volume. One obvious example is that volume patterns would be similar on days with extreme weather conditions, such as heavy snow or heavy rain.

3.3 Volume Affinity Graph Construction

The proposed ST-SSL model is a graph-based semi-supervised learning model. Its success depends on not only the model itself, but also the corresponding affinity graph. We first define the affinity graph in the following.

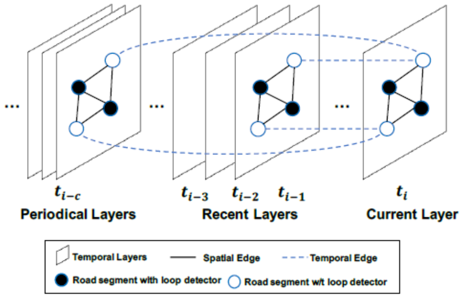


Figure 3: Affinity Graph Structure

Definition 3.1. Affinity Graph (Figure 3) is a multi-layer weighted graph where each layer denotes a time slot. Each node in the graph is the lane volume of a road segment , i.e. average volume per lane, on one specific time slot. There are edges connecting among the nodes, and their weights represent the correlations among the nodes. Edges within the same time slot are “spatial” correlations and those between different time slots are “temporal” correlations. The node set V = L ∪ U , where L is the set of nodes whose volume are known (or “Labelled”), in other words, those road segments with loop detectors. U is the set of nodes whose volume are not known (or “Unlabellded”).

Spatial and temporal knowledge play important roles in the construction of the affinity graph. By connecting nodes with deliberately designed spatio-temporal rules, we encode those prior knowledge into the affinity graph.

Spatial Affinity Graph. Within each time slot, the construction of the spatial affinity graph consists of the following parts:

- Edge between nearby road segments. Common sense tells us that road segments in nearby areas are likely to have similar traffic conditions. Based on this observation, for each road segment, we select its nearby roads (e.g., within 300 meters) as candidates. However, a road segment may have too many nearby neighbors. As shown in

Figure 4, there are quite a few roads with more than 50 neighbors. Then it is not reasonable to assume that all these nearby roads have similar lane volume. In order to deal with this problem, we further filter the candidates with road segments' daily speed patterns. Specifically, we filter several (e.g., 20) road segments from the candidate set with the most similar daily average speed, then further filter a small set (e.g., 10) with the similar daily speed patterns. By this means, we connect road segments that share the similar traffic patterns.

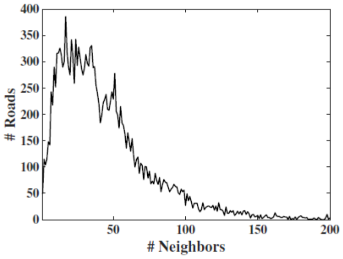


Figure 4: Number of Neighbors Histogram

- Edge between reachable road segments. Some road segments are relatively long, and the previous rule may only extract too few neighbors. However, road segments within several hops (e.g., 2 or 3 hops) are likely to share similar volume. Based on this idea, we extract nodes within several hops as candidates, then filter the candidate set by their speed patterns as described above.

- Edge to similar road segments with loop detectors. The loop detectors provide very reliable volume readings. In order to make the best use of these data, we build edges between a road segment and a few others that equip loop detectors with similar traffic patterns. This is also achieved by filtering the daily speed patterns.

Temporal Affinity Graph. For nodes on different temporal layers, the construction of edges consists of the following parts:

- Edge between recent layers. Obviously, the volume values would not change dramatically in a relatively short time period. The idea here is to build edges between nodes in nearby temporal neighbors, for example, within 30 minutes.
- Edge between periodical temporal layers. For each road segment, the volume follows a regular pattern every day, every week, and even every season. Figure 5 shows the volume value of a road during a 10-day interval. The daily and weekly periodical pattern can be easily depicted. Based on this observation, we connect road segments with all of their periodical neighbors.
- Edge between layers with extreme weather conditions. The traffic volume would follow similar patterns when encountered with extreme weather conditions, such as heavy snow and heavy rain. As a result, we build edges among temporal layers under similar extreme weather conditions.

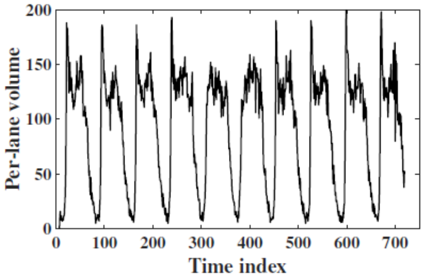


Figure 5: Periodical Volume Pattern

Graph Edge Weight Learning. After constructing the affinity graph, we should learn the edge weights in the graph which rep- resents the volume similarity among nodes. The basic idea is that two nodes should have similar lane volume patterns if they share similar urban features, and vice versa. However, urban features may have different correlations with the volume patterns which can be learned from the data. Specifically, the spatial edge weights are calculated in the following process:

- Step 1: Normalize the feature values fi for all features i. This is achieved by feature standardization which makes the values of each feature in the data have zero-mean and unit-variance, i.e.,

$f_i := \frac{f_i - mean(f_i)}{std(f_i)}$

- Step 2: For each feature i, calculate the Pearson correlation corrfi between the feature value differences |Δ fi | and lane volume value differences |Δx |. Specifically, for any two roads u and v with loop detector readings on the same temporal layer, the lane volume value difference is |Δxu,v | = |xu − xv |, and the feature value differ- ence is |Δ fi (u, v)| = | fi (u) − fi (v)|. Then with all the loop detector data, we can calculate the Pearson correlation denoted by corrfi . In other words, corrfi measures the impact of different features on the volume values.

- Step 3: Calculate spatial edge weight au,v between any two roads u and v as follows:

$$a_{u,v} = \exp\left(-\sum_i corrf_i \times |f_i(u) - f_i(v)|\right)$$

In addition, it is easy to be noticed that nodes in temporal neighbors should have similar lane volume. Thus, we set the temporal edge weights as 1.

3.4 Spatio-Temporal Semi-Supervised Learning

Based on the constructed volume affinity graph and the learned edge weights, we propose a graph-based semi-supervised learning model(ST-SSL) which incorporates the knowledge of spatio-temporal volume patterns. Generally speaking, there are several advantages of the proposed method. First, the ST-SSL model can make good use of the observed volume information by loop detectors, and infer all missing volume values in the graph. Second, ST-SSL assures that nodes with similar feature values should have similar lane volume values, which is achieved by learning the edge weights. Third, ST- SSL follows the spatio-temporal prior knowledge which makes it behave under physical constraints.

The first rule that we take into consideration is the change rate similarity of spatial neighbors. Assume that there are totally T temporal layers in the graph, and xu(t) denotes the lane volume of road u on time slot t . The corresponding loss function is

$$\mathcal{L}_s = \sum_{t\in T}\sum_{(u,v)\in E} a_{u,v}\left(\frac{\triangle x_u(t)}{x_u(t-1)} - \frac{\triangle x_v(t)}{x_v(t-1)}\right)^2,$$

where Δx (t) = x (t)−x (t −1). This rule captures the fact that volume of the correlated roads usually has similar change rates over time. For example, during the morning rush hours, many vehicles would take similar routes to work and it makes the volume around working areas increase at the same time.

The second rule is that volume on consecutive and periodical time slots should be similar. Let corr(t) represents the set of correlated temporal layers with time t, and it contains recent and periodical layers. Then the corresponding loss function is

$$\mathcal{L}_t = \sum_{t\in T}\sum_{c\in corr(t)}\sum_{u\in U}(x_u(t) - x_u(c))^2$$

By putting the equations (2) and (3) together, we have the loss function of the proposed ST-SSL model

$$\mathcal{L}_{STSSL} = \mathcal{L}_s + \mathcal{L}_t = \sum_{t\in T}\sum_{(u,v)\in E} a_{u,v}\left(\frac{\triangle x_u(t)}{x_u(t-1)} - \frac{\triangle x_v(t)}{x_v(t-1)}\right)^2 + \alpha\sum_{t\in T}\sum_{c\in corr(t)}\sum_{u\in U}(x_u(t) - x_u(c))^2$$

where α is a hyper-parameter which gives different emphases on the two terms. Then our goal is to find the lane volume for all roads X = {xu |u ∈ V}, such that the loss function (4) is minimized

$$X = \arg\min_X \mathcal{L}_{STSSL}$$

In order to solve the above problem, we adopt an iterative update algorithm. By taking the partial derivative of $\mathcal{L}_{STSSL}$ to xv (t) and let it be zero, we get

$$x_v(t) = \frac{\sum_{u \in N(v)} \frac{a_{u,v} x_u(t)}{x_u(t) x_v(t-1)} + \alpha \sum_{c \in corr(t)} x_v(t+c)}{\sum_{u \in N(v)} \frac{a_{u,v}}{x_v^2(t-1)} + \alpha C}$$

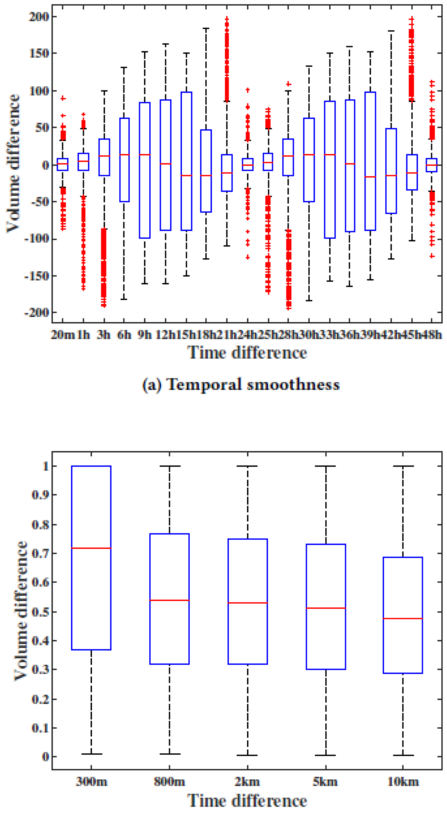
where N (v) represents the set of nodes that are connected with v, and C = |corr(t)| is the number of correlated temporal layers with time t.

By carefully examining the equation (6), we can observe that it successfully captures the spatio-temporal properties of the traffic volume. Specifically, the volume of one road segment is correlated with its spatial neighbors because they have similar volume change rates, as shown by the first term of the nominator. In addition, it also correlates with its temporal neighbors because it changes smoothly over time, as shown by the second term of the nominator. Overall, the inference of traffic volume is a weighted average over spatial and temporal components.

3.5 Inference Confidence Estimation

The inference confidence is a vital problem that haunts many learning methods, especially for semi-supervised algorithms. Tradition- ally, for the purpose of evaluation, we can divide the labelled data into training and evaluation sets, but this would further reduce the amount of data for inference. In this section, we propose to use the property of spatio-temporal smoothness and graph structure to determine the confidence of the inference results. Because of their generality, these measurements can also be applied to other spatio-temporal and/or graph-based models.

Spatio-temporal Smoothness Confidence. In many real world ap- plications, the target variables would not change dramatically over both temporal and spatial dimensions. For example, the temperature usually changes gradually over time and geo-space. With respect to the traffic volume, Figure 6 shows that it tends to have similar values in recent time slot and locations. Specifically, Figure 6(a) presents the volume differences as time goes on. It is obvious that the volume values on recent and periodical time slots are more similar; Figure 6(b) demonstrates the volume ratio with different spatial distances. This shows that nearby roads should have similar lane volume values.



(a) Temporal smoothness

(b) Spatial smoothness

Figure 6: Spatio-temporal smoothness of volume

With the above observations, we define a confidence metric on the spatio-temporal smoothness as follows:

Definition 3.2. Spatio-temporal Smoothness Confidence (CST). For any given node v in the affinity graph G, we assume the value of its correlated nodes follows a normal distribution N (μ, σ 2), where μ and σ can be estimated accordingly. Then the spatio-temporal smoothness confidence of the node v is defined as

$$C_{ST} = \begin{cases} P(x < x_v \cap x > 2\mu - x_v) = 2CDF(x_v), & x_v < \mu \\ P(x > x_v \cap x < x_v - 2\mu) = 2(1 - CDF(x_v)), & x_v < \mu \end{cases}$$

Graph Structure Confidence. In graph-based semi-supervised learning methods, the knowledge of labelled data is propagated through the affinity graph. However, some unlabelled nodes might be connected to labelled nodes directly or with larger edge weights, and some others may connect to labelled nodes remotely or with smaller edge weights. Following this path, we define a graph structure based confidence measurement. The basic idea is that nodes connected to labelled data with higher weights should receive more confidence, and vice versa.

Definition 3.3. Graph structure confidence (CG). For any given node v in the affinity graph G, we calculate the edge weights be- tween v and its connected nodes with volume labels. Then we take the maximum weight as the graph structure confidence.

Moreover, a typical method of aggregating these two measurements is using F-measure as follows

$$F_\beta = (1 + \beta^2) \cdot \frac{C_{ST} \cdot C_G}{\beta^2 \cdot C_{ST} + C_G}$$

where we can assign different weights on these measurements with parameter β.

The proposed framework is evaluated with a real-world data and its practical usage is demonstrated. In the following, we introduce the data sets, the baselines, and the performance measurements for the volume inference. Then the results are shown and discussed.

4 EXPERIMENT

4.1 Data Set

All the data is collected in Guiyang, a south-west city of China. There are totally 155 road segments deployed with loop detectors, and Figure 7(a) shows their locations in the road network. These loop detectors generate timely readings about vehicles passing by, and we aggregate them into traffic volume values for every 20 minutes. We also collect real-time GPS readings of 6,918 taxicabs. Figure 7(b) shows the heatmap of taxi appearances on the road where the frequency decreases from color red to green. Both the loop detector data and taxi trajectories are collected during the period of March 16 to April 1, 2016.

We also collected road network and POI data in Guiyang, and there are totally 19,165 road segments in the road network. The corresponding features are extracted as described in the Section 3.2.

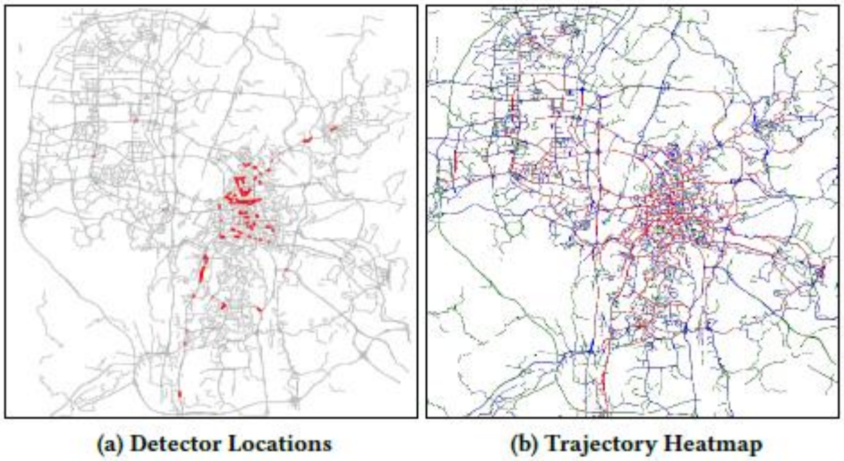


Figure 7: Loop Detectors and Taxi Trajectories

4.2 Baselines

In this experiment, we compared the proposed ST-SSL algorithm with several supervised and semi-supervised methods.

- Linear regression (LR). A linear regression model is trained with all the extracted features.

- Artificial neural network (ANN). We train an multi-layer perceptron using back propagation with no activation function in the output layer.

- Random forest regression (RF). A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Basic semi-supervised learning model [30] (Basic-SSL). This is a classical semi-supervised learning model, and its loss function is $\sum_{u,v} a_{u,v} (x_u - x_v)^2$. Compared with the proposed model, this baseline does not consider the spatio-temporal patterns.

4.3 Performance Measurements

Because volume values varies a lot on different roads, the absolute values of Mean Square Error and Mean Absolute Error are not reasonable measurements for the inference performance. This is because the final errors will be biased towards roads with higher volume values, and this is unfair for many roads that only have relatively low volume. In order to avoid the aforementioned bias, we use the following metrics to judge the performance of each method:

- MSPE: Root Mean Square Percentage Errors. For each road, the percentage volume error is defined as its error divided by the ground truth value. This measures how much errors are made compared with the ground truths. Then for all the roads in evaluation data set, we calculate the root of the mean squared relative errors as the MSPE.

- VarMSPE : Variance of the MSPE. For each method, we ran the experiment several times, and the variance of their MSPE represents the robust of the corresponding method.

- MAPE: Mean Absolute Percentage Errors. Similarly, it measures the overall absolute relative errors between each method's outputs and the ground truths. This is calculated by averaging the absolute difference of relative errors.

- VarMAPE : Variance of the MAPE. For each method, we ran the experiment several times, and the variance of their MAPE represents the robust of the corresponding method.

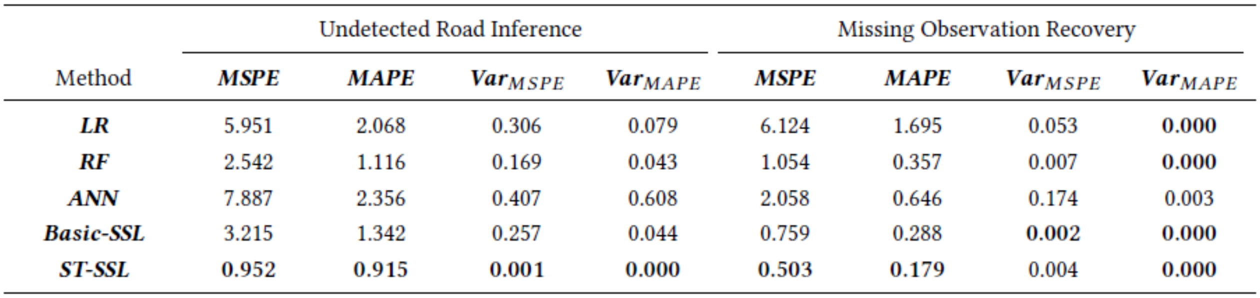
4.4 Evaluation Results

In the following, we perform extensive experiments on the proposed method, and demonstrate its advantages with the results.

Inference Accuracy. To compare the performance of all the methods on the volume inference accuracy, we conduct experiments on the urban data set in Guiyang. In this experiment, 80% of all the volume data collected from loop detectors are treated as training data, and the rest is used for testing.

In real-world applications, loop detectors may malfunction occasionally and those readings will be discarded. This results in missing values at some time slots, and poses more challenge on the volume inference task. For the rest of this section, we refer "undetected road inference" as the problem of inferring the volume of roads without loop detectors, and "missing observation recovery" as the problem of inferring the missing volume of roads with loop detectors. In order to test the performance on both of the two volume inference tasks, we simulate the scenarios as follows. In the former task, we randomly select the volume records from all roads with 80% percentage as the training data; while for the later task, we randomly select 80% roads and treat their readings as the training data, and test on the remaining 20% roads. The results are shown in Table 1. It clearly shows the better performance of ST-SSL in both tasks compared with all the baselines. As can be expected, the inference accuracy on "missing observation recovery" is quite lower than that on task "undetected road inference". Because it is easier to infer the volume value of a road if we have some historical observations, and vice versa. In addition, the proposed method has great robust since the variance of MSPE and MAPE are always near to 0.

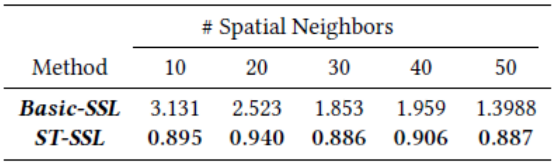
Table 1: Volume Inference Accuracy



	Undetected Road Inference				Missing Observation Recovery			
Method	***MSPE***	***MAPE***	***Var***$_{MSPE}$	***Var***$_{MAPE}$	***MSPE***	***MAPE***	***Var***$_{MSPE}$	***Var***$_{MAPE}$
LR	5.951	2.068	0.306	0.079	6.124	1.695	0.053	**0.000**
RF	2.542	1.116	0.169	0.043	1.054	0.357	0.007	**0.000**
ANN	7.887	2.356	0.407	0.608	2.058	0.646	0.174	0.003
Basic-SSL	3.215	1.342	0.257	0.044	0.759	0.288	**0.002**	**0.000**
ST-SSL	**0.952**	**0.915**	**0.001**	**0.000**	**0.503**	**0.179**	0.004	**0.000**

Vary the number of spatial neighbors The performance of graph-based semi-supervised models is affected by the graph structure. In this experiment, we vary the number of spatial neighbors of each node in the affinity graph, and see how it affects the volume inference. From Table 2, we can see that ST-SSL performs similarly with different number of neighbors. The reason is two-folded. On one hand, the edge weights of our affinity graph are learned intelligently. This means that even though we build many edges among the nodes, they can distinguish the importance of the correlations wisely with the learned edge weights. On the other hand, the pro- posed ST-SSL method incorporates spatio-temporal patterns in the model, and this helps to achieve a good volume inference. In comparison, although Basic-SSL performs worse than the proposed method, the performance of Basic-SSL gets better when the graph has more spatial neighbors in the affinity graph. This proves that the proposed Spatio-temporal semi-supervised learning requires less information passed through the graph, because it genuinely incorporates those prior knowledge.

Table 2: Performance Varying # Spatial Neighbors



	# Spatial Neighbors				
Method	10	20	30	40	50
Basic-SSL	3.131	2.523	1.853	1.959	1.3988
ST-SSL	**0.895**	**0.940**	**0.886**	**0.906**	**0.887**

Vary the spatio-temporal factor α. In equation (4), hyper- parameter α is a factor that gives different weights on the spatial and temporal terms. More emphasis will be put onto the temporal term with a larger α , and vice versa. In fact, this factor is application dependent because the two terms in equation (4) are not normalized. In order to evaluate the effect of α on the inference errors, we perform experiments with different α and the results are shown in the Figure 8. As depicted from the figure, the inference error can reach reasonably low when the α is within a certain range, i.e., around 1 in this case. However, both larger or smaller values have negative effects on the inference results.

Table 3: Performance Varying Training:Testing

	Training:Testing			
Method	0.5:0.5	0.6:0.4	0.7:0.3	0.8:0.2
LR	5.308	5.673	4.314	4.280
RF	2.840	2.790	2.139	1.639
ANN	6.067	5.206	3.819	4.059
Basic-SSL	3.370	3.955	3.519	3.374
ST-SSL	**0.943**	**1.220**	**0.940**	**0.935**

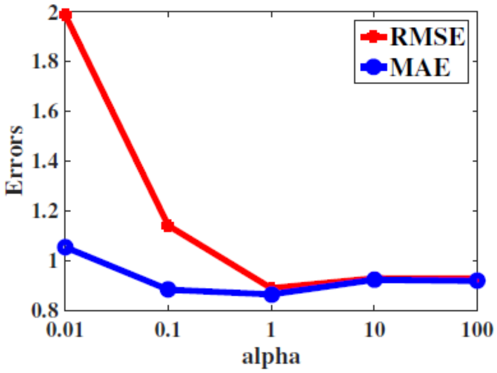


Figure 8: Vary Spatio-temporal Factor α

Vary the data size. We also conduct experiment with different amount of data. The data size is varied from 6 hours to 15 days, and the inference accuracies of all methods are aggregated in Table 4. Obviously, the propose method ST-SSL performs consistently well over baselines under all the settings. This means that no matter how much data we have, ST-SSL can give quite accurate estimations on the volume information. On the contrary, supervised learning based models cannot provide meaningful estimations. This is be- cause the urban features only correlate with the traffic volume in a very obscure way, such that baselines cannot learn meaningful relationships between those features and the traffic volume.

Table 4: Performance (MSPE) w.r.t. Varying Data Size

	Undetected Road Inference							
Method	6h	12h	18h	1d	3d	5d	10d	15d
LR	1.850	3.155	3.451	4.280	5.558	5.322	5.951	4.775
RF	2.086	2.971	2.850	1.639	2.672	2.072	2.542	1.875
ANN	2.527	3.898	3.734	4.059	5.497	3.998	7.887	4.313
Basic-SSL	2.112	3.131	3.204	3.374	3.737	3.117	3.215	2.687
ST-SSL	**0.917**	**0.895**	**0.980**	**0.935**	**0.948**	**0.969**	**0.952**	**0.955**
	Missing Observation Recovery							
	6h	12h	18h	1d	3d	5d	10d	15d
LR	2.735	2.880	3.062	4.738	4.888	4.536	5.124	4.720
RF	0.820	1.549	1.419	1.363	1.275	1.085	1.054	0.944
ANN	3.999	2.455	2.775	3.180	2.579	1.982	2.058	1.802
Basic-SSL	0.825	0.922	0.912	1.032	0.950	0.788	0.759	0.658
ST-SSL	**0.382**	**0.412**	**0.508**	**0.639**	**0.510**	**0.507**	**0.503**	**0.483**

Vary the ratio of training and testing data. The results are shown in Table 3. Similar with what we have discussed, ST-SSL performs quite well even with very few collected volume data. In addition, although supervised methods perform better when given more training data, they still cannot catch up with the proposed method.

Visualization of Inference Confidence In Figure 9, we display the confidence on a map for the end users. Each mark represents the confidence level of a road segment. Levels of confidence are represented with different colors (green represents the most confident and red represents the least confident). In addition, users can zoom in and out to see the details.

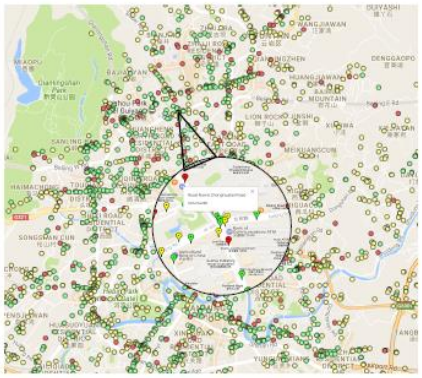


Figure 9: Visualization of Inference Confidence

Convergence. The proposed method converges quickly after only a few iterations which is shown in Figure 10. This demonstrates the efficiency of the proposed method.

5 RELATED WORK

5.1 Traffic Condition Inference

Loop detectors [11, 15, 18, 22] and surveillance cameras [20, 23] are traditional equipments in traffic condition monitoring. On the bright side, the volume of a road can be directly extracted from these devices. They are mainly used to predict the future volume of a road segment with filtering-based models [18, 22] or time series models [13]. However, the data sparsity problem is critical and it is hard to infer the volume of a city-wide area, because they are too expensive to deploy across the whole city. Fundamental diagram [4–6, 8, 15, 25] characterizes the relationship among traffic density, speed and volume. But this approach requires great amount of training data to learn the relationships, which is unrealistic for our problem. Moreover, there is a unique relationship for each road, such that it makes the city-wide estimation infeasible. In recent researches, more attentions have been put onto the traffic condition estimation with floating car data [9, 10, 19, 21, 24, 26, 27, 32], where vehicles are equipped with GPS sensors and collect geo-spatial coordinates on the move. After mapping the trajectories onto the road network, it is easy to derive the average travel speed [21, 26] on some of the road segments. However, infer the traffic volume with floating car data is nontrivial, because the cars are only a small and biased sample of the total traffic volume (e.g. taxi trajectories). In [19], the authors first estimate the travel speed on each road, then adopt an unsupervised Bayesian model to learn the volume. However, it is hard to learn a universal model for all the road segments with handcrafted Bayesian network. [1] explored the possibility of learning a regression model with floating cars' occurrence, but it still cannot solve the sparsity problem. In addition, this work also proved that the floating cars are biased sample because the taxi distribution usually under-estimates the volume on rush hours and over-estimates otherwise. Compared with the existing work, the proposed framework in this paper handles both the floating car data and loop detector data wisely, and can derive city-wide volume information with more accuracy.

5.2 Semi-supervised Learning

Supervised learning models use "labelled" data to build a model, then apply on the unlabelled data for inference. However, acquiring enough labelled data in real-world applications is quite difficult, sometimes impossible. Semi-supervised learning (SSL) models [31] addresses this problem by exploiting unlabelled data to help with the learning performance. There have been extensive research on this field, and many models are proposed. Generative models [16, 17] usually treat the class labels of the unlabelled data as missing values and employ the EM algorithm to conduct maximum likelihood estimation; Low-density separation models [7] use a maximum margin algorithm, such as support vector machines, to push the decision boundary away from the unlabelled points. Graph- based models are learned to find the minimum cut of the graph for the classification tasks [2, 3], and to propagate the knowledge across the graph for the continuous prediction tasks [12, 29, 30]. In this paper, we adopt the graph-based semi-supervised model which is suitable for inferring the continuous values, and can propagate the volume knowledge collected by loop detector across the affinity graph.

6 CONCLUSIONS

In this paper, we propose a framework to infer the city-wide traffic volume information with loop detectors and taxi trajectories. However, both of these data are incomplete, sparse, and from quite different domains. In order to solve these problems, we propose a spatio-temporal semi-supervised learning model to tackle the data sparsity problem associated with loop detectors. In addition, the knowledge from trajectories is also integrated by learning the traffic patterns and utilize it to construct the affinity graph. The experiment was conducted on real-world data and the results demonstrate advantages of the proposed framework on inferring the city-wide traffic volume.

7 ACKNOWLEDGEMENTS

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. The work was sup- ported by the National Natural Science Foundation of China (Grant No. 61672399 and No. U1609217), China National Basic Research Program (973 Program, No. 2015CB352400), US National Science Foundation (CNS-1553411, and CNS-1737590), and Transportation Informatics (TransInfo) University Transportation Center at University at Buffalo. Any opinions, findings, and conclusions or rec- ommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Javed Aslam, Sejoon Lim, Xinghao Pan, and Daniela Rus. 2012. City-scale traffic estimation from a roving sensor network. In Proc. of SenSys. ACM, 141–154.

[2] Avrim Blum and Shuchi Chawla. 2001. Learning from labeled and unlabeled data using graph mincuts. (2001).

[3] Avrim Blum, John Lafferty, Mugizi Robert Rwebangira, and Rajashekar Reddy.

2004. Semi-supervised learning using randomized mincuts. In Proc. of ICML. ACM, 13.

[4] BD Greenshields, Ws Channing, Hh Miller, et al. 1935. A study of traffic capacity. In Highway research board proceedings, Vol. 1935. National Research Council (USA), Highway Research Board.

[5] Astrid Gühnemann, Ralf-Peter Schäfer, Kai-Uwe Thiessenhusen, and Peter Wagner. 2004. Monitoring traffic and emissions by floating car data. (2004).

[6] Dirk Helbing. 2001. Traffic and related self-driven many-particle systems. Reviews of modern physics 73, 4 (2001), 1067.

[7] Thorsten Joachims. 1999. Transductive inference for text classification using support vector machines. In ICML, Vol. 99. 200–209.

[8] Boris S Kerner. 2004. Three-phase traffic theory and highway capacity. Physica A: Statistical Mechanics and its Applications 333 (2004), 379–440.

[9] Benjamin Krogh, Ove Andersen, Edwin Lewis-Kelham, Nikos Pelekis, Yannis Theodoridis, and Kristian Torp. 2013. Trajectory based traffic analysis. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 536–539.

[10] Benjamin Krogh, Ove Andersen, and Kristian Torp. 2012. Trajectories for novel and detailed traffic information. In Proceedings of the Third ACM SIGSPATIAL International Workshop on GeoStreaming. ACM, 32–39.

[11] Jaimyoung Kwon, Pravin Varaiya, and Alexander Skabardonis. 2003. Estimation of truck traffic volume from single loop detectors with lane-to-lane speed correlation. Transportation Research Record: Journal of the Transportation Research Board 1856 (2003), 106–117.

[12] Neil D Lawrence and Michael I Jordan. 2004. Semi-supervised learning via Gaussian processes.. In NIPS, Vol. 17. 753–760.

[13] Sangsoo Lee and Daniel Fambro. 1999. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. Transportation Research Record: Journal of the Transportation Research Board 1678 (1999), 179–188.

[14] Yin Lou, Chengyang Zhang, Yu Zheng, Xing Xie, Wei Wang, and Yan Huang. 2009. Map-matching for low-sampling-rate GPS trajectories. In Proc. of SIGSPATIAL. ACM, 352–361.

[15] Laura Muñoz, Xiaotian Sun, Roberto Horowitz, and Luis Alvarez. 2003. Traffic density estimation with the cell transmission model. In American Control Conference, 2003. Proceedings of the 2003, Vol. 5. IEEE, 3750–3755.

[16] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. Ma- chine learning 39, 2-3 (2000), 103–134.

[17] Kamal Paul Nigam. 2001. Using unlabeled data to improve text classification. Ph.D. Dissertation. Massachusetts Institute of Technology.

[18] Iwao Okutani and Yorgos J Stephanedes. 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research Part B: Method- ological 18, 1 (1984), 1–11.

[19] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. 2014. Inferring gas consumption and pollution emission of vehicles throughout a city. In Proc. of SIGKDD. ACM, 1027–1036.

[20] Gautam S Thakurzx, Pan Huiz, and Ahmed Helmyx. 2012. Modeling and characterization of urban vehicular mobility using web cameras. In Computer Com- munications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on. IEEE, 262–267.

[21] Yilun Wang, Yu Zheng, and Yexiang Xue. 2014. Travel time estimation of a path using sparse trajectories. In Proc. of SIGKDD. ACM, 25–34.

[22] David Wilkie, Jason Sewall, and Ming Lin. 2013. Flow reconstruction for data- driven traffic animation. ACM Transactions on Graphics (TOG) 32, 4 (2013), 89.

[23] Xianyuan Zhan, Ruimin Li, and Satish V Ukkusuri. 2015. Lane-based real-time queue length estimation using license plate recognition data. Transportation Research Part C: Emerging Technologies 57 (2015), 85–102.

[24] Xianyuan Zhan, Satish V Ukkusuri, and Chao Yang. 2016. A Bayesian mixture model for short-term average link travel time estimation using large-scale limited information trip-based data. Automation in Construction 72 (2016), 237–246.

[25] Xianyuan Zhan, Yu Zheng, Xiuwen Yi, and Satish V Ukkusuri. 2017. Citywide Traffic Volume Estimation Using Trajectory Data. IEEE Transactions on Knowledge and Data Engineering 29, 2 (2017), 272–285.

[26] Fangfang Zheng and Henk Van Zuylen. 2013. Urban link travel time estimation based on sparse probe vehicle data. Transportation Research Part C: Emerging Technologies 31 (2013), 145–157.

[27] Yu Zheng. 2015. Trajectory data mining: an overview. ACM Transactions on Intelligent Systems and Technology (TIST) 6, 3 (2015), 29.

[28] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology (TIST) 5, 3 (2014), 38.

[29] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bern- hard Schölkopf. 2003. Learning with local and global consistency. In NIPS, Vol. 16. 321–328.

[30] Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In ICML, Vol. 3. 912–919.

[31] Xiaojin Zhu and Andrew B Goldberg. 2009. Introduction to semi-supervised learning. Synthesis lectures on artificial intelligence and machine learning 3, 1 (2009), 1–130.

[32] Yanmin Zhu, Zhi Li, Hongzi Zhu, Minglu Li, and Qian Zhang. 2013. A com- pressive sensing approach to urban traffic estimation with probe vehicles. IEEE Transactions on Mobile Computing 12, 11 (2013), 2289–2302.