
Proceedings Paper:
Wang, P., Kim, Y., Vaci, L. et al. (2 more authors) (2018) Short-term traffic prediction with vicinity Gaussian process in the presence of missing data. In: 2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF). 12th Symposium Sensor Data Fusion: Trends, Solutions, and Applications, 09-11 Oct 2018, Bonn, Germany. IEEE . ISBN 978-1-5386-9398-8

https://doi.org/10.1109/SDF.2018.8547118

Short-Term Traffic Prediction with Vicinity Gaussian Process in the Presence of Missing Data

Peng Wang, Youngjoo Kim, Lubos Vaci, Haoze Yang, and Lyudmila Mihaylova
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, United Kingdom
{Peng.Wang, Youngjoo.Kim, Lubos.Vaci, hyang48, l.s.mihaylova}@Sheffield.ac.uk

***Abstract*— This paper considers the problem of short-term traffic flow prediction in the context of missing data and other measurement errors. These can be caused by many factors due to the complexity of the large scale city road network, such as sensors not being operational and communication failures. The proposed method called vicinity Gaussian Processes provides a flexible framework for dealing with missing data and prediction in vehicular traffic network. First, a weighted directed graph of the network is built up. Next, a dissimilarity matrix is derived that accounts for the selection of training subsets. A suitable cost function to find the best subsets is also defined. Experimental results show that with appropriately selected subsets, the prediction root mean square error of the traffic flow obtained by the vicinity Gaussian Processes method reaches 18.9% average improvement with lower costs, which is with comparison to inappropriately chosen training subsets.**

I. INTRODUCTION

"Smart City" is now quite a popular concept that aims at making the city smarter from different perspectives with the minimum changes to the existing infrastructures. Intelligent transportation systems (ITS) play a key role in building smart cities. One of the critical elements for the successful deployment of ITS lies in traffic prediction [1], [2], especially when it comes to large traffic networks with a limited number of sensors as shown in Fig. 1.

Basically, traffic prediction methods can be divided into model-based and data-driven methods [3], with the criterion of whether models or data are exploited to accomplish the prediction. In the model-based group, a physical traffic model is explicitly defined to describe the dynamics of the traffic road system. In the 1970s, Ahmed et al. [4] propose the autoregressive integrated moving average (ARIMA) to cope with short-term highway traffic prediction. Model-based methods have been extensively researched thereafter. Up to now, there are mainly three categories of models, i.e., microscopic, macroscopic and mesoscopic. Microscopic models provide high level details of each individual vehicle [5], [6], which are intuitively both time and resource consuming. Worse still, details of an individual vehicle are not always available. Macroscopic models, on the other hand, represent the aggregated behaviour of the traffic, usually in terms of average speed and density. They are the direct compromise between computational efficiency and prediction accuracy. Macroscopic models are suitable for real-time traffic prediction and management. Most model-based methods fall into the macroscopic category, such as the model the cell transmission model (CTM) [7] and the interval CTM [8]. Mesoscopic models are hybrids of microscopic and macroscopic models with the emphasis in varying levels of details [9]. Under the assumption that models can describe the traffic system dynamics, the results are highly reliable and therefore competitive. Although a number of models have been proposed, there is still no general model for all traffic scenarios. This limits the application of model-based methods.

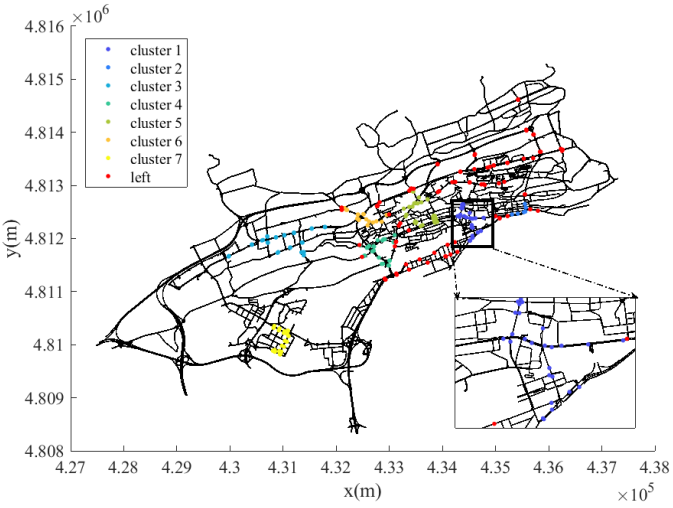

Fig. 1. Road network and sensors of Santander, Spain

Compared with model-based methods with the explicit requirement to a physical model, the data-driven methods only demand historical data. Statistical and machine learning methods are developed for finding the inherent dependencies in data and then based on them future events are predicted. Ni et al. [10] propose a Bayesian network based method, having the advantages of reducing the bias and accuracy in traffic prediction. Recently, deep learning methods have been proposed for traffic prediction. After the successful application [1] of a deep stacked autoencoder (SAE) approach to traffic

prediction, a lot of researchers have focused on deep learning methods for traffic prediction. Related publications are [11], [12]. Nevertheless, deep learning methods still suffer from computational complexity during the model training phase. Also, these methods heavily rely on the data preparation or pre-processing procedures, which influence the real-time application.

The Gaussian process (GP) method [13] is another data driven solution with a big potential in the traffic prediction, a kernel-based learning algorithm just like SVMs [14]. GP have been repeatedly demonstrated to be a powerful tool in implicit relationship exploring and difficult non-linear regression addressing, with applications in mobility demand and short-term traffic volume prediction. Comparative studies have shown that GP outperform ARIMA, SVM and neural networks on short-term traffic prediction [15], [16]. However, GP still suffer from cubic time complexity in the size of training data. Fortunately, both parallel/distributed computation [15] and non-negative matrix factorization (NMF) techniques [17] provide the possibilities to decrease the computational complexity.

In this paper, GP models are trained from vicinity sensor data and then we employ it to do traffic prediction for data missing segments. To start, the road network is divided into shorter segments (loop detectors are mounted in a certain number of segments). With the direction information of the roads in hand, a weighted directed graph (wDG) and the corresponding dissimilarity matrix are consequentially constructed. The dissimilarity matrix serves as the heuristic information to choose the measurements from neighbouring sensors to get the training data ready. GP models trained by the selected data generally report the best prediction when the local sensor malfunctions or the communication fails. As we only use the vicinity sensor measurements, the GP model hereafter is abbreviated as v-GP in case of confusion.

The remainder of the paper is organized as follows. In section II, GP and its application in the traffic prediction is formulated. The methodologies to build the wDG and dissimilarity matrix are detailed in section III. Section IV provides the implementation of v-GP. In Section V, experimental results are provided with analyses. Section VI concludes the paper.

II. PROBLEM DESCRIPTION

A. The Gaussian Process Framework

A GP is generally regarded as an extension of a multivariate Gaussian distribution in an infinite dimensional space, with any finite number of which subjects to a joint Gaussian distribution. Normally, the real process $f(\mathbf{x})$ is not available. Fortunately, one of the powerful aspects of GP lies in using the Bayesian paradigm to learn an approximation of $f(\mathbf{x})$ from the training data.

The GP prior is fully defined by the mean $m(\mathbf{x})$ and the covariance matrix $k(\mathbf{x}, \mathbf{x}^{'})$ as in (1),

$$p(f(\mathbf{x})|\theta) \propto \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^{'})), \tag{1}$$

in which, $\mathbf{x}$ is the input vector, $m(\mathbf{x}) = \mathbb{E}(f(\mathbf{x}))$, $k(\mathbf{x}, \mathbf{x}^{'}) = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}^{'}) - m(\mathbf{x}^{'}))]$, θ is the prior's hyperparameter vector, $\mathcal{N}(\cdot)$ denotes a Gaussian distribution and $\mathbb{E}(\cdot)$ is the mathematical expectation operator. The mean $m(\mathbf{x})$ is usually assumed to be 0, and $k(\mathbf{x}, \mathbf{x}^{'})$ is the kernel function. One of the widely-used kernel function is the squared exponential covariance function

$$\sigma_{\mathbf{x}\mathbf{x}'} \triangleq \sigma_{\mathbf{x}}^2 \exp\left(-\frac{1}{2}\sum\nolimits_{i=1}^{p}\left(\frac{[x_{\mathbf{x}}]_i - [x_{\mathbf{x}'}]_i}{\ell_i}\right)^2\right) + \sigma_n^2 \delta_{\mathbf{x}\mathbf{x}'}, \tag{2}$$

where $[x_{\mathbf{x}}]_i$ and $[x_{\mathbf{x}'}]_i$ are the i-th components of the corresponding inputs, $[\sigma_{\mathbf{x}}^2, \sigma_n^2, \ell_i, \cdots, \ell_p] \triangleq \theta$ are hyperparameter defined as noise and input variances, and length-scales that can be learned by the maximum likelihood estimation, and $\delta_{\mathbf{x}\mathbf{x}'}$ is a Kronecker delta that equals to 1 if $\mathbf{x} = \mathbf{x}^{'}$ and 0 otherwise.

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a training data set, with $\mathbf{x}_i \in \mathbb{R}^d$ the d-dimensional input and y_i the corresponding one dimensional measurement at $\mathbf{x}_i$, which can be temporal, spatial or hybrids of the both. With inputs $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_N^T]$, we can get the corresponding outputs through $f(\mathbf{x})$ as $\mathbf{f}(\mathbf{X}) = [f(\mathbf{x}_1), f(\mathbf{x}_2), \cdots, f(\mathbf{x}_N)]^T$. Normally, $y_i \neq f(\mathbf{x}_i)$ stands because of noise, which can be assumed to be drawn from a Gaussian distribution determined by the likelihood $p(\mathbf{y}|\mathbf{f})$ between the outputs and the measurements. The posterior can then be obtained by updating the prior according to Bayesian theorem

$$p(\mathbf{f}|\mathcal{D}, \theta) = \frac{p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{X}, \theta)}{p(\mathcal{D}|\theta)}, \tag{3}$$

in which, $\mathbf{y} = [y_1, y_2, \cdots, y_N]^T$ is the measurement vector.

Now given any new input $\mathbf{x}_*$ and the posterior (3), then the corresponded output is constrained by the predictive distribution

$$p(f_*|\mathbf{x}_*, \mathcal{D}, \theta) = \int p(f_*, \mathbf{f}|\mathcal{D}, \theta)\, d\mathbf{f}. \tag{4}$$

For comprehension, denote the joint distribution of measurements from the training data set and the function output at $\mathbf{x}_*$ under the prior as

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) + \sigma_N^2 \mathbf{I} & K(\mathbf{X}, \mathbf{x}_*) \\ K(\mathbf{x}_*, \mathbf{X}) & K(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right), \tag{5}$$

where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ can be directly computed from (2), and $\sigma_N^2 \mathbf{I}$ are noise covariances. Then equation (4) can be rewritten as

$$p(f_*|\mathbf{x}_*, \mathcal{D}, \theta) \sim \mathcal{N}(f_*, \mathrm{cov}(f_*)), \tag{6}$$

in which,

$$\begin{aligned} f_* \triangleq \mu_{\mathbf{x}_*|\mathbf{X}} &\triangleq \mathbb{E}(f_*|\mathbf{x}_*, \mathcal{D}, \theta) \\ &= K(\mathbf{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_N^2 \mathbf{I}]^{-1}\mathbf{y} \\ &= \mathbf{\Sigma}_{\mathbf{x}_*\mathbf{X}} \mathbf{\Sigma}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{y}, \end{aligned} \tag{7}$$

$$\begin{aligned} \mathrm{cov}(f_*) \triangleq &\mathbf{\Sigma}_{\mathbf{x}_*\mathbf{x}_*|\mathbf{X}} = K(\mathbf{x}_*, \mathbf{x}_*) - \\ &K(\mathbf{x}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \mathbf{\Sigma}_N^2 \mathbf{I}]^{-1} K(\mathbf{X}, \mathbf{x}_*) \\ = &\mathbf{\Sigma}_{\mathbf{x}_*\mathbf{X}} \mathbf{\Sigma}_{\mathbf{X}\mathbf{X}}^{-1} \mathbf{\Sigma}_{\mathbf{X}\mathbf{x}_*}. \end{aligned} \tag{8}$$

B. Traffic prediction with GP

In the traffic prediction scenario, sensors are installed in certain road segments to record counts and speeds of the vehicles passing by. Let V_s be the set of road segments with sensor installed, with each segment related to input $\mathbf{x}$. The observation equation can be represented in the general form

$$y = f(\mathbf{x}, \epsilon). \tag{9}$$

As stated before, the problem of missing data, caused by sensor or communication failures, is one of the most frequent phenomena in the traffic prediction. Hereafter, we call the segments without data received as local segment, and denote them as $S \subseteq V_s$. When data are missing, e.g. due to nonoperational sensors, the most intuitive solution is to use the historical data of S to do prediction. However in such cases, one GP model has to be trained for each malfunctioning sensor, which is normally time consuming and computational resource expensive. Another solution is integrating sensor data from vicinity segments $D_v \subseteq V_s$. Thus, less models are needed and dependencies among vicinity sensors are considered. Without loss of generality, we use D_v to represent functioning segments in neighbourhood, and the corresponding feature vector and output are denoted as $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \cdots, \mathbf{x}_n^T]^T$ and $\mathbf{y} = [y_1, y_2, \cdots, y_n]^T$, respectively. According to (7) and (8), the problem can be formulated as determining the Gaussian predictive distribution $\mathcal{N}(\boldsymbol{\mu}_{S|D_v}, \boldsymbol{\Sigma}_{SS|D_v})$ with $\boldsymbol{\mu}_{S|D_v}$ and $\boldsymbol{\Sigma}_{SS|D_v}$ given in (10) and (11),

$$\boldsymbol{\mu}_{S|D_v} \triangleq \boldsymbol{\Sigma}_{SD_v}\boldsymbol{\Sigma}_{D_vD_v}^{-1}\mathbf{y}, \tag{10}$$

$$\boldsymbol{\Sigma}_{SS|D_v} \triangleq \boldsymbol{\Sigma}_{SD_v}\boldsymbol{\Sigma}_{D_vD_v}^{-1}\boldsymbol{\Sigma}_{D_vS}. \tag{11}$$

The next section describes the design of the weighted directed graph and dissimilarity matrix which are key elements of the proposed approach.

III. Weighted Directed Graph and Dissimilarity Matrix

A. Weighted Directed Graph

The road network of even a small city can be of high complexity. For this reason, it is sensible to partition the road network into multi-scale segments according to the junctions, or even with the consideration of population or commercial factors. Here, we define the wDG of a city's road network in the same way as in [14]. Given a road network, the corresponding wDG is $G \triangleq (V, E, w)$. V is the vertex set representing all possible road segments. $E = V \times V$ is the edge set, with the constraint that there is an edge between segments v_i and v_j iff the end of v_i is connected to the start of v_j. The edge is denoted as e_{ij}. Be aware that e_{ji} can be totally different to e_{ij} because of the nature of the road networks. The weight of e_{ij} is denoted as w_{ij}, which is defined as the weighted average of the "distance" or dissimilarity between each attribute of v_i and v_j. The segment attributes considered here include: length of the segment, number of lanes, limitation of the highest speed, direction of the segment and the classification of the segment. Let's suppose that the attribute vector is $\mathbf{a} = [a_1, a_2, \cdots, a_5]$, then w_{ij} is computed by

$$w_{ij} = \sum\nolimits_{k=1}^{5} \alpha_k |a_k^i - a_k^j| / r_k, \tag{12}$$

with r_k the range of the k-th attribute, and α_k the weight of the k-th attribute. w_{ij} is the sum of each attribute dissimilarity. Bigger w_{ij} indicates significant difference between v_i and v_j, i.e., inversion in direction, big changes in lanes etc.

With the wDG in hand, we can construct a dissimilarity matrix

$$\mathbf{M} \triangleq \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ m_{N1} & m_{N2} & \cdots & m_{NN} \end{bmatrix}, \tag{13}$$

where N is the total number of segments, and m_{ij} indicates the distance between v_i and v_j, which is computed through

$$m_{ij} = \min\left\{\sum w_{E'} | E' \subseteq E\right\}, \tag{14}$$

in which, E' is a candidate edge set that starts from v_i and ends at v_j.

We call it the dissimilarity matrix because bigger m_{ij} indicates higher costs for a vehicle to transfer from v_i to v_j. Therefore, it can be regarded as the basis to choose more temporally and spatially related road segments.

B. Asymmetrical multidimensional scaling for spatial information

From the way how the dissimilarity matrix $\mathbf{M}$ is constructed, it can be easily observed that it violates symmetrical assumption imposed on covariance of the GP prior. One possible way is to perform the asymmetrical multidimensional scaling (AMDS) to embed the higher dimensional matrix into lower dimensional Euclidean space first, and then compute the GP prior covariance from the lower dimensional matrix.

The core idea of AMDS is to find a lower dimensional matrix $\mathbf{C} \in \mathcal{R}^{N \times p}$, such that (15) is satisfied,

$$\begin{aligned} d(\mathbf{M}_{i,:}, \mathbf{M}_{j,:}) =& \sum\nolimits_{k=1}^{N} (m_{ik} - m_{jk})^2 \\ \approx& \sum\nolimits_{k=1}^{p} (c_{ik} - c_{jk})^2 \\ =& d(\mathbf{C}_{i,:}, \mathbf{C}_{j,:}) \\ s.t., \min& \sum\nolimits_{ij} \big(d(\mathbf{M}_{i,:}, \mathbf{M}_{j,:}) - d(\mathbf{C}_{i,:}, \mathbf{C}_{j,:})\big) \end{aligned} \tag{15}$$

in which, $d(\cdot, \cdot)$ indicates the Euclidean distance between two vectors and $\mathbf{M}_{i,:}(\mathbf{C}_{i,:})$ indicates the i-th row of matrix $\mathbf{M}(\mathbf{C})$.

Now the lower dimensional matrix $\mathbf{C}$ can be substituted into (2) to compute the symmetrical prior covariance if needed. Another benefit of AMDS is that each $\mathbf{C}_{i,:}$ can be regarded as the spatial information of the i-th road segment. Together with the time stamp when the data were collected, we can construct the temporal-spatial inputs for GP.

IV. Implementation of v-GP

Without loss of generality, we suppose $s \in S$ is one of the segment with malfunctioning sensor, and $\mathbf{D} = [D_1, D_2, \cdots, D_Q]$ are functioning sensors spatially near to s. Q can be large or small, which is varying in different cities or even in different areas of the same city. To alleviate computational complexity, we partition $\mathbf{D}$ into subsets with the same length l. Thus $q = \mathrm{C}_Q^l$ subsets $T = \{T_1, T_2, \cdots, T_q\}$ can be generated. Since both s and $\mathbf{D}$ are known, we can easily get a sub-matrix $\mathbf{M}^{'}$ from $\mathbf{M}$ to indicate the distance from s to $\mathbf{D}$. Consequently, distances from s to T can be efficiently inquired from $\mathbf{M}^{'}$ whose size is normally decreased compared to $\mathbf{M}$. Denote the distance from s to T_i as $\mathbf{M}_i$, then subset T_i with the minimum entry-wise distance sum $d_{min}^i = \Sigma_{k=1}^l m_{ik}$ is regarded as the best candidate for building up the training set. For robustness, we choose n sets ξ with d_{min}^{ξ} smaller than a threshold r.

The structure of the training data used in this paper is as follows. Each input contains time stamp, spatial and temporal information. Spatial information is the matrix by embedding the dissimilarity matrix into a lower dimensional space, and is denoted as $c_1, c_2, \cdots, c_p$. Temporal information is constructed by L sensor observations immediately precede the to be predicted observation o_{i+L+1}, and is denoted as $o_{i+1}, o_{i+2}, \cdots, o_{i+L}$. Time stamp t can be directly converted from the exact time when o_{i+L+1} was collected. Observations are the vehicle numbers aggregated in 15 minutes. Index i can be 0 or any number indicating the beginning of the observations to be incorporated, L is the model input length. Now, the training data set can be denoted as $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_t}$, with $\mathbf{x}_i = [t, c_1, \cdots, c_p, o_i, \cdots, o_{i+L}]^T$ and $y_i = o_{i+L+1}$. We only consider the observations from one sensor, i.e. $l_0 \in T$ in the training dataset. This is a reasonable assumption, because the aim of v-GP is to predict in the data missing segments by integrating observations from vicinity sensors. Be aware that the intersection of training data set and testing data set is empty, which is controlled by i. Symbols N_t and N_s denote the size of the training data set and testing data set, respectively.

In summary of the description above, we present a v-GP algorithm is now given by Alg. 1. For real-time applications both l and r can be decreased to shrink the amount of sensors to be considered. On the other hand, lines 4-8 in Alg. 1 can be executed in parallel to accelerate the training speed. Once the training phase is finished, we select the GP models with the minimum root mean square error (RMSE) to do prediction for data missing segments. The RMSE is given by

$$\mathrm{RMSE} = \left(\frac{1}{N_s} \sum_{i=1}^{N_s} (\hat{y}_i - y_i) \right)^{\frac{1}{2}}, \tag{16}$$

in which, $\hat{y}_i$ is the i-th prediction and y_i is the measurement.

Algorithm 1 v-GP algorithm

Input: $\mathbf{M}$, s, $\mathbf{D}$, $\mathbf{C}$
Output: $\mathcal{N}(\mu_{s|[s,\xi_\gamma]}, \Sigma_{ss|[s,\xi_\gamma]})$

1: Initialization
2: generate $\mathbf{M}^{'}$ and $T = \{T_1, T_2, \cdots, T_q\}$
3: $\Xi \triangleq \{\xi \in T, |\Sigma_{k=1}^l m_{ik} \leq r, i = 1, \cdots, q\}$,
4: **for** $\xi_i \in \Xi$ **do**
5: GP training
6: $f_*^i \sim \mathcal{N}(\mu_{s|[s,\xi_i]}, \Sigma_{ss|[s,\xi_i]})$
7: GP testing
8: $\Gamma_i = \mathrm{RMSE}(f_*^i)$
9: $\gamma = \min\Gamma$
10: $f_* \sim \mathcal{N}(\mu_{s|[s,\xi_\gamma]}, \Sigma_{ss|[s,\xi_\gamma]})$

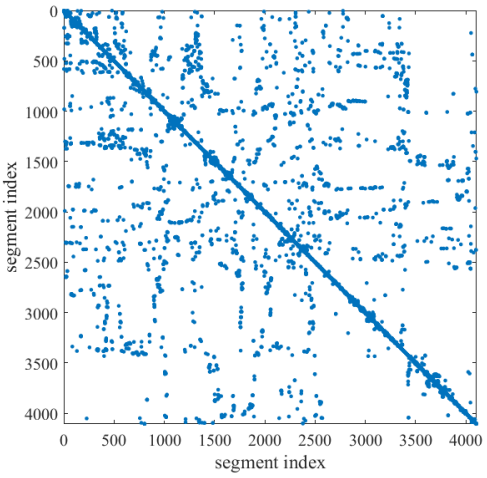

Fig. 2. Adjacency Matrix

V. Experiments and Analyses

A. Experiment settings

In this paper, we consider the road network of Santander city in Spain as shown in Fig. 1. The volume dataset is from the case studies of the EU SETA project [18]. The road network was partitioned into 4106 segments and total number of 296 sensors were installed in some of the segments. The wDG of the road network is described by the adjacency matrix shown in Fig. 2. $I(x, y) = 1$ iff e_{xy} exists, otherwise it is 0. Theoretically, the dissimilarity matrix for the whole directed graph can be computed. However in this paper, we are only interested in the segments with sensors installed. It is worth reminding that some segments are covered by more than one sensor. Here we only consider one sensor on each segment. The dissimilarity matrix is shown in Fig. 3. Please note that the dissimilarity matrix is computed only for segments with the sensors and it is asymmetrical. For better visualization, entries in the matrix are multiplied by 10.

We used the clustering algorithm from [19] to cluster the sensor location first. The results are shown in Fig. 1. Then we randomly selected one of the clusters, denoted as $\mathbf{D} = [D_1, D_2, \cdots, D_Q]$ with $Q = 21$ to build up the vicinity area. To facilitate evaluation, we assume sensor s is malfunctioning. $\mathbf{D}$ is partitioned into subsets $T = \{T_1, T_2, \cdots, T_q\}$ with $q = \mathrm{C}_Q^l = \mathrm{C}_{21}^5$, where $l = 5$ could be any value between 1 and $Q = 21$, the length of

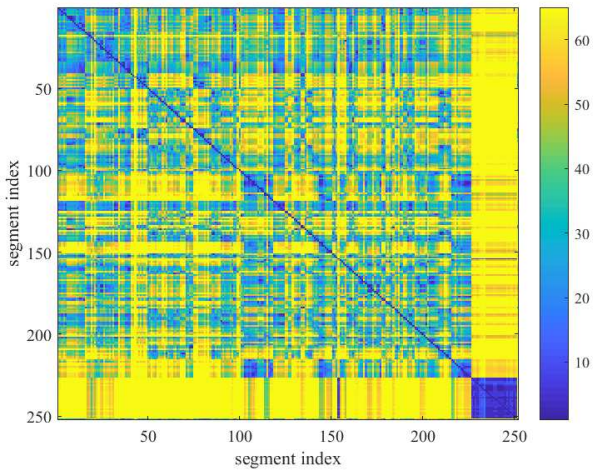

Fig. 3. Dissimilarity Matrix

the subset T_i. The GP models were trained by using the MATLAB Gaussian Process Toolbox, which determines the hyperparameters automatically. We used the kernel function shown in (2).

B. Experiments and Analyses

Two sets of experiments were conducted with $s = \#11$ and $s = \#13$ being as the data missing segments separately, which were again randomly selected. Though we have $q = C_{21-2}^{5-2} = 969$ subsets in total, we have randomly selected only 20 subsets to finish the experiments. This put us at the risk of not selecting the best candidate subsets. We are going to show that given any subsets we are capable to pick the best candidates. When the prediction for a whole city is assembled, we can distribute the task to multiple processors (either in a distributed or parallelized way) and further refine the choice of the globally best subset in the future.

Fig. 4 shows the distances between the local segment $\#13$ and its vicinity segments. The upper sub-figure shows the distance from the local segment to the vicinity segments and the lower sub-figure vice versa. Since we are only interested in the prediction for the local segment, we only consider the distance shown in the lower sub-figure. Fig. 5 shows the RMSEs of the GP models trained from different vicinity sensor data while doing prediction on the testing data set. We can see that when the distance reaches the minimum (the 18-th subset), the RMSE almost reaches the minimum. Obviously, there is no linear or quadratic relation between these two. This is why a threshold is needed in Alg. 1. The threshold not only helps in reducing the risk of excluding better candidate subsets but it also helps in determining the cost of finding the best subsets. That is, if the subsets are not properly selected, the cost to find the best candidate subsets is much higher. In this paper, we define the rank percentage of RMSE corresponding to the minimum distance η to measure the cost. More specifically, we executed the Alg. 1 K times with different subsets. The expectation $\hat{\kappa}$ of the RMSE rank corresponding to the minimum distance $\kappa_i, i = 1, \cdots, K$ is

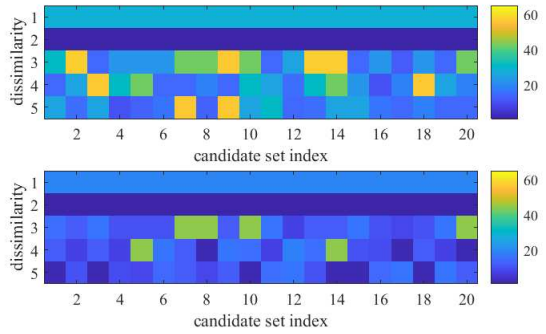

Fig. 4. Dissimilarities between $\#13$ and vicinity segments

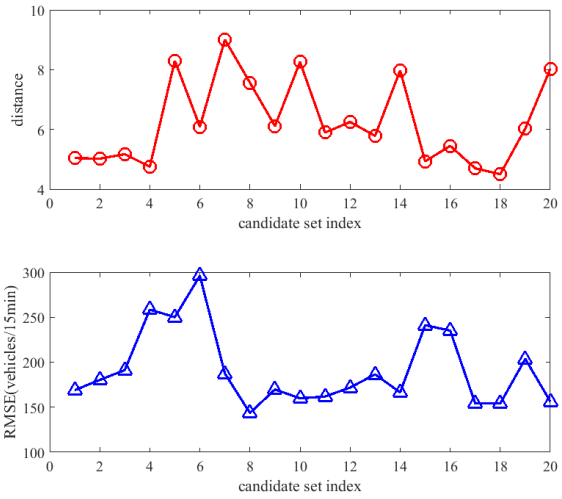

Fig. 5. Relation between dissimilarities and RMSEs

computed according to (17). Then η is determined by (18).

$$\hat{\kappa} = \frac{1}{K} \sum_{i=1}^{K} \kappa_i, \tag{17}$$

$$\eta = \hat{\kappa}/K. \tag{18}$$

We show the rank expectations for both sensor $\#11$ and $\#13$ with $K = 5$ in Tab. I. We can conclude that $\hat{\kappa} = 12.8$ is the best choice for sensor $\#11$, and $\hat{\kappa} = 4$ for sensor $\#13$. The rank percentages are $\eta_{\#11} = 64\%$ and $\eta_{\#13} = 20\%$, respectively. This indicates that the current selected subsets are proper for the prediction of sensor $\#13$ as the best prediction can be obtained within training 4 GP models while almost 13 models need to be trained for sensor $\#11$ (Be aware that 4 models are less than training a model for each sensor, which is 5). The reason lies in the fact that the distance between vicinity sensors to sensor $\#11$ is much bigger than sensor $\#13$, which is shown in Fig. 6. Therefore, we can set the threshold $r \in [6, 8]$.

The counts prediction RMSEs of the $K = 5$ experiments for sensors $\#11$ and $\#13$ are given in Tab. II. Obviously, RMSEs of sensor $\#11$ is bigger than that of sensor $\#13$. Also, from Tab. I we know that the cost to find the best

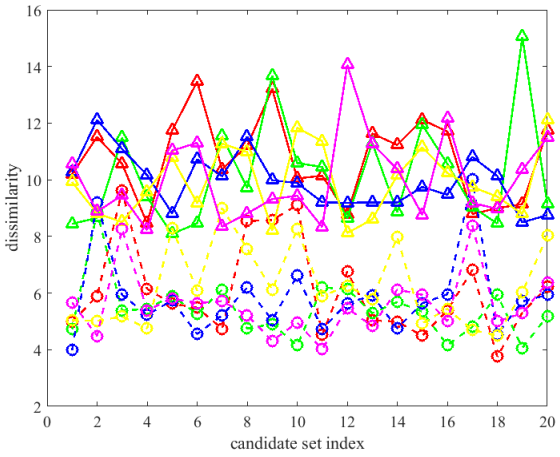

Fig. 6. Comparison of dissimilarities

candidate training subset for sensor #11 is higher than sensor #13. This is because the distance between vicinity segments to sensor #13 is much smaller than the distance to sensor #11. If we compare the RMSEs of the two sensors, we will see that almost 18.9% average improvement can be obtained. Now we can conclude that by using Alg. 1, with properly chosen threshold r, better prediction results can be obtained with lower costs.

TABLE I

RANK EXPECTATION

Sensor	κ_1	κ_2	κ_3	κ_4	κ_5	$\hat{\kappa}$
#11	11	8	14	17	14	12.8
#13	3	3	2	4	8	4

TABLE II

PREDICTION RMSES (VEHICLES/15MIN)

No.	1	2	3	4	5
#11	205.469	176.987	199.068	210.126	206.186
#13	156.765	159.677	154.160	167.319	169.066

VI. CONCLUSIONS

This paper proposes a Gaussian process algorithm using vicinity sensor measurements which we call a v-GP algorithm. The v-GP algorithm predicts the traffic flow even when the sensor data are missing, e.g. due to sensor failures. The algorithm consist two main parts. First, a dissimilarity matrix of the wDG is derivation and calculated. Unless there are no major changes to the traffic network both operations are executed only once. Second, in order to train v-GP model for the prediction, the local segment needs to be determined and the best vicinity subsets are selected. Results with real data show, that with the help of the dissimilarity matrix, one can choose the best subsets with lower costs, while still better prediction results can be achieved.

A future perspective is to design the policy that distributes the training task of v-GP and integrate the prediction results from different v-GP models to obtain the globally optimized prediction for data missing segments.

VII. ACKNOWLEDGEMENT

We appreciate the support of SETA project funded from the European Unions Horizon 2020 research and innovation programme under grant agreement No. 688082. We also thank the support of NSFC (61703387), Anhui Provincial NSF (1708085QF159) and the Fundamental Research Funds for the Central Universities (BJ2100100039).

REFERENCES

[1] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.Y. Wang, Traffic flow prediction with big data: A deep learning approach, *IEEE Transaction on Intelligent Transportation Systems*, vol. 16, no. 2, 2015, pp 865-873.

[2] Y. Xie, K. Zhao, Y. Sun, and D. Chen, Gaussian processes for short-term traffic volume forecasting, *Transportation Research Record* , vol. 2165, 2010, pp 69-78.

[3] T. Seo, A. M.Bayen, T. Kusakabe, and Y. Asakura, Traffic state estimation on highway: A comprehensive survey, *Annual Reviews in Control* , vol. 43, 2017, pp 128-151.

[4] M.S. Ahmed and A.R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, 1979.

[5] L.A. Pipes, An operational analysis of traffic dynamics. *Journal of applied physics* , vol. 24, 1953, pp 274-281.

[6] P.G. Gipps, A behavioural car-following model for computer simulation, *Transportation Research Part B: Methodological*, vol. 15, no. 2, 1981, pp.105-111.

[7] C.F. Daganzo, The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory, *Transportation Research Part B: Methodological* , vol. 28, no. 4, 1994, pp.269-287.

[8] A. Gning, L. Mihaylova, and R.K. Boel, Interval macroscopic models for traffic networks. *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, 2011, pp 523-536.

[9] S.P. Hoogendoorn, and P.H. Bovy, State-of-the-art of vehicular traffic flow modelling, *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering* , vol. 215, no. 4, 2001, pp 283-303.

[10] D. Ni, and J.D. Leonard, Markov chain monte carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data, *Transportation research record* , vol. 1935, no. 1, 2005, pp 57-67.

[11] T. Epelbaum, F. Gamboa, J.M. Loubes, and J. Martin, Deep Learning applied to Road Traffic Speed forecasting, *arXiv preprint arXiv:1710.08266*,2017.

[12] S. Du, T. Li, X. Gong, Z. Yu, and S.J. Horng, A Hybrid Method for Traffic Flow Forecasting Using Multimodal Deep Learning, textitarXiv preprint arXiv:1803.02099, 2018.

[13] C. E. Rasmussen and C. Williams, Gaussian Processes for Machine Learning, MIT Press, 2006.

[14] Y. Zhang and Y. Xie, Forecasting of short-term freeway volume with v-support vector machines, *Transportation Research Record* , vol. 2024, no. 1, 2007, pp 92-99.

[15] J. Chen, K.H. Low, Y. Yao and P. Jaillet, Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems, *IEEE Transactions on Automation Science and Engineering* , vol. 12, no. 3, 2015, pp 901-921.

[16] Y. Xie, K. Zhao, Y. Sun, and D. Chen, Gaussian processes for short-term traffic volume forecasting, *Transportation Research Record* , vol. 2165, no. 1, 2010, pp 69-78.

[17] T.V. Le, R. Oentaryo, S. Liu, and H.C. Lau, Local Gaussian processes for efficient fine-grained traffic speed prediction, *IEEE Transactions on Big Data*, vol. 3, no. 2, 2017, pp 194-207.

[18] (2018). EU SETA Project. Available: http://setamobility.weebly.com/

[19] A. Rodriguez, and A. Laio, Clustering by fast search and find of density peaks, *Science*, vol. 344, no. 6191, 2014, pp 1492-1496.