High-Order Gaussian Process Dynamical Models for Traffic Flow Prediction

Jing Zhao and Shiliang Sun

***Abstract*—Traffic flow prediction which predicts the future flow using the historic flows is an important task in intelligent transportation systems (ITS). Efficient and accurate models for traffic flow prediction greatly contribute to the development of ITS. In this paper, we adapt the Gaussian process dynamical model (GPDM) to a fourth-order GPDM which is more suitable for modeling traffic flow data. Specifically, the latent variables in the fourth-order GPDM is a fourth-order Markov Gaussian process, and the weighted k-NN is incorporated in the model to predict latent variables for efficient prediction. After training the model, the future flow is estimated by the average of the results predicted by the fourth-order GPDM and k-NN. Compared with other popular methods, the proposed method performs best and yields significant improvements of prediction performance.**

***Index Terms*—traffic flow prediction, high-order GPDM, dynamical system, Gaussian process, weighted k-NN.**

I. Introduction

TRAFFIC flow prediction is an important task for the application of intelligent transportation systems (ITS) [1, 2, 3, 4, 5]. With the rapid development of the society, there has been a large increase in urban traffic in recent years, resulting in many transportation problems such as congestions or accidents. ITS aims to address these problems through utilizing synergistic technologies and system engineering concepts, which can develop and improve transportation intelligently. Traffic flow prediction, as an essential task of ITS, is to predict traffic flows of a certain road link at a future time interval. Accurate and efficient prediction will make great significance for ITS. Since the traffic flow data are complex and varied, how to construct a good model is the key to the traffic flow prediction problem. In terms of timing, there are two types of traffic flow prediction: short-term and long-term. Short-term traffic flow prediction is to determine the traffic flow data in the next time interval, usually five to thirty minutes. We focus on the short-term prediction, in the time interval of 15 minutes, which is a difficult and very important application.

There exist some methods for traffic flow prediction. At the beginning, simple methods such as random walk (RW) and historical average (HA) [6] were proposed for specific situations. RW is to predict the current value using the last value. Then, there have been some elaborate methods presented for traffic flow prediction in succession such as Kalman filters [7], kernel regression [8], neural networks [9], Markov chain models [1] and selective random subspace predictor [10]. These methods have promoted the research of traffic flow modeling. However, these methods just used the historic flows from a single road link, which cannot make full use of the information of the data. For this purpose, Bayesian networks (BN) [11] were proposed to take advantage of the historic flow patterns of the target and its adjacent road links, and achieved promising experimental results. In order to capture more relationship among data, Jin and Sun [12] and Sun [13] have applied multi-task learning neural network (MTLNN) to improve traffic flow prediction by taking advantage of the MTL. For example for traffic flow prediction, when studying the relationship between y_1, y_2, ..., y_t and y_{t+1}, MTL can consider this current task as well as other related tasks, such as studying the relationship between y_1, y_2, ..., y_t and y_{t+2}. With regard to some other approaches of traffic flow prediction such as support vector regression, one can refer to Lippi et al. [14] in which the strengths and weaknesses of most existing methods were analyzed, and the seasonality in data was considered.

Besides the aforementioned methods, Gaussian process (GP) based methods have been developed for traffic flow prediction. Gaussian process regression (GPR) [15, 16] is a regression model which establishes a nonlinear mapping between the input and the output. It can be used for many forecasting applications such as weather forecasting [17] and traffic flow forecasting [18]. In order to model the multi-modal features of the traffic flow data, the infinite mixture of Gaussian processes (IMGP) [19] was applied to traffic flow prediction. It showed superiority than BN on the prediction performance when using historic flows from a single road link.

Differently from these supervised learning methods, there are some unsupervised learning methods for modeling time series. The Gaussian process dynamical model (GPDM) was recently proposed for modeling dynamic data. It augmented the Gaussian process latent variable model [20] with a dynamic prior to model sequential motion data and predict the latent positions [21]. GPDMs are widely applicable to sequential data analysis such as people tracking [22], motion grasping [23] and phoneme and gesture recognition [24, 25, 26]. It is an effective approach to modeling the dynamic data. However, its feasibility is unknown for traffic flow prediction. We adopt it for the following reasons. As a dynamical model, the GPDM can well model the dynamics of the traffic flow data. Moreover, the latent variable model has superiority of capturing the characteristics implicit in the complex data. In our work, considering the popularity of fourth-order traffic models [11, 12, 13, 19], we develop a fourth-order GPDM and apply

Manuscript received October 6, 2015; revised November 23, 2015; accepted December 28, 2015. This work was supported by NSFC Project 61370175.

Jing Zhao and Shiliang Sun (corresponding author) are with the Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China (e-mail: jzhao@ecnu.cn, jzhao2011@gmail.com; slsun@cs.ecnu.edu.cn, shiliangsun@gmail.com).

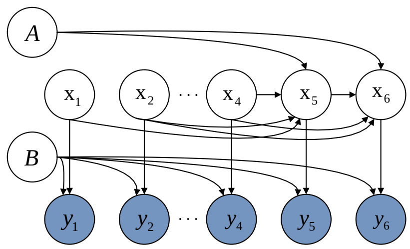

Fig. 1. Graphical model for the fourth-order dynamical system.

it to traffic flow prediction. Specifically, the latent variables in the fourth-order GPDM is assumed to be a fourth-order Markov Gaussian process, and the weighted k nearest neighbor (k-NN) is embedded into the model for prediction. There exists four algorithms for learning GPDMs [27, 28]: maximizing a posteriori (MAP), fixing the kernel hyperparameters $\bar{\alpha}$ (Fix.$\bar{\alpha}$), balanced GPDM and two-stage MAP. We employed MAP for estimation of parameters and latent variables in the fourth-order GPDM because traffic flow data have high variance and frequent volatility.

The highlights of this paper are summarized as follows. First, we adapt the original GPDM to a fourth-order GPDM, which is more applicable for traffic flow data. This is the first time to apply GPDMs to traffic flow prediction. Second, the weighted k-NN is well embedded into the fourth-order GPDM to achieve efficient prediction. Finally, compared with other methods, the proposed method performs best for predictions on multiple road links.

II. Fourth-Order Gaussian Process Dynamical Models

Traffic flow data are multivariate time series when taking multiple road links data into consideration. Dynamical systems are usually used for modeling such time series. GPDM is a kind of dynamical systems which has elegant formulations thanks to the Gaussian assumption. We first define a suitable dynamical system for traffic flow data. Then the fourth-order GPDM corresponding to this dynamical system is presented. Note that the dynamical function can be arbitrary order Markov functions. Since we focus on the problem of traffic flow prediction, we directly give descriptions of the fourth-order GPDM.

A. Dynamical Systems

When modeling multivariate time series, it is natural to incorporate dynamics into latent variable models. Under this framework, the system can be formulated as two functions. A dynamical function f, which is parameterized by A, with additive process noise $\mathbf{n}_{\mathbf{x},t}$ governing the evolution of $\mathbf{x}_t$. In our application, we use a fourth-order Markov dynamical function $\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \mathbf{x}_{t-3}, \mathbf{x}_{t-4}; A) + \mathbf{n}_{\mathbf{x},t}$. The other function is an observation function g, which is parameterized by B, with measurement noise $\mathbf{n}_{\mathbf{y},t}$ generating $\mathbf{y}_t$. The function can be expressed as $\mathbf{y}_t = g(\mathbf{x}_t; B) + \mathbf{n}_{\mathbf{y},t}$. From the perspective of probabilistic interpretation, this system can be illustrated as a graphical model in Figure 1.

B. Fourth-Order Gaussian Process Dynamical Models

Learning the dynamical systems typically involves estimating the parameters A and B and parameters for the noises. However, from a Bayesian perspective, the parameters should be marginalized out. Indeed, the GPDM estimates the latent variables while marginalizing over the model parameters. Similar to the first-order GPDM which assumes Gaussian priors of A and B and noises, the fourth-order GPDM can marginalize over f and g in closed form. Let $Y = [\mathbf{y}_1, ..., \mathbf{y}_N]^\top$ be the data in the observation space and $X = [\mathbf{x}_1, ..., \mathbf{x}_N]^\top$ be the variables in the latent space where $\mathbf{y}_i \in \mathbb{R}^D$ and $\mathbf{x}_i \in \mathbb{R}^d$. The likelihood of Y given X is expressed as a product of GPs (one for each of the D data dimensions)

$$\begin{aligned} &p(Y|X, \bar{\beta}, W) \\ &= \frac{|W|^N}{\sqrt{(2\pi)^{ND}|K_Y|^D}} \exp(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_Y^{-1} Y W^2 Y^\top)), \end{aligned} \tag{1}$$

where W is a scaling diagonal matrix and $\mathbf{K}_Y$ is an $N \times N$ kernel matrix constructed by a kernel function κ_Y with parameters $\bar{\beta} = \{\beta_i\}_{i=1}^3$:

$$\kappa_Y(\mathbf{x}, \mathbf{x}') = \beta_1 \exp(-\frac{\beta_2}{2}||\mathbf{x} - \mathbf{x}'||^2) + \beta_3^{-1}\delta_{\mathbf{x},\mathbf{x}'}. \tag{2}$$

The distribution of X is given by a fourth-order Markov Gaussian process

$$\begin{aligned} p(X|\bar{\alpha}) =& \frac{p(\mathbf{x}_1)p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_3|\mathbf{x}_2)p(\mathbf{x}_4|\mathbf{x}_3)}{\sqrt{(2\pi)^{(N-4)d}|\mathbf{K}_X|^d}} \\ & \exp(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_X^{-1} X_{5:N} X_{5:N}^\top)), \end{aligned} \tag{3}$$

where $\mathbf{x}_1$, $\mathbf{x}_2 - \mathbf{x}_1$, $\mathbf{x}_3 - \mathbf{x}_2$, $\mathbf{x}_4 - \mathbf{x}_3$ has the same standard Gaussian prior $\mathcal{N}(\mathbf{0}, I)$. $X_{5:N} = [\mathbf{x}_5^\top, ..., \mathbf{x}_N^\top]^\top$, $\mathbf{K}_X$ is a kernel matrix constructed from X_{in}=$[[\mathbf{x}_4, \mathbf{x}_3, \mathbf{x}_2, \mathbf{x}_1]^\top, ..., [\mathbf{x}_{N-1}, \mathbf{x}_{N-2}, \mathbf{x}_{N-3}, \mathbf{x}_{N-4}]^\top]^\top$. Here we use a "linear+RBF" kernel for $\mathbf{K}_X$ with parameters $\bar{\alpha} = \{\alpha_i\}_{i=1}^{10}$:

$$\begin{aligned} &k_X([\mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \mathbf{x}_{t-3}, \mathbf{x}_{t-4}], [\mathbf{x}_{\tau-1}, \mathbf{x}_{\tau-2}, \mathbf{x}_{\tau-3}, \mathbf{x}_{\tau-4}] \\ &= \alpha_9 \exp(-\frac{\alpha_1}{2}\|\mathbf{x}_{t-1} - \mathbf{x}_{\tau-1}\|^2 - ... - \frac{\alpha_4}{2}\|\mathbf{x}_{t-4} - \mathbf{x}_{\tau-4}\|^2) \\ &\quad + \alpha_5 \mathbf{x}_{t-1}^\top \mathbf{x}_{\tau-1} + ... + \alpha_8 \mathbf{x}_{t-4}^\top \mathbf{x}_{\tau-4} + \alpha_{10}^{-1}\delta_{t,\tau}, \end{aligned} \tag{4}$$

which is very different from the general kernel function. The priors of the kernel hyperparameters are placed with $p(\bar{\alpha}) \propto \prod_i \alpha_i^{-1}$ and $p(\bar{\beta}) \propto \prod_i \beta_i^{-1}$. Parameter W has a broad half-normal prior, $p(W) = \prod_{d=1}^D \frac{2}{\sigma\sqrt{2\pi}}\exp(-\frac{w_d^2}{2\sigma^2})$, where $w_d > 0$ corresponds to the diagonal elements of W and σ is often fixed to 10^3 as in [27].

Note that, Y can represent multiple sequences $Y^{(1)}, ..., Y^{(P)}$, with lengths $N_1, ..., N_P$. Then $X_{5:N}$ is composed of the associated latent variables $X^{(1)}, ..., X^{(P)}$ as $X_{5:N} = [X_{5:N_1}^{(1)\top}, ..., X_{5:N_P}^{(P)\top}]^\top$ and X_{in} is given by $X_{\text{in}} = [[X_{4:N_1-1}, X_{3:N_1-2}, X_{2:N_1-3}, X_{1:N_1-4}]^{(1)\top}, ..., [X_{4:N_P-1}, X_{3:N_P-2}, X_{2:N_P-3}, X_{1:N_P-4}]^{(P)\top}]^\top$. Given the above expressions, the joint probability distribution of latent variables, observations, and parameters is given by

$$p(X, Y, \bar{\alpha}, \bar{\beta}, W) = p(Y|X, \bar{\beta}, W)p(X|\bar{\alpha})p(\bar{\alpha})p(\bar{\beta})p(W).$$

C. Learning Methods

The existing learning methods for GPDMs are MAP, Fix.α, B-GPDM and T.MAP. As analyzed in [27, 28], the latter three algorithms tend to get smoother trajectories of latent variables here. In the application of traffic flow prediction, data have high variance and frequent volatility. Thus it is undesirable to get smooth latent trajectories here. Therefore, we employ the MAP for the estimation of parameters and latent variables. MAP requires minimizing the joint negative log-posterior of the unknowns $-\ln p(X, \bar{\alpha}, \bar{\beta}, W|Y)$. It is expressed as $\mathcal{L}+C$ where C represents a constant and $\mathcal{L}$ is expressed as

$$\mathcal{L} = \mathcal{L}_Y + \mathcal{L}_X + \sum\nolimits_j \ln\beta_j + \sum\nolimits_j \ln\alpha_j + \frac{\mathrm{tr}(W^2)}{2\sigma^2}, \quad (5)$$

with

$$\mathcal{L}_Y = \frac{D}{2}\ln|\mathbf{K}_Y| + \frac{1}{2}\mathrm{tr}(\mathbf{K}_Y^{-1}YW^2Y^\top) - N\ln|W|, \quad (6)$$

$$\begin{aligned}\mathcal{L}_X = &\frac{d}{2}\ln|\mathbf{K}_X| + \frac{1}{2}\mathrm{tr}(\mathbf{K}_X^{-1}X_{5:N}X_{5:N}^\top) + \frac{1}{2}\mathbf{x}_1^\top\mathbf{x}_1 \\ &+ \frac{1}{2}\sum_{i=1}^{3}(\mathbf{x}_{i+1}-\mathbf{x}_i)^\top(\mathbf{x}_{i+1}-\mathbf{x}_i).\end{aligned} \quad (7)$$

III. Traffic Flow Prediction

So far, we have defined the fourth-order GPDM, and the parameters and latent variables can be optimized by MAP estimation. This model can be used for predicting the future data given the previous data. Specific to the application of traffic flow prediction, the flows from adjacent road links at previous four time intervals $Y_{\text{prev}} = [\mathbf{y}_{t-4}^\top, \mathbf{y}_{t-3}^\top, \mathbf{y}_{t-2}^\top, \mathbf{y}_{t-1}^\top]^\top$ are used for predicting future flow y_t^* denoted by Y_{curr}. In our fourth-order GPDM, prediction is realized by two procedures: latent variable prediction and future flow prediction. For latent variable prediction, the original GPDM optimized the X^* by maximizing the conditional distribution $p(Y_{\text{prev}}^*, X_{\text{prev}}^*|\Gamma)$ where Γ represents the learned model. However, this is not suitable for efficient prediction. In our methods, X^* is obtained by the weighted k-NN method on the observed data. For future flow prediction, data are estimated by combining GPDM reconstruction and k-NN regression.

A. Latent Variable Prediction

Given the learned fourth-order GPDM, we compute the most likely latent variable $X^* = [X_{\text{prev}}^{*\top}, X_{\text{curr}}^{*\top}]^\top$ through the weighted k-NN on observed data. We can get information from the learned model. The parameters $\bar{\alpha}$ and W reflect the time and spatial dependence among data.

From the perspective of time dependence, we know that data from different time intervals make different effect on the current data. In the fourth-order GPDM, for the current latent variable $\mathbf{x}_t$, the hyperparameter $\bar{\alpha}$ controls the affects of $\mathbf{x}_{t-4}, ..., \mathbf{x}_{t-1}$ on the value the kernel k_X. For example in (4), α_1 and α_5 as the coefficients of $\|\mathbf{x}_{t-1} - \mathbf{x}_{\tau-1}\|$ and $\mathbf{x}_{t-1}^\top\mathbf{x}_{\tau-1}$ control the importance of $\mathbf{x}_{t-1}$'s role. We want to use the relative magnitudes of $\alpha_1, ..., \alpha_8$ to measure the affects of historic flows at different time intervals on the current flow. For this purpose, we define a new variable $\tilde{\boldsymbol{\alpha}}$ which will be used for calculating the distance of two time series. It is a four-element vector, with each element

$$\tilde{\alpha}_i = \frac{\alpha_i + \alpha_{4+i}}{\sum_1^8 \alpha_i}. \quad (8)$$

From the perspective of spatial dependence, we know that the affects of different road links on the target road link are also different. In our GPDM, the parameter W models the variance in each observation dimension. It plays a role at the connection between the observed data Y and the latent variable X. We can analyze that the bigger the value of w_d is, the larger the affect of $\mathbf{y}_d$ on the model. Thus we use $\frac{1}{\sqrt{w_d}}$ as the weight of distance. We use $\frac{1}{\sqrt{w_d}}$ instead of $\frac{1}{w_d}$ to reduce the spatial dependence to a certain extent. This makes a contribution to balancing spatial and time dependence during calculating distances.

For latent variable prediction, the distances should be calculated between Y_{prev}^* and Y. The distance between two series e.g., S and S', is defined as

$$\mathrm{DIS} = \sum_{d=1}^{D}\frac{1}{\sqrt{w_d}}\|(\mathbf{s}_d - \mathbf{s}_d')\odot\tilde{\boldsymbol{\alpha}}\|_2, \quad (9)$$

where $\odot$ represents pointwise product. With the distance defined by (9), and the latent variable X learned by the fourth-order GPDM, we can compute the latent variable X^* according to the weighted k-NN.

Algorithm 1 Prediction with fourth-order GPDM and k-NN.

Input: Data Y, Y_{prev}^*. Integers $\{d, I, J, k\}$.
Output: Data y_{curr}^*.

1: Alternate for I iterations to learn the GPDM with MAP estimation of $\{X, \bar{\alpha}, \bar{\beta}, W\}$:
I. **Do** $w_j^2 \leftarrow N(Y_{:,j}^\top\mathbf{K}_Y^{-1}Y_{:,j} + \frac{1}{\sigma^2})^{-1}$ with $j = 1...D$.
II. Optimize $\{X, \bar{\alpha}, \bar{\beta}\}$ by maximizing (5) using the scaled conjugate gradient (SCG) for J iterations.
2: Calculate the latent variable X^* for test data Y^*.
I. Calculate the distances between Y_{prev}^* and Y by (9).
II. Calculate X^* using the weighted k-NN.
3: Predict Y_{curr}^* by combining GPDM reconstruction and k-NN regression.
I. Estimate the future data Y_{GPDM}^* according to GPDM reconstruction and $Y_{k\text{-NN}}^*$ according to k-NN regression.
II. Ensure the final Y^* as the average of Y_{GPDM}^* and $Y_{k\text{-NN}}^*$.

B. Future Flow Prediction

We have computed the previous latent variables X_{prev}^* corresponding to Y_{prev}^* and the current latent variable X_{curr}^* corresponding to Y_{curr}^* according to the weighted k-NN. The current data $Y_{\text{curr}}^* \in \mathbb{R}^D$ can be estimated using the learned fourth-order GPDM and previous data Y_{prev} as well as X^*. Defining $\tilde{Y} = [Y^\top, Y_{\text{prev},:}^{*\top}]^\top$, and $\tilde{X} = [X^\top, X_{\text{prev}}^{*\top}]^\top$, the posterior density of Y_{curr}^* is

$$\begin{aligned}&p(Y_{\text{curr}}^*|X^*, Y_{\text{prev}}^*, \Gamma) \\ &= \frac{|W|^4}{\sqrt{(2\pi)^{4D}|\mathbf{K}_{\tilde{Y}^*}|^D}}\exp(-\frac{1}{2}\mathrm{tr}(\mathbf{K}_{\tilde{Y}^*}^{-1}Z_{\tilde{Y}}W^2Z_{\tilde{Y}}^\top)),\end{aligned}$$

where $\tilde{Z}_Y = Y_{\text{curr}}^* - \tilde{A}^\top\mathbf{K}_{\tilde{Y}}^{-1}\tilde{Y}$ and $\mathbf{K}_{\tilde{Y}^*} = \tilde{B} - \tilde{A}^\top\mathbf{K}_{\tilde{Y}}^{-1}\tilde{A}$. $(\tilde{A})_{ij} = \kappa_Y(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{\text{curr}}^*)$ and $(\tilde{B})_{ij} = \kappa_Y(\tilde{\mathbf{x}}_{\text{curr}}^*, \tilde{\mathbf{x}}_{\text{curr}}^*)$ are the

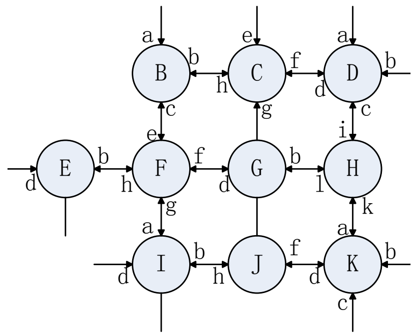

Fig. 2. A patch taken from the urban traffic map of highways.

elements of the $(N+4)\times 1$ and 1×1 kernel matrices, respectively. Considering that the posterior of Y^*_{curr} is Gaussian, we can use its mean $\tilde{A}^\top \mathbf{K}^{-1}_{\tilde{Y}}\tilde{Y}$ to estimate it.

Since the training data are sufficient, k-NN regression can obtain good performance in this case. We finally combine the results from the fourth-order GPDM denoted with Y^*_{GPDM} and the result from k-NN regression denoted with $Y^*_{k\text{-NN}}$ to predict the future data $Y^*_{\text{curr}} = \frac{1}{2}(Y^*_{\text{GPDM}} + Y^*_{k\text{-NN}})$. In order to make the whole process easy understanding, we give the framework of our method for traffic flow prediction in Algorithm 1. The things to notice are as follows. Integers I and J are the iteration numbers which are set by hand. Integer d is set according to specific data. Number k for k-NN is chosen from a range by validation on the training set.

IV. Experiments

In our experiments, the task is to predict the future traffic flows of a certain road link in a real urban transportation network [10]. Figure 2 shows a patch from the urban traffic map of highways in Beijing. Each circle node in the sketch map denotes a road link. An arrow shows the direction of the traffic flow. Paths without arrows are of no traffic flow records. The raw data are records from 25 days which include 2400 recording points for each road link. The first 22 days are used for training, and the remaining three days are for testing.

We predict the traffic volumes (vehicles/hour) of the road links (‘Bb’, ‘Bc’, ‘Cf’, ‘Ch’, ‘Dc’, ‘Dd’, ‘Fe’, ‘Gb’, ‘Hi’, ‘Ka’) based on their own historic traffic flows and their direct upstream flows. For example, for the target road link ‘Ka’, the traffic flows from its upstream road links ‘Hi’, ‘Hl’, ‘Hk’ are used. In our fourth-order GPDM, the dimension of latent variables is fixed as two. This is because one target road link may have one, two or three neighbors, which leads to the dimension of observed data being two, three or four. The experiments using the fourth-order GPDM with one-dimensional latent space resulted in poor prediction performance due to the fact that the one-dimensional latent space cannot capture enough information. Thus the dimension of latent variables is set to two uniformly for all the road links. We fix the number of outer loop iteration as $I = 10$ and the number of SCG iteration as $J = 10$. After learning the fourth-order GPDM, number k for k-NN is determined in the training phase. It is chosen by validating on the training set. Since the traffic flow data have sequential relations, it is not suitable to use cross-validation. We simply validate on the last three days of the training set. We choose the best k from the range of $16 \sim 20$ for every road link.

TABLE I
RMSE / MAPE of prediction results for ten road links with 4th-order GPDM, 2nd-order GPDM and RW.

Road Link	4th-order GPDM	2nd-order GPDM	RW
Bb	**69.19 / 8.02**	69.59 / 8.04	89.60 / 10.63
Bc	**106.37 / 8.99**	107.09 / 9.03	124.24 / 11.75
Cf	**89.84 / 8.48**	90.08 / 8.52	108.24 / 9.91
Ch	**61.92 / 9.31**	61.95 / 9.38	80.03 / 13.33
Dc	**79.04 / 13.26**	79.36 / 13.27	97.91 / 16.70
Dd	**55.22** / 10.22	55.53 / **10.20**	71.44 /14.90
Fe	113.72 / **7.55**	**113.37** / 7.56	158.55 /10.22
Gb	**83.03 / 12.47**	83.24 / 12.47	101.80 / 15.60
Hi	85.20 / 11.34	**85.07 / 11.33**	108.34 / 15.67
Ka	**70.09** / 8.26	70.99 / **8.22**	99.77 / 11.65
AVG.	**81.36 / 9.79**	81.63 / 9.80	103.99 / 13.04

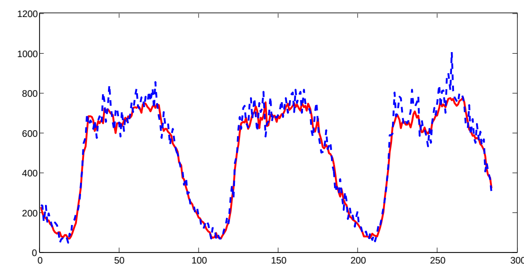

Fig. 3. Prediction results of the fourth-order GPDM for road link ‘Dd’.

We compare our method to multiple existing methods including basic methods (RW) and classical methods (k-NN and GPR) as well as some advanced methods (IMGP and MTLNN). The prediction performance is measured by the criteria of root mean square error (RMSE) and mean absolute percentage error (MAPE). For a time series $\mathbf{y}^* = (y^*_1, y^*_2, ...y^*_M)$ and its estimation $\tilde{\mathbf{y}}^*$, RMSE and MAPE are given by the following formulae: $\text{RMSE} = (\frac{1}{M}\sum_{i=1}^{M}(\tilde{y}^*_i - y^*_i)^2)^{1/2}$, $\text{MAPE} = \frac{1}{M}\sum_{i=1}^{M}\frac{|\tilde{y}^*_i - y^*_i|}{|y^*_i|}$. RMSE is a measuring criterion from the perspective of the absolute error while MAPE is from the perspective of the relative error.

First of all, in order to illustrate the reasonability and advantage of using the high-order GPDM, we compare our method with the second-order GPDM and RW. In the second-order GPDM, the training process is similar to the fourth-order GPDM, but with the latent variables being a second-order Markov Gaussian process. Table I shows the results of prediction using these three methods in terms of RMSE and MAPE. The best results for each road link are in bold. From the table, we find that both GPDM based methods have a large increase on the prediction accuracy than the baseline RW method. Further, the fourth-order GPDM works better than the second-order GPDM since it has considered the previous four time intervals’ data from own and adjacent road links. It is popular to set the order number to four because four time intervals’ historic data can provide enough information for traffic flow prediction empirically [11]. Figure 3 shows the prediction results of road link ‘Dd’ with the fourth-order GPDM. The solid line shows the predicted data while the dotted line shows the observed data.

Then, we compare our method with some classical methods, GPR and k-NN regression. GPR is a kind of supervised learning method. In our experiment, we construct the input

TABLE II
RMSE / MAPE OF PREDICTION RESULTS FOR TEN ROAD LINKS WITH 4TH-ORDER GPDM, k-NN AND GPR.

Road Link	4th-order GPDM	k-NN	GPR
Bb	69.19 / **8.02**	**69.05** / 8.11	69.49 / 8.51
Bc	**106.37 / 8.99**	110.83 / 9.23	107.14 / 9.22
Cf	89.84 / 8.48	**87.96 / 8.29**	100.13 / 9.16
Ch	61.92 / **9.31**	64.44 / 9.75	**61.17** / 9.81
Dc	**79.04** / 13.26	81.58 / 13.55	83.40 / **13.16**
Dd	**55.22 / 10.22**	56.50 / 10.31	56.20 / 10.86
Fe	113.72 / 7.55	115.59 / 7.73	**111.91 / 7.32**
Gb	83.03 / **12.47**	83.39 / 12.53	**82.52** / 13.30
Hi	85.20 / **11.34**	**84.29** / 11.56	86.32 / 11.61
Ka	**70.09 / 8.26**	71.34 / 8.35	71.19 / 8.56
AVG.	**81.36 / 9.79**	82.50 / 9.94	82.95 / 10.15

TABLE III
RMSE / MAPE OF PREDICTION RESULTS FOR TEN ROAD LINKS WITH 4TH-ORDER GPDM, IMGP AND MTLNN.

Road Link	4th-Order GPDM	IMGP	MTLNN
Bb	**69.19 / 8.02**	71.33 / 8.12	72.26 / 8.42
Bc	106.37 / 8.99	**98.10** / 8.87	103.23 / **8.59**
Cf	**89.84 / 8.48**	101.09 / 9.50	94.95 / 8.94
Ch	61.92 / **9.31**	62.25 / 9.40	**60.01** / 9.52
Dc	79.04 / 13.26	**79.03 / 12.99**	80.13 / 13.22
Dd	**55.22 / 10.22**	59.42 / 11.00	55.74 / 10.64
Fe	113.72 / 7.55	**111.99 / 7.30**	116.09 / 7.72
Gb	**83.03 / 12.47**	85.48 / 12.80	82.67 / 13.52
Hi	**85.20 / 11.34**	88.62 / 11.90	88.54 / 11.75
Ka	**70.09 / 8.26**	70.83 / 8.26	69.73 / 8.34
AVG.	**81.36 / 9.79**	82.81 / 10.01	82.34 / 10.07

data by Y_{prev} and the output data by the current flow y_t from the target road link. Exact inference is used in the GPR and the iteration number of SCG optimization for hyperparameters is set to 100. The other compared method, k-NN regression, is simply averaging the numerical target of the k nearest neighbors. The number k is chosen through validating the prediction results of the last three days in the training set. From Table II, we conclude that our method performs better than the other two methods on average, no matter in terms of RMSE and MAPE. When making pairwise comparison, we can see traffic flows from most road links are predicted more accurately by the fourth-order GPDM than k-NN regression and GPR, respectively. Especially in terms of MAPE, our method improves a lot in most cases. We can analyze from the point of view of the characteristics of these methods. The fourth-order GPDM as a multiple layer GPs can grasp more information from data than GPR. GPR makes little use of the dynamics in the data. Moreover, the fourth-order GPDM reconstructs data using latent variables instead of directly finding data from the training set like k-NN regression. The results estimated by the fourth-order GPDM are the best.

Finally, we compare our method with two advanced methods, IMGP and MTLNN. IMGP is an mixture model of GPs, which can model multi-modal data. In Sun and Xu [19], data used for prediction are restricted to recordings from a single road link. Here, we use the data from own and adjacent road links to analyze the trend of the target road link. The input and output are constructed as in GPR. The specific training process for IMGP can be found in [19]. In the current experiment, parameters in the algorithm input are fixed as T = 5, $C = 100$, $S = 50$, $M_s = 10$, $M_{em} = 20$, $M_e = 20$, $M_m = 20$ referring to [19]. MTLNN in [13] for traffic flow prediction adds extra but related units to the output layer of the neural network. The input and output used for prediction is taken from a single road link. Unlike [13], for one target road link, we use several adjacent road links' recordings to train the neural network. Predictions of current traffic flows of multiple road links used for training are added as multiple tasks. The layer number and the method of ensuring unit number for one layer are the same as [13]. Table III shows the experimental results, in which our method performs best. This is attributed to the ability of capturing dynamics of data by latent variables.

A. Overall Evaluation

In this part, we make the overall evaluation of our method in terms of effectiveness, reasonability and efficiency. We perform paired t-tests for our method with the others, respectively. Table IV lists the test results, in which symbol $\upuparrows$ represents the proposed method makes a significant improvement at 95% confidence level and the value in parentheses represents the confidence level. We find that at least under one kind of criterions, the proposed approach has an improvement that is statistically significant with respect to other methods except the IMGP which is a mixture model.

TABLE IV
PAIR-T-TEST WITH ALL OTHER COMPARED METHODS.

Methods	w.r.t. RMSE	w.r.t. MAPE
2nd-Order GPDM	$\upuparrows$ (95%)	$\uparrow$ (80%)
GPR	$\uparrow$ (82%)	$\upuparrows$ (99%)
k-NN	$\uparrow$ (92%)	$\upuparrows$ (98%)
IMGP	$\uparrow$ (63%)	$\uparrow$ (86%)
MTLNN	$\uparrow$ (74%)	$\upuparrows$ (95%)
RW	$\upuparrows$ (100%)	$\upuparrows$ (100%)

The GPDM is a kind of latent variable model. The advantage of the latent variable is to map the complex characteristics of data into a low-dimensional space. The latent variables in the low-dimensional space can capture the internal driving force in dynamic data. From the experiments on ten road links, we can verify that the fourth-order GPDM with two dimensional latent space is reasonable. We plot the learned latent variables for three days on road link 'Ka' in Figure 4. From the figure, we can find that the latent variables on two dimensions fluctuate at different frequencies and have physical meanings. One dimension is at low frequencies, which grasps the macroscopic dynamics of flows from the related road links. Further, its fluctuation trend basically coincides with that of the observed traffic flows. The other dimension is at high frequencies, which grasps the details including the influences from the adjacent road links. Therefore, it is reasonable to use latent variable models to model traffic flow data.

In order to evaluate our method in terms of efficiency, we compare the computation complexity and runtime for different methods. We first analyze the complexity in training procedure. The complexity of our method is $cO(N^3)$, where c is a constant representing the number of iterations. For GPR, the most time-consuming part is also to calculate the inverse of covariance matrix whose complexity is $O(N^3)$. For k-NN regression, the complexity of choosing k is $cO(N)$ where c is the size of the searching range for k. For IMGP, the most time-consuming operation is to optimize the hyperparameters with complexity $O(NT)$. In addition, there is nested loops in IMGP for optimization, which greatly increases the runtime. For MTLNN, the complexity depends on the architecture of

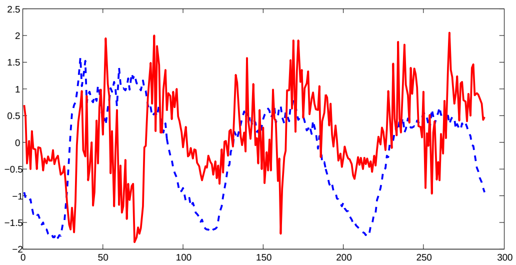

Fig. 4. Learned latent variables for three days on road link 'Ka'.

TABLE V
RUNTIME (IN MINUTES) OF TRAINING FOR ONE TARGET ROAD LINK.

Methods	Proposed Method	GPR	k-NN	IMGP	MTLNN
Time	68	2	3	2005	6

the network. We list the runtime of training processes for one target road link in Table V. Note that the training time for the second-order GPDM is almost the same to the fourth-order GPDM. We only list the runtime for the fourth-order GPDM. In general, the training time of our method is acceptable as the training procedure is offline. In addition, the reconstruction and averaging procedures in our method lead to efficient prediction.

V. CONCLUSION

In this paper, we adapted the GPDM to a high-order GPDM in which the latent variables are assumed to be a fourth-order Markov GP. The resulting fourth-order GPDM can well model the traffic flow data. Weighted k-NN is embedded into the model, which helps to achieve efficient prediction. Compared with other popular methods, the proposed method performs best on prediction and yields significant improvements. In the future work, it is interesting to use mixtures of GPDMs to model time series and try other ways of defining distances for the weighted k-NN.

REFERENCES

[1] G. Yu, J. Hu, C. Zhang, L. Zhuang, and J. Song, "Short-term traffic flow forecasting based on Markov chain model," in *Proceedings of IEEE Intelligent Vehicles Symposium*, 2003, pp. 208–212.

[2] T. Thomas, W. Weijermars, and E. van Berkum, "Predictions of urban volumes in single time series," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 71–80, 2010.

[3] G. Vigos and M. Papageorgiou, "A simplified estimaton scheme for the number of vehicles in signalized links," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 312–321, 2010.

[4] F. Y. Wang, "Parallel control and management for intelligent transportation systems: concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 630–638, 2010.

[5] ——, "Building an intellectual highway for its research and development," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 2–3, 2010.

[6] B. M. Williams, "Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process," Ph.D. dissertation, University of Virginia, 1999.

[7] I. Okutani and Y. Stephanedes, "Dynamic prediction of traffic volume through Kalman filter theory," *Transportation Research Part B: Methodological*, vol. 18, pp. 1–11, 1984.

[8] G. Davis and N. Nihan, "Nonparameteric regression and short-term freeway traffic forecasting," *Journal of Transportaion*, vol. 177, pp. 177–188, 1991.

[9] E. Yu and C. Chen, "Traffic prediction using neural networks," in *Proceedings of IEEE Global Telecommun Conference*, 1993, pp. 991–995.

[10] S. Sun and C. Zhang, "The selective random subspace predictor for traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, pp. 367–373, 2007.

[11] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, pp. 124–132, 2006.

[12] F. Jin and S. Sun, "Neural network multitask learning for traffic flow forecasting," in *Proceedings of International Joint Conference on Neural Networks*, 2008, pp. 1898–1902.

[13] S. Sun, "Traffic flow forecasting based on multitask ensemble learning," in *Proceedings of Genetic and Evolutionary Computation Conference*, 2009, pp. 961–964.

[14] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, pp. 871 – 882, 2013.

[15] C. E. Rasmussen and C. K. I. Williams, *Gaussian Process for Machine Learning*. MIT Press, 2006.

[16] C. M. Bishop, *Patten Recognition and Machine Learning*. New York: Springer, 2006.

[17] P. Boyle, "Gaussian process for regression and optimisation," Ph.D. dissertation, Victoria University of Wellington, 2007.

[18] Y. Xie, K. Zhao, Y. Sun, and D. Chen, "Gaussian processes for short-term traffic volume forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2165, pp. 69–78, 2010.

[19] S. Sun and X. Xu, "Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, pp. 466–475, 2011.

[20] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," *Advances in Neural Information Processing Systems*, vol. 17, pp. 329–336, 2004.

[21] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," *Advances in Neural Information Processing Systems*, vol. 19, pp. 1441–1448, 2006.

[22] R. Urtasun, D. J. Fleet, and P. Fua, "3D people tracking with Gaussian process dynamic models," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 238–245.

[23] B. An, H. K, and F. C. Park, "Grasp motion learning with Gaussian process dynamic models," in *Proceedings of IEEE International Conference on Automation Science and Engineering*, 2012, pp. 1114–1119.

[24] H. Park and C. D. Yoo, "Gaussian process dynamical models for phoneme classification," in *Proceedings of Neural Information Processing System Workshop on Bayesian Nonparametrics: Hope or Hype*, 2011, pp. 1–2.

[25] G. E. Henter, M. R. Frean, and W. B. Kleijn, "Gaussian process dynamical models for nonparametric speech representation and synthesis," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 4505–4508.

[26] N. Gamage, T. C. Kuang, R. Akmeliawati, and S. Demidenko, "Gaussian process dynamical models for hand gesture interpretation in sign language," *Pattern Recognition Letters*, vol. 32, pp. 2009–2014, 2011.

[27] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 283–398, 2008.

[28] J. Zhao and S. Sun, "Revisting Gaussian process dynamical models," in *Proceedings of 24th International Joint Conference on Artificial Intelligence*, 2015, pp. 1047–1053.