Road Traffic Speed Prediction: A Probabilistic Model Fusing Multi-Source Data

Lu Lin, Jianxin Li, Feng Chen, Jieping Ye, *Senior Member, IEEE*, and Jinpeng Huai

Abstract—Road traffic speed prediction is a challenging problem in intelligent transportation system (ITS) and has gained increasing attentions. Existing works are mainly based on raw speed sensing data obtained from infrastructure sensors or probe vehicles, which, however, are limited by expensive cost of sensor deployment and maintenance. With sparse speed observations, traditional methods based only on speed sensing data are insufficient, especially when emergencies like traffic accidents occur. To address the issue, this paper aims to improve the road traffic speed prediction by fusing traditional speed sensing data with new-type "sensing" data from cross domain sources, such as tweet sensors from social media and trajectory sensors from map and traffic service platforms. Jointly modeling information from different datasets brings many challenges, including location uncertainty of low-resolution data, language ambiguity of traffic description in texts, and heterogeneity of cross-domain data. In response to these challenges, we present a unified probabilistic framework, called Topic-Enhanced Gaussian Process Aggregation Model (TEGPAM), consisting of three components, i.e., location disaggregation model, traffic topic model, and traffic speed Gaussian Process model, which integrate new-type data with traditional data. Experiments on real world data from two large cities validate the effectiveness and efficiency of our model.

Index Terms—Traffic prediction, gaussian process, topic modeling, multi-source data

1 Introduction

1.1 Background and Motivation

Road traffic monitoring is of great importance for urban transportation system. Traffic control agencies and drivers could benefit from timely and accurate road traffic prediction and make prompt, or even advance decisions possible for detecting and avoiding road congestions. Existing methods mainly focus on raw speed sensing data collected from cameras or road sensors, and suffer severe data sparsity issue because the installation and maintenance of sensors are very expensive [1]. At the same time, most existing techniques based only on past and current traffic conditions (e.g., [2], [3], [4], [5]) do not fit well when real-world factors such as traffic accidents play a part.

To address the above issues, in this paper we introduce new-type traffic related data arising from public services: 1) *Social media data*, which is posted on social networking websites, e.g., Twitter and Facebook. With the popularization of mobile devices, people are more likely to exchange news and trifles in their life through social media services, where messages about traffic conditions, such as "*Stuck in traffic on E 32nd St. Stay away!*", are posted by drivers, passengers and pedestrians who can be viewed as sensors observing the ongoing traffic conditions near their physical locations. Meanwhile, traffic authorities register public accounts and post tweets to inform the public of the traffic status, such as "*Slow traffic on I-95 SB from Girard Ave to Vine St.*" posted by local transportation bureau account. Such text messages describing traffic conditions—and some of them tagged with location information—are accessible by public and could be a complementary information source of raw speed sensing data. 2) *Car trajectory data*, which is obtained from moving cars by applications installed in the drivers' cell phones. Real-time map and traffic services, e.g., Google Map and Uber, featured by location and navigation functions make travel more convenient. Given an origin-destination (OD) pair on a map, such services can recommend optimal route from the origin to the destination with least time, and trajectories can be collected once drivers use the service to navigate. Here a *trajectory* is a sequence of links for a given OD pair, and a *link* is a road segment between neighboring intersections. Correspondently, a *trajectory travel time* is an integration of *link travel times*, which are related to the real-time road traffic speeds. Longer trajectory travel time indicates that some involving road links may be congested with lower traffic speed. Trajectory data is useful for a wide range of transportation analyses and applications [2], [6].

Based on the above observations, where traditional traffic sensing data are limited while new-type data from social media and map service begin to spring up, our goal is to predict the road-level traffic speed by incorporating new-type data with traditional speed sensing data. To motivate this scenario, consider a road traffic prediction example depicted in Fig. 1. Those links in red question marks are not covered by traditional speed sensors, but may be passed by trajectories attached with travel time information, or mentioned in tweets describing traffic conditions, so their speeds can be inferred fusing multiple cross-domain data.

- *L. Lin, J. Li, and J. Huai are with Beihang University, Beijing 100191, China. E-mail: {linlu, lijx}@act.buaa.edu.cn, huaijp@buaa.edu.cn.*
- *F. Chen is with the State University of New York at Albany, Albany, NY 12222. E-mail: fchen5@albany.edu.*
- *J. Ye is with the University of Michigan, Ann Arbor, MI 48109. E-mail: jpye@umich.edu.*

Manuscript received 23 Jan. 2016; revised 16 May 2017; accepted 12 June 2017. Date of publication 22 June 2017; date of current version 1 June 2018. (Corresponding author: Jianxin Li.)
Recommended for acceptance by J. Bailey.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2017.2718525

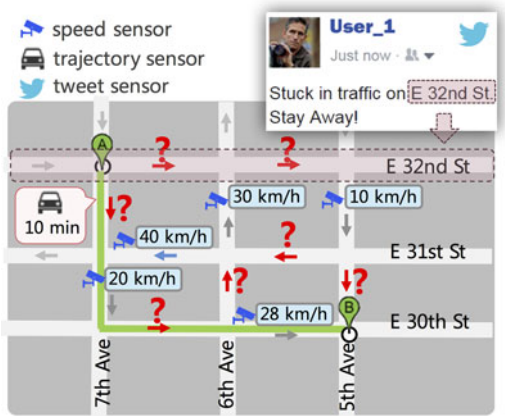


Fig. 1. Problem setting. Our goal is to predict the traffic speed of specific road links, as shown with the red question marks, given: 1) Some speed observations collected by speed sensors, as shown in blue; 2) trajectory and travel time of OD pairs. Note that speeds of passed road links are either observed or to be predicted; 3) tweets describing traffic conditions. Note that the location mentioned by a tweet may be a street covering multiple road links.

1.2 Challenges

When integrating traditional traffic speed data (e.g., sensing data) with new-type data (e.g., Twitter data and trajectory data) to predict road traffic speed, technical challenges arise due to the characteristic of each data source:

- *Location uncertainty of low-resolution data.* Tweet data and trajectory data are called low-resolution data because we cannot directly locate them into specific road links. Most tweets have no location tags, so geographic location language is the main clue, which however is vague. For example, expression like "*Stuck in traffic on E 32nd St. Stay away!*" covers the whole street without precise road locations. Meanwhile, travel time of a trajectory is an aggregate measure based on the speed of multiple links, which may vary widely. Thus a strategy is required to disaggregate the data to specific road links.
- *Language ambiguity of traffic description in tweets.* The expressions depicting traffic conditions are diverse, and may denote different speed values. Fig. 2 shows an example of word frequency distribution over the degree of congestion when people use congestion-related words. Meanwhile some words not directly related to traffic may also have strong implication to link speed, such as words complaining bad weather. Thus a linguistic model is required to capture the patterns between discrete descriptive words and continuous speed values.
- *Heterogeneity of multi-source data.* Cross-domain data sources have diverse properties and contain latent relations with road traffic speed. For example, tweets possess latent topics which cluster based on speed levels, and negative correlation existed between trajectory travel time and traffic speed of involving links. Therefore a unified framework is required to model these properties and aggregate the latent relations between heterogeneous data to predict speed synthetically.

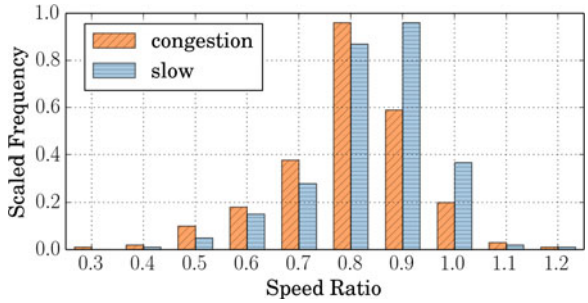


Fig. 2. The distribution of word frequencies when people use words "congestion" and "slow" to describe traffic, w.r.t. the ratio between current speed and a reference speed, which is defined by INRIX as the "uncongested free flow speed" for each road segment. X-axis denotes the speed ratio and Y-axis denotes the frequency scaled w.r.t. the biggest value.

1.3 Contributions

In spite of the good potential of these new-type data, to the best of our knowledge, the problem of road-level traffic speed prediction using multiple data sources has not been well explored before, especially with the aforementioned challenges. In this paper, we propose a unified statistical framework, entitled Topic Enhanced Gaussian Process Aggregation Model (TEGPAM) fusing multi-source data, which includes traditional speed sensing data, and new-type "sensing" data from social media and map services. The framework combines the location disaggregation model to decompose vague locations into specific links, the traffic topic model to handle the language ambiguity in tweets and the Gaussian Process model to capture the spatial correlation in traffic sensing data. Specifically, this paper makes the following contributions:

- *Integration of data from multiple cross-domain sources.* We implement the idea of improving traffic speed prediction by integrating speed sensing data with new-type traffic-related data, such as tweets and trajectories.
- *Formulation of the unified TEGPAM framework.* We propose a unified probabilistic framework TEGPAM that combines the disaggregation model, topic model with Gaussian Process model and is learned by variational methods and a stochastic EM algorithm.
- *Extensive experiments to validate the performance of the proposed method.* We validate our approach using real-world data collected from two large American cities. The extensive experiments show the effectiveness of TEGPAM, as well as the model efficiency and reliability.
- *Elaborate analyses of introduced traffic-related data.* We explore the impacts of different data sources, by decomposing TEGPAM into sub models and changing the combination ratio of datasets. Comparative experiments demonstrate the effectiveness of each data source.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 gives a preliminary to Gaussian Process. Section 4 defines the problem and presents the model design. Section 5 gives model inference. Section 6 analyzes the results of experiments on real data. Section 7 concludes the paper and suggests future directions.

2 Related Works

Traffic prediction problem can be broadly classified into short-term and long-term prediction [7], considering three main basic traffic measurements: traffic flow, an equivalent flow rate in vehicles; speed, mean of the observed vehicle speeds; lane occupancy, the percentage of time that the sensor is detecting vehicle presence. This paper focuses on the short-term traffic speed prediction combining multi-source heterogeneous data, which, as far as we know, has not been well explored before. This part gives a summary on short-term traffic speed prediction and the exploration on fusing multiple information sources.

Short-Term Traffic Speed Prediction. The presented methods can be classified into two categories:

1) *Parametric methods*, assume that traffic speed follows a probability distribution based on a fixed set of parameters. Time series analysis technique is applied in traffic speed prediction based on the periodicity of traffic speed during a day or a week. Auto-Regressive Moving Average (ARMA) model and Multivariate Spatial-Temporal Auto-Regressive (MSTAR) model are adopted to include dependency among observations from neighboring locations [5], [8]. Auto-Regressive Integrated Moving Average (ARIMA) time series methods are reviewed for modeling and forecasting vehicular traffic flow [9]. ARIMA and Winters exponential smoothing techniques are used to forecast urban freeway flow [3]. A single Space-Time Auto-Regressive Integrated Moving Average (STARIMA) model is proposed to describe the spatiotemporal evolution of traffic flow in an urban network [10], which is essentially a constrained Vector Autoregressive Moving Average (VARIMA) model [11] with constraints that reflect the topology of a spatial network and result in a drastic reduction in the number of parameters. A Generalized Space-Time ARIMA (GSTARIMA) method is proposed to extend ARIMA in spatial and temporal dimension and is more flexible because parameters are designed to vary per spacial location [12]. Kalman filter-based approaches are used and show advantages for on-line estimation of traffic flows [13], [14]. Markov logic network is used to simultaneously predict the congestion state [15]. A structured time series model is proposed in multivariate form for short-term traffic prediction [16].
2) *Non-parametric methods*, make no distribution assumptions and the number of parameters scales with the number of training data. K-nearest neighbor (KNN) nonparametric regression methods find the k-nearest neighbors using Euclidean distance and calculate the weight [2], [17], [18]. Neutral Networks (NNs) are trained to approximate any nonlinear function given adequate traffic sensing data and a proper network architecture [19], [20]. NNs have many derivatives for short-term prediction, such as back propagation neutral network with genetic algorithms [7] and wavelet networks [21]. Travel speed of each road segment is computed using the GPS trajectories by a context-aware matrix factorization approach [22]. To adaptively route a fleet of cooperative vehicles under the uncertain and dynamic road congestion conditions, a GP probabilistic model is proposed to capture the spatial and temporal relationships of travel speeds over road segments and temporal contexts [23], [24], especially with estimating the mean and covariance of the GP prior from the historical data. Geostatistical interpolation techniques named Kriging are proposed to capture spatial and temporal evolutions of traffic flows [25].

Traffic Modeling with Multi-Source Heterogeneous Data. Traffic modeling problems gain further insights through fusing heterogeneous data from multiple sources, e.g., road sensors, social media and floating cars, to handle external factors such as traffic accidents [26], [27], mobile sensors [28], [29] and weather [30], [31]. [30] reviews the literature on the impact of weather on traffic demand, traffic safety, and traffic flow relationships. A trajectory-based community discovery method is proposed [32], where the trajectory similarity is modeled by several types of kernels for different information markers (e.g., semantic properties of the locations and the movement velocity). The prediction problem of rents/returns bike number is tackled using multiple features, e.g., time and meteorology, as measures of similarity functions in multi-similarity-based inference model [33]. While [32] and [33] introduce different information sources as new features for computing the similarity, our work assumes the latent relations between these informations, and constructs a Bayesian generative process. As crowdsourcing data from a crowd of online social platform become more available, researchers begin utilizing social content to estimate traffic conditions. Twitter data are matched to detect traffic incidents in [26], [34], [35]. In [28], traffic anomaly detection uses crowd sensing with two forms of data, human mobility and social media, and the detected anomalies are described by mining representative terms from the social media that people posted when the anomaly happened. Few methods incorporate social media text data (e.g., Twitter data) to improve traffic speed prediction. [36] extends spatiotemporal GP in [24] to three dimensional topic-aware GP, where topics on road links are probabilistic modeled based on the user, space and time of tweets. [37] do not tackle the location uncertainty problem of tweets, because the inference of traffic status based on words of tweets only focuses on the average *regional* traffic flow, which is insufficient for predicting *road* speed.

3 Gaussian Process Preliminaries

Gaussian Processes (GPs) have been widely studied in many fields, such as spatio-temporal modeling [24], [38]. Given a set of road segments $\mathcal{S}$ under a specified time stamp, we spatially model the traffic speed of road segments via a function $f : \mathcal{S} \rightarrow \mathbb{R}_+$, which outputs the traffic speed for a given road link s.

Assume that f is sampled from a Gaussian process prior: $f(s) \sim \mathcal{GP}(\mu(s), k(s, s'))$, which is fully specified by the mean function and the covariance, or kernel, function

$$\mu(s) = \mathrm{E}[f(s)]$$

$$k(s, s') = \mathrm{E}[(f(s) - \mu(s))(f(s') - \mu(s'))].$$

An important property of GP is that if two sets of variables are jointly Gaussian, the conditional distribution of one set conditioned on the other is Gaussian, that is the basis to compute the posterior analytically [39].

Suppose that there are currently observed links $\boldsymbol{S} \subset \mathcal{S}$ with speed observations $\boldsymbol{V} = \{v_s, s \in \boldsymbol{S}\}$, where the traffic speed v_s for each link $s \in \mathcal{S}$ follows $v_s \sim \mathcal{N}(f(s), \sigma^2)$, where σ^2 is i.i.d. Gaussian noise. Then we can calculate the posterior distribution given the prior distribution with mean and kernel function, and the current observations $\boldsymbol{V}$, which is still a GP distribution

$$v_s|\boldsymbol{V}, \mu, k \sim \mathcal{GP}(\mu^{post}, k^{post}), \tag{1}$$

where

$$\mu^{post}(s) = \mu(s) + \boldsymbol{k}(\boldsymbol{S}, s)^\top [\boldsymbol{K} + \sigma^2 \boldsymbol{I}]^{-1} (\boldsymbol{V} - \boldsymbol{\mu}) \tag{2}$$

$$k^{post}(s, s') = k(s, s') - \boldsymbol{k}(\boldsymbol{S}, s)^\top [\boldsymbol{K} + \sigma^2 \boldsymbol{I}]^{-1} \boldsymbol{k}(\boldsymbol{S}, s'), \tag{3}$$

where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{K}$ is the kernel Gram matrix, which are generated through historical speed records at observed links $\boldsymbol{S}$

$$\boldsymbol{\mu} = [\mu(s)]_{s \in \boldsymbol{S}} \in \mathbb{R}^{|\boldsymbol{S}|}$$

$$\boldsymbol{K} = [k(s, s')]_{s, s' \in \boldsymbol{S}} \in \mathbb{R}^{|\boldsymbol{S}| \times |\boldsymbol{S}|}.$$

Column vector $\boldsymbol{k}(\boldsymbol{S}, s)$ is the kernel values between $s \in \mathcal{S}$ and every current observations in $\boldsymbol{S}$

$$\boldsymbol{k}(\boldsymbol{S}, s) = [k(s_1, s), \ldots, k(s_i, s), \ldots, k(s_{|\boldsymbol{S}|}, s)]^\top_{s_i \in \boldsymbol{S}} \in \mathbb{R}^{|\boldsymbol{S}|}.$$

Eq. (2) implies that the posterior mean $\mu^{post}(s)$ is determined by its prior mean $\mu(s)$ and the deviation between the historical observations and their prior means. If the positive covariance $k(s, s')$ between road links s and s' is high, the current observation of s' will have more impacts on $\mu^{post}(s)$ with $(v_{s'} - \mu(s'))$. Eq. (3) presents the property that the posterior covariance $k^{post}(s, s')$ between s and s' will decrease if we have more current observations related to s and s'. Meanwhile, the posterior $k^{post}(s, s')$ decreases faster with high $k(s, s')$ between s and s'.

Essentially, the kernel function k, generated from historical observations depicting the relation between road links, captures the spatial correlation of road network. If the covariance of two road links s and s' is high, we may intuitively infer that they are close in the network structure.

4 Model Design

This section begins by formalizing the speed prediction problem in Section 4.1. Then we introduce three models from Sections 4.2, 4.3, and 4.4 to tackle the challenges aforementioned in the introduction, i.e., a disaggregation model for location uncertainty in tweet and trajectory data, a traffic topic model for tweet language ambiguity and a GP model for capturing the spatial correlation of speed sensing data. Section 4.5 integrates three models dealing with different information source into a novel probabilistic model, named TEGPAM, under the Bayesian framework.

4.1 Problem Formulation

Consider a road network denoted by $\mathcal{S} = \{1, \ldots, S\}$ containing S road links, and a time duration denoted by $\mathcal{T} = \{1, \ldots, T\}$ containing T time stamps. Our goal is to predict traffic speed of some links at a certain time stamp using the past and current observations from multiple data sources, including traffic sensing data, Tweets and trajectories. The terms and formal definitions used throughout this paper are listed as follows.

Traffic Condition. Road traffic condition is described by two variables: continuous traffic speed and binary traffic status. The speed at time $t \in \mathcal{T}$ and link $s \in \mathcal{S}$ is denoted by $v_{t,s} \in \mathbb{R}$, and the status is denoted by $x_{t,s} \in \{0, 1\}$, where 1 refers to congested traffic and 0 refers to normal traffic. Denote $\boldsymbol{S}_t \subset \mathcal{S}$ as speed-observed links at time t, and $\boldsymbol{V}_t = \{v_{t,s}, s \in \boldsymbol{S}_t\}$ correspondingly as observations.

Traffic Related Tweet. A tweet d is depicted as a tuple $(t, \boldsymbol{S}_d, \boldsymbol{w}_d)$, where $t \in \mathcal{T}$ denotes the time that the tweet is posted, $\boldsymbol{S}_d \subset \mathcal{S}$ is the set of possible links implied by the tweet text, and $\boldsymbol{w}_d = \{w_1, \ldots, w_{N_d}\}$ denotes the sequence of traffic related words in the tweet text. Note that $\boldsymbol{S}_d$ will contain multiple links if the location mentioned in tweet d is not specific, such as a street name containing multiple road segments without finer information. Denote $\mathcal{D}_t = \{d_1, \ldots, d_{D_t}\}$ as the traffic related tweet set at time t.

Trajectory and Travel Time. A trajectory or path p is denoted as a tuple $(t, \boldsymbol{S}_p, c_p)$, where $t \in \mathcal{T}$ is the time when the trajectory is generated given an OD pair, $\boldsymbol{S}_p \subset \mathcal{S}$ represents consecutively connected links in the trajectory and $c_p \in \mathbb{R}$ is the time cost traveling through the trajectory. Denote $\mathcal{P}_t = \{p_1, \ldots, p_{P_t}\}$ as the trajectory set at time t, then $\mathcal{C}_t = \{c_1, \ldots, c_{P_t}\}$ is the corresponding travel time cost set. The road length is represented as $\boldsymbol{L} \in \mathbb{R}^S$ with each component l_s equal to the road length of link $s \in \mathcal{S}$.

Problem Formulation (Road Traffic Speed Prediction Fusing Multi-source Data). Consider a set of road links $\mathcal{S}$ in the time duration of $\mathcal{T}$, given a traffic related tweet corpus $\mathcal{D} = \{\mathcal{D}_t | t \in \mathcal{T}\}$, a set of travel times $\mathcal{C} = \{\mathcal{C}_t | t \in \mathcal{T}\}$ with known road length $\boldsymbol{L}$, and speed observations $\boldsymbol{V} = \{\boldsymbol{V}_t | t \in \mathcal{T}\}$, our problem is to predict these unobserved traffic speed variables $\{v_{s,t} | t \in \mathcal{T}, s \in \mathcal{S} - \boldsymbol{S}_t\}$.

4.2 Location Disaggregation Model

To handle the challenge of location uncertainty of new-type data, this section presents a disaggregation strategy to map the low-resolution data, which are tweets and trajectories, into specific road links.

4.2.1 Disaggregation of Tweets

Since only 1 percent of tweets have geo-coordinates [40], most location information are extracted from tweet text by mapping road names or alias. However, the difficulty lies in the low-resolution of road names mentioned in tweet contents, for example, if d says "*E 32nd St. is stuck*", each link in the street has a probability to be congested. Based on the observation, we assume that if a tweet expresses congestion, one or more links involved in that location will be congested.

Recall that a tweet d is represented by $(t, \boldsymbol{S}_d, \boldsymbol{w}_d)$, where $\boldsymbol{S}_d$ is a set of possible links geocoded by d, and the traffic status expressed by tweet d at time t is denoted by $h(x_{t,s}) \in \{0, 1\}$. We define a rule to express the assumption as

$$h(x_{t,s}) = \begin{cases} 1 & \text{if } \exists x_{t,s} = 1, s \in \boldsymbol{S}_d, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Denote $\boldsymbol{x}_t = \{x_{t,s}, s \in \mathcal{S}\}$ as statuses over all the links at time t, $\mathrm{ID}_{t,d} = \{0,1\}^S$ as indicative function of links involved in tweet d, and the indexes of those nonzero components constitute $\boldsymbol{S}_d$. Eq. (4) is transformed to

$$h(\boldsymbol{x}_t) = 1 - \prod_{s=1}^{S}(1 - \mathrm{ID}_{t,d,s} x_{t,s}), \tag{5}$$

where $h(\boldsymbol{x}_t)$ is 1, when the traffic status expressed by the tweet is congestion, as long as the status value $x_{t,s}$ of at least one component in $\mathrm{ID}_{t,d,s}$ is 1.

4.2.2 Disaggregation of Trajectory Travel Time

The time traveling through a trajectory is a sum of time cost on each link. Recall that $v_{t,s}$ is the traffic speed at time t and $l_s \in \boldsymbol{L}$ is the road length of link s. By denoting $\boldsymbol{v}_t = \{v_{t,s}, s \in \mathcal{S}\}$ as speeds over all links at time t, and $\mathrm{IP}_{t,p} = \{0,1\}^S$ as indicative function of links passed by trajectory p, we define a function as

$$y(\boldsymbol{v}_t, \boldsymbol{L}) = \sum_{s=1}^{S} \frac{1}{v_{t,s}} \mathrm{IP}_{t,p,s} l_s, \tag{6}$$

to disaggregate travel time of a trajectory into the speed of specific links. Assume that travel time is corrupted by i.i.d. Gaussian noise, we define the distribution of travel time cost $c_{t,p}$ as: $c_{t,p}|\boldsymbol{v}_t, \boldsymbol{L} \sim \mathcal{N}(y(\boldsymbol{v}_t, \boldsymbol{L}), \sigma^2)$.

4.3 Traffic Topic Model

To address the challenge of language ambiguity and capture the traffic description in tweets, a traffic topic model is proposed, similar to [41]. Fast or slow speed $v_{t,s}$ indicates different traffic status $x_{t,s}$, and the binary traffic status is first modeled by a Bernoulli distribution using a Logistic function

$$x_{t,s}|v_{t,s} \sim Ber(g(v_{t,s})), g(v_{t,s}) = \frac{1}{1 + e^{\delta(v_{t,s}+\beta)}}, \tag{7}$$

to map numerical value $v_{t,s} \in \mathbb{R}$ to binary value $x_{t,s}$, so that we can distinguish the traffic speed between two traffic status: congested or normal.

Different status will trigger different tweet topic. Assume that at time t, each tweet d is assigned to a single topic $z_{t,d} \in \{1, \ldots, K\}$ and K refers to the number of topics. We model tweet topic as a multinomial distribution [42] conditioned on the traffic status expressed by the tweet

$$z_{t,d}|\boldsymbol{x}_t \sim \mathrm{Multi}(\boldsymbol{\psi}_{h(\boldsymbol{x}_t)}), \tag{8}$$

where $\boldsymbol{\psi}_{h(\boldsymbol{x}_t)} \in [0,1]^K$, and recall that $h(\boldsymbol{x}_t) = \{0,1\}$, calculated by Eq. (5), is the status of the tweet expression. Conditioned on traffic status expressed by this tweet $h(\boldsymbol{x}_t)$, there are two distributions over topic modeling free or congested traffic conditions: Multi($\boldsymbol{\psi}_0$) and Multi($\boldsymbol{\psi}_1$). Given a traffic related vocabulary, each word of the tweet text is sampled from a multinomial distribution depended on the tweet topic

$$w_{t,d,n}|z_{t,d} \sim \mathrm{Multi}(\boldsymbol{\phi}_{z_{t,d}}), \tag{9}$$

where $\boldsymbol{\phi}_{z_{t,d}} \in [0,1]^M$, and M refers to the vocabulary size.

4.4 Traffic Speed Gaussian Process Model

Our goal is to model the spatial correlation of road speed, so given recent speed observations, we can infer the speed of unobserved links. For a set of links $\mathcal{S}$, we sample a function $f : \mathcal{S} \to \mathbb{R}^S$, from the GP prior: $f(s) \sim \mathcal{GP}(\mu(s), k(s,s'))$, which outputs the traffic speeds for all road links.

Since the spatial correlation varies with time stamps, GPs are assumed to be independent temporally, for example GPs are different at different time-of-day: $f_t(s) \sim \mathcal{GP}(\mu_t(s), k_t(s,s'))$. Recall that $\boldsymbol{V}_t = \{v_{t,s}, s \in \boldsymbol{S}_t\}$ are observations at time $t \in \mathcal{T}$, then a GP posterior distribution given observations $\boldsymbol{V}_t$ is obtained analytically: $f_t(s)|\boldsymbol{V}_t \sim \mathcal{GP}(\mu_t^{post}(s), k_t^{post}(s,s'))$. Suppose $\boldsymbol{v}_t \in \mathbb{R}^S$ is denoted as the speed for all links at time t, and is corrupted by Gaussian noise: $\boldsymbol{v}_t|f_t \sim \mathcal{N}(f_t, \varepsilon^2 \boldsymbol{I})$, then the predictive distribution is

$$\boldsymbol{v}_t|\boldsymbol{V}_t, \mu_t, k_t, \varepsilon \sim \mathcal{N}(\mu_t^{post}, k_t^{post} + \varepsilon^2 \boldsymbol{I}), \tag{10}$$

with mean and covariance, or kernel

$$\begin{aligned} \mu_t^{post}(s) &= \mu_t(s) + k(\boldsymbol{S}_t, s)^\top [\boldsymbol{K}_t + \varepsilon^2 \boldsymbol{I}]^{-1} (\boldsymbol{V}_t - \boldsymbol{\mu}_t) \\ k_t^{post}(s,s') &= k_t(s,s') - k(\boldsymbol{S}_t, s)^\top [\boldsymbol{K}_t + \varepsilon^2 \boldsymbol{I}]^{-1} k(\boldsymbol{S}_t, s'), \end{aligned} \tag{11}$$

where $k(\boldsymbol{S}_t, s) = [k(s_1, s), \ldots, k(s_{|\boldsymbol{S}_t|}, s)]^\top$ is the kernel values between s and every observed link in $\boldsymbol{S}_t$, $\boldsymbol{\mu}_t$ is the mean vector and $\boldsymbol{K}_t$ is the kernel Gram matrix generated at observed links $\boldsymbol{S}_t$ at a time context t, which is time-of-day, or more complex, day-of-week.

To estimate the mean μ_t and kernel k_t for each temporal context, one strategy is using a compact functional form, for example a widely-used squared exponential function $k(s,s') = \sigma_1^2 \exp(-d^*(s,s')^2/\sigma_2^2)$, where $d^*(s,s')$ is a distance function, such as a Euclidean distance embedded by multi-dimensional scaling [43]. However, designing suitable function forms that well approximate the equation can be difficult. Here we adopt another data-driven strategy estimating mean and covariance directly from historical traffic speed data [24], correspond to "true" mean and covariance. The advantage is that a suitable covariance function form is not required. For a temporal context t, suppose time-of-day is denoted as $\pi(t)$, then the mean and variance at each link s are estimated as

$$\begin{aligned} \mu_t(s) &= \mathrm{mean}\{v_{t',s} : (\pi(t') = \pi(t)) \wedge (s \in \boldsymbol{S}_{t'})\}, \\ k_t(s,s') &= \mathrm{mean}\{(v_{t',s} - \mu_{t'}(s))(v_{t',s'} - \mu_{t'}(s')) : \\ &\quad (\pi(t') = \pi(t)) \wedge (s, s' \in \boldsymbol{S}_{t'})\} \end{aligned} \tag{12}$$

where $v_{t',s}$ is the historical traffic speed on time-of-day $\pi(t') = \pi(t)$ and road s, while $v_{t',s}$ and $v_{t',s'}$ are belong to the same temporal context.

Note that the link mean $\boldsymbol{\mu} = \{\boldsymbol{\mu}_t\}_{t=1}^T$ and link-link kernel matrix $\boldsymbol{K} = \{\boldsymbol{K}_t\}_{t=1}^T$ for all the temporal contexts can be precomputed since the links and time stamps form a finite set. Essentially, the magnitude of $k_t(s,s')$ is high when s and s' have high covariance, and observing one speed will better help predicting the other.

However Eq. (12) should be performed on sufficient historical data so each link more or less has observations, in order to achieve a good approximation. For cases where speed sensor data are extremely sparse and some links have no data reported at all, data imputation techniques are

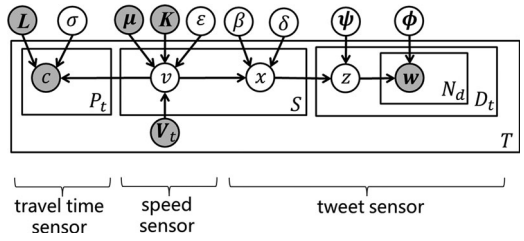


Fig. 3. Graphical model for TEGPAM, with three parts dealing with the three aforementioned data sources.

adopted for preprocessing [44]. The issue of sparsity is further discussed in Section 6.2.3.

4.5 TEGPAM

Integrating the components introduced in the above sections completes the design of the new probabilistic model, named the Topic-Enhanced Gaussian Process Aggregation Model (TEGPAM). Fig. 3 gives the graphical representation of our model.

For each time stamp t, the model (shown in Fig. 3) assumes the following generative process:

1. For each link s in $\mathcal{S}$:
   - Draw a speed $v_{t,s}|\boldsymbol{V}_t, \mu_t, k_t, \varepsilon \sim \mathcal{N}(\mu_t^{post}(s), k_t^{post}(s,s') + \varepsilon^2 \boldsymbol{I})$
   - Choose a traffic status $x_{t,s}|v_{t,s} \sim Ber(g(v_{t,s}, \beta, \delta))$, where $g(v_{t,s})$ is calculated by Eq. (7)
2. For each tweet d in $\mathcal{D}_t$:
   - Choose a topic $z_{t,d}|x_{t,s} \sim Multi(\boldsymbol{\psi}_{h(x_{t,s})})$, where $h(x_{t,s})$ is calculated by Eq. (5)
   - For each of the N words $w_{t,d,n}$:
     - Choose a word $w_{t,d,n}|z_{t,d} \sim Multi(\boldsymbol{\phi}_{z_{t,d}})$
3. For each path p in $\mathcal{P}_t$:
   - Choose travel time $c_{t,p}|v_{t,s}, \boldsymbol{L} \sim \mathcal{N}(y(v_{t,s}, L), \sigma^2)$, where $y(v_{t,s}, L)$ is calculated by Eq. (6)

5 Learning TEGPAM

Given GP prior mean $\boldsymbol{\mu}$ and kernel $\boldsymbol{K}$ directly precomputed from historical traffic speed data as introduced in Section 4.4, speed observations $\boldsymbol{V} = \{\boldsymbol{V}_t\}_{t=1}^T$, road length vector $\boldsymbol{L}$ and model parameters $\Theta = \{\sigma, \varepsilon, \beta, \delta, \boldsymbol{\psi}, \boldsymbol{\phi}\}$, the joint distribution of travel times $\mathcal{C} = \{\mathcal{C}_t\}_{t=1}^T$, traffic speeds $\mathcal{V} = \{v_{t,s} | s \in \mathcal{S}\}_{t=1}^T$, traffic statuses $\mathcal{X} = \{x_{t,s} | s \in \mathcal{S}\}_{t=1}^T$, topics $\mathcal{Z} = \{z_{t,d} | d \in \mathcal{D}_t\}_{t=1}^T$ and word corpus $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^T$ is

$$\begin{aligned} &p(\mathcal{C}, \mathcal{V}, \mathcal{X}, \mathcal{Z}, \mathcal{D} | \Theta, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{K}, \boldsymbol{L}) \\ &= \prod_{t=1}^{T} \left(\prod_{s=1}^{S} p(v_{t,s} | \mu_t^{post}(s), k_t^{post}(s,s') + \varepsilon^2 \boldsymbol{I}) p(x_{t,s} | v_{t,s}, \beta, \delta) \right) \\ &\times \left(\prod_{d=1}^{D_t} p(z_{t,d} | \boldsymbol{x}_t, \boldsymbol{\psi}) p(w_{t,d} | z_{t,d}, \boldsymbol{\phi}) \right) \times \left(\prod_{p=1}^{P_t} p(c_{t,p} | \boldsymbol{v}_t, \boldsymbol{L}, \sigma) \right), \end{aligned} \tag{13}$$

by taking the marginal distribution of which, the problem is formalized as a maximum likelihood estimation (MLE)

$$\max_{\Theta} \ \log p(\mathcal{C}, \mathcal{D} | \Theta, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{K}, \boldsymbol{L}), \tag{14}$$

then the inferential problem using TEGPAM to predict speed is to compute the posterior distribution $p(\mathcal{V} | \Theta, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{K}, \boldsymbol{L})$, which, however, is intractable due to the coupling between the latent variables. We use a variational inference approach [45] to approximate the posterior and obtain the lower bound of the log likelihood.

5.1 TEGPAM Inference

The goal of learning TEGPAM is to find parameters Θ that maximize the likelihood. Consider a family of fully factored variational distribution as

$$q(\mathcal{V}, \mathcal{X}, \mathcal{Z}) = \prod_{t=1}^{T} q(\boldsymbol{v}_t | \boldsymbol{\lambda}, \nu^2 \boldsymbol{I}) \prod_{s=1}^{S} q(x_{t,s} | \eta_s) \prod_{d=1}^{D_t} q(z_{t,d} | \boldsymbol{\gamma}_d), \tag{15}$$

where $\boldsymbol{\lambda}$ and $\nu^2 \boldsymbol{I}$ are mean and covariance for the multivariate Gaussian distribution since speeds at the same time stamp are spatially correlated according to the GP model in Section 4.4, $\{\eta_s\}_{s=1}^S$ are the Bernoulli parameters and $\{\boldsymbol{\gamma}_d\}_{d=1}^{D_t}$ are the multinomial parameters.

Using Jensen's inequality [45], the variational lower bound can be obtained

$$\begin{aligned} \mathcal{L} &= \log p(\mathcal{C}, \mathcal{D} | \Theta, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{K}, \boldsymbol{L}) \\ &\geq \mathrm{E}_q[\log p(\mathcal{C}, \mathcal{V}, \mathcal{X}, \mathcal{Z}, \mathcal{D} | \Theta, \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{K}, \boldsymbol{L})] + \mathrm{H}(q) \\ &= \mathrm{E}_{\mathcal{V}}[\log p(\mathcal{V} | \boldsymbol{V}, \boldsymbol{\mu}, \boldsymbol{K}, \varepsilon)] + \mathrm{E}_{\mathcal{V},\mathcal{X}}[\log p(\mathcal{X} | \mathcal{V}, \beta, \delta)] \\ &\quad + \mathrm{E}_{\mathcal{X},\mathcal{Z}}[\log p(\mathcal{Z} | \mathcal{X}, \boldsymbol{\psi})] + \mathrm{E}_{\mathcal{Z}}[\log p(\mathcal{D} | \mathcal{Z}, \boldsymbol{\phi})] \\ &\quad + \mathrm{E}_{\mathcal{V}}[\log p(\mathcal{C} | \mathcal{V}, \boldsymbol{L}, \sigma)] + \mathrm{H}(q) \end{aligned} \tag{16}$$

where each term is shown in Table 1. Note that the posterior mean $\boldsymbol{\mu}^{post}$ and variance $\boldsymbol{K}^{post}$ are analytically obtained by Eq. (11) using the observations $\boldsymbol{V}$, prior mean $\boldsymbol{\mu}$ and variance $\boldsymbol{K}$, at observed links for each time stamp.

Because both the node x for topic model and c for disaggregation model are not conditionally conjugate, some terms cannot be computed in closed form, so new variational parameters $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$ are introduced by Taylor expansion [46], [47]. The detailed calculation of the expectations with Taylor expansion are presented in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2017.2718525, while the calculation of other terms are omitted, which are direct and basic integral operations.

5.2 Parameter Updates

We optimize the log likelihood lower bound in Eq. (16), by updating the variational parameters and the model parameters iteratively. We use variational expectation-maximization (EM) algorithm here. Note that variational EM cannot guarantee the increase of log-marginal likelihood in each iteration. In E-step, we maximize the lower bound with respect to the variational parameters $\{\lambda, \nu^2, \boldsymbol{\gamma}, \eta, \boldsymbol{\zeta}, \boldsymbol{\xi}\}$. In M-step, we maximize the bound with respect to the model parameters $\{\sigma, \varepsilon, \beta, \delta, \boldsymbol{\psi}, \boldsymbol{\phi}\}$. E-step and M-step are repeated till the bound on the likelihood converges.

5.2.1 In E-Step

Updating $\boldsymbol{\zeta}$ and $\boldsymbol{\xi}$. The newly introduced variational parameters for Taylor expansion is given by

TABLE 1
The Expectation of Each Term in Lower Bound

Term	Expression
$\mathrm{E}_{\mathcal{V}}[\log p(\mathcal{V}\|\boldsymbol{\mu}^{post}, \boldsymbol{K}^{post}, \varepsilon)]$	$-\frac{T}{2}[\log\|\boldsymbol{K}_{new}\| + \nu^2\mathrm{tr}(\boldsymbol{K}_{new}^{-1}) + (\boldsymbol{\lambda}-\boldsymbol{\mu}^{post})^\top \boldsymbol{K}_{new}^{-1}(\boldsymbol{\lambda}-\boldsymbol{\mu}^{post})]$
$\mathrm{E}_{\mathcal{V},\mathcal{X}}[\log p(\mathcal{X}\|\mathcal{V},\beta,\delta)]^1$	$T\sum_{s=1}^{S}(\delta(1-\eta_s)(\beta+\lambda_s)) - (\frac{1}{\zeta_s} + \boldsymbol{e}^\top\boldsymbol{E} - 1 + \log\zeta_s)$
$\mathrm{E}_{\mathcal{X},\mathcal{Z}}[\log p(\mathcal{Z}\|\mathcal{X},\boldsymbol{\psi})]$	$\sum_{t=1}^{T}\sum_{d=1}^{D_t}\sum_{j=1}^{K}\gamma_{d,j}(\log\psi_{1,j} + \log\frac{\psi_{2,j}}{\psi_{1,j}}\prod_{s=1}^{S}(1-\mathrm{ID}_{t,d,s}\eta_s))$
$\mathrm{E}_{\mathcal{Z}}[\log p(\mathcal{D}\|\mathcal{Z},\boldsymbol{\phi})]$	$\sum_{t=1}^{T}\sum_{d=1}^{D_t}\sum_{j=1}^{K}\sum_{k=1}^{M} wn_{t,d}^{(k)}\gamma_{d,j}\log\phi_{j,k}$
$\mathrm{E}_{\mathcal{V}}[\log p(\mathcal{C}\|\mathcal{V},\boldsymbol{L},\sigma)]^1$	$\sum_{t=1}^{T}\sum_{p=1}^{P_t} -\frac{1}{2}\log\sigma^2 - \frac{1}{2\sigma^2}[m(c_{t,p})^2 + 2m(c_{t,p})\boldsymbol{\lambda}^\top\mathrm{A} + \boldsymbol{\lambda}^\top(\mathrm{AA}^\top)\boldsymbol{\lambda} + \mathrm{tr}(\mathrm{AA}^\top(\nu^2\mathrm{I}))]$
$\mathrm{H}(q)$	$\sum_{t=1}^{T}[-\frac{1}{2}\log\|\nu^2\mathrm{I}\| - \sum_{d=1}^{D_t}\sum_{j=1}^{K}\gamma_{d,j}\log\gamma_{d,j} - \sum_{s=1}^{S}(\eta_s\log\eta_s + (1-\eta_s)\log(1-\eta_s))]$

[1] *where* $\boldsymbol{K}_{new} = \boldsymbol{K}^{post} + \varepsilon^2\boldsymbol{I}$, $\boldsymbol{e} = [1]_{s=1}^{S}$, $\boldsymbol{E} = \left[\frac{e^{\delta(\beta+\lambda_s)+\frac{1}{2}\delta^2\nu^2}}{\zeta_s}\right]_{s=1}^{S}$ *and* $m(c_{t,p}) = c_{t,p} + \sum_{s=1}^{S}\frac{1}{\xi_s}$, *vector* $\mathrm{A} = \left[\frac{1}{\xi_s^2\mathrm{IP}_{t,p,s}l_s}\right]_{s=1}^{S}$.

$$\zeta_s = 1 + \exp\left\{\delta(\beta+\lambda_s) + \frac{1}{2}\delta^2\nu^2\right\} \tag{17}$$

$$\xi_s = -\frac{\lambda_s}{\mathrm{IP}_{t,p,s}l_s}. \tag{18}$$

Updating $\boldsymbol{\eta}$. The parameters for Bernoulli variational distribution is given by

$$\eta_s = 1/\exp\left\{\frac{1}{T}\sum_{t=1}^{T}\left[\delta(\beta+\lambda_s) + \sum_{d=1}^{D_t}\sum_{j=1}^{C}\gamma_{d,j}\log\frac{\psi_{2,j}}{\psi_{1,j}}\mathrm{ID}_{t,d,s}\prod_{s'\neq s}^{S}(1-\mathrm{ID}_{t,d,s'}\eta_{s'})\right]\right\} \tag{19}$$

Updating $\boldsymbol{\gamma}$. Following [41], the parameters related to topics are updated by

$$\gamma_{d,j} \propto \exp\left\{\sum_{k=1}^{M} wn_{t,d}^{(k)}\log\phi_{j,k} + \frac{1}{T}\sum_{t=1}^{T}\log\psi_{1,j} + \log\frac{\psi_{2,j}}{\psi_{1,j}}\prod_{s=1}^{S}(1-\mathrm{ID}_{t,d,s}\eta_s)\right\} \tag{20}$$

5.2.2 In M-Step

Updating $\boldsymbol{\psi}$ *and* $\boldsymbol{\phi}$. Following [41], the updating functions are

$$\psi_{1,j} \propto \frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D_t}\gamma_{d,j}\left(1-\prod_{s=1}^{S}(1-\mathrm{ID}_{t,d,s}\eta_s)\right)$$
$$\psi_{2,j} \propto \frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D_t}\gamma_{d,j}\prod_{s=1}^{S}(1-\mathrm{ID}_{t,d,s}\eta_s) \tag{21}$$

$$\phi_{j,k} \propto \frac{1}{T}\sum_{t=1}^{T}\sum_{d=1}^{D_t} wn_{t,d}^{(k)}\gamma_{d,j}. \tag{22}$$

Updating σ^2. The noise parameters are updated by

$$\sigma^2 = \frac{1}{T\times P_t}\sum_{t=1}^{T}\sum_{p=1}^{P_t}\left[\left(c_{t,p} + 2\sum_{s=1}^{S}\frac{1}{\xi_s}\right)^2 + 2\left(c_{t,p} + 2\sum_{s=1}^{S}\frac{1}{\xi_s}\right)\boldsymbol{\lambda}^\top\mathrm{A} + \boldsymbol{\lambda}^\top(\mathrm{AA}^\top)\boldsymbol{\lambda} + \mathrm{tr}(\mathrm{AA}^\top(\nu^2\mathrm{I}))\right]. \tag{23}$$

The Gaussian parameters $\boldsymbol{\lambda}, \nu^2$ and Logistic parameters β, δ are optimized using Newton method in Appendix B, available in the online supplemental material, and the Gaussian noise ε^2 is empirically tuned because it cannot be updated in closed form.

Eqs. (19) and (20) have intuitive interpretations. The multinomial update for $\boldsymbol{\gamma}$ is obtained by Bayes' theorem, $p(\mathcal{Z}|\mathcal{D},\mathcal{X}) \propto p(\mathcal{D}|\mathcal{Z})p(\mathcal{Z}|\mathcal{X})$, where $p(\mathcal{Z}|\mathcal{X})$ is approximated by the exponential of $\mathrm{E}_{\mathcal{Z},\mathcal{X}}[\log p(\mathcal{Z}|\mathcal{X})]$ under the variational distribution. Likewise, the multinomial update for $\boldsymbol{\eta}$ is obtained from $p(\mathcal{X}|\mathcal{V},\mathcal{Z}) \propto p(\mathcal{X}|\mathcal{V})p(\mathcal{Z}|\mathcal{X})$, where $p(\mathcal{X}|\mathcal{V})$ is approximated by the exponential of $\mathrm{E}_{\mathcal{V},\mathcal{X}}[\log p(\mathcal{X}|\mathcal{V})]$ under the variational distribution. Thus the variational distribution $q(\mathcal{V}|\boldsymbol{\lambda}, \nu^2\boldsymbol{I})$ can be viewed as an approximation to the posterior distribution $p(\mathcal{V}|\Theta, V, \boldsymbol{\mu}, K, \boldsymbol{L})$.

5.3 Algorithm and Complexity

Algorithm 1 trains model parameters using EM process. Algorithm 2 tests the model by optimizing variational parameters and approximating posterior distributions.

Algorithm 1. TEGPAM Training

Input: $V, \mathcal{D}, \mathcal{C}, L$ /* link speed observations, traffic-related corpus, travel times, link lengths */
Output: Θ /*model parameters */
1: Estimate $\boldsymbol{\mu}, \boldsymbol{K}$ of GP prior $\leftarrow$ Eq. (12)
2: Initialize model parameters $\Theta^0 = \{\sigma, \varepsilon, \beta, \delta, \boldsymbol{\psi}, \boldsymbol{\phi}\}$ $\leftarrow \{0, 0, \mathrm{median}\{\boldsymbol{V}\}, -1, [1/K]^K, [1/M]^M\}$
3: Initialize variational parameters $\Gamma^0 = \{\lambda, \nu, \boldsymbol{\gamma}, \boldsymbol{\eta}, \boldsymbol{\zeta}, \boldsymbol{\xi}\}$ $\leftarrow \{[\mathrm{mean}\{\boldsymbol{V}\}]^S, \mathrm{cov}\{\boldsymbol{V}\}, [1/K]^K, [0.5]^S,$ Eqs. (17) and (18)$\}$
4: Calculate the initial lower bound $\mathcal{L}^0 \leftarrow$ Eq. (16)
5: **repeat**
6: **repeat**
7: E-step: Fix model parameters Θ and update variational parameters Γ.
8: **until** *convergence*;
9: **repeat**
10: M-step: Fix variational parameters Γ and update model parameters Θ.
11: **until** *convergence*;
12: **until** *lower bound* $\mathcal{L}^{n+1}$ converges
13: **return** *model* parameters Θ

Algorithm 2. TEGPAM Test

Input: $\Theta, L, V_{t^*}, \mathcal{D}_{t^*}, \mathcal{C}_{t^*}$ /* learned model parameters, link lengths, link speed observations at time t^*, traffic-related corpus at time t^*, travel times at time t^* */
Output: λ, ν^2 /*posterior Gaussian */
1: **repeat**
2: Update variational parameters Γ
3: **until** *convergence*;
4: **return** *variational parameters* Γ

Note that the updates of GP related parameters require a time cost of $O(S^3)$ with storage demand of $O(S^2)$ incurred by the reversion of covariance matrix, which is unacceptable for large scale networks. There have been many attempts to make sparse approximations to the full GP with a low-rank matrix. One simple way is to select a subset U (active set) of size $U \ll S$ from the entire training set[43] which reduced the complexity to $O(U^3)$, and the selection can be performed through an iterative greedy algorithm on some sort of information criterion [48]. Another way is to select inducing variables (pseudo targets) $\bar{U}$ based on the assumption that training and test sets are conditionally independent given $\bar{U}$[49], and those inducing variables can be found by maximum likelihood (ML). This method leads to a computational complexity of $O(U^2S)$.

6 Experiments

Experiments on traffic speed prediction of two large American cities are conducted to evaluate the following performance indicators: prediction accuracy, model efficiency, and model stability. This section is organized as follows: Section 6.1 introduces the experiment setting, including datasets, benchmark methods and predictive metrics. Section 6.2 validates our model of the overall performance regarding the prediction accuracy and efficiency. Section 6.3 provides an elaborate evaluation of the TEGPAM's effectiveness when applied to different data combinations. Section 6.4 discusses the model efficiency, and two factors of model stability: 1) sensitivity to parameters and 2) reliability on noisy tweets.

6.1 Experiment Setting

6.1.1 Datasets

We obtain three data sources for road traffic speed prediction: 1) *Traffic speed data*. INRIX database [50] provides traffic speeds for each road link at a 5-minute rate, from June 1, 2013 to March 31, 2014, across two cities: Washington D.C. and Philadelphia. 2) *Trajectory data*. Trajectories are generated from INRIX database at a 5-minute rate. Given a random OD pair, we synthesize a trajectory by computing the shortest path between them (i.e., using Johnson's algorithm [51]). With the length and speed information of links from INRIX, the travel time of this trajectory is obtained by adding the time of each link up and corrupting it with a Gaussian noise. 3) *Twitter data*. Tweets in the same time period and cities are collected via the Twitter REST search API. Traffic related tweets are preliminarily extracted by matching at least one term of a predefined vocabulary developed by domain experts, which included terms like "traffic", "accident", "stuck", "crash", etc, then further classified and filtered using an SVM classifier that was trained based on manually labeled 10,000 tweets (50 percent positive and 50 percent negative tweets). With road records containing the geo-coordinates, names and aliases, we geocode tweets to road links by matching their geo-tag and text content to the front end of those links, which corresponds to the driving out direction and is denoted as Head. Different driving directions are denoted as different road links. After geocoding, there are 5 major roads with 35 road links mentioned in the Philadelphia twitter data, and 8 major roads with 44 links in Washington D.C. respectively. Details of each data source are show in Table 2.

TABLE 2
Dataset Size for Each City

CityData	# total links	# trajectories	# tweets
Philadelphia	847,714	3,888,000,000	333,764
Washington D.C.	1,286,284	7,776,000,000	404,872

6.1.2 Benchmark Methods

To validate the performance of our approach fusing multiple data sources, particularly, to explore the impact of each data source, this section designs comparative methods based on the decomposition of our approach TEGPAM.

To make the expression clear, label data sources {traffic speed, trajectory, twitter} as $\{1, 2, 3\}$. Denote M-i as the model excluding the data sources $i \subset \{1, 2, 3\}$. Then we design sub models in terms of different data combinations:

TEGPAM: our full model introduced in Sections 5 and 6, using traffic speed, travel time and twitter data. The model is learned by variational inference.
M-1: trajectory and twitter based model, without incorporating traffic speed sensing data.
M-2: traffic speed and twitter based model, without handling trajectory data.
M-3: traffic speed and trajectory based model, without handling twitter data.
M-13: trajectory based model.
M-23: traffic speed based model, which is essentially a simplification of Gaussian Process Dynamic Congestion Model (GPDCM) in [23].

We infer the parameters of those models based on the same distribution assumptions, and we train parameters under the same settings.

Baseline Methods. We also compare our approach with three baseline methods: K-nearest neighbor model (KNN) [18], GSTARIMA [12] and the tweet semantic based method (TwiSemantic) [37]. KNN and GSTARIMA are based on recent speed observations with considering the road network topology. TwiSemantic combines recent traffic speed with tweets semantics using linear regression.

In KNN, we use non-weighted algorithm and the neighbor number is 5 with the best result here. In GSTARIMA, we set the spatial weighted matrix following the paper. In TwiSemantic, tweet semantics are mapped into the same vocabulary as our model used, which contains $1,857$ words and is obtained by removing stop words and words with frequencies lower than 10 from traffic related tweets. Our

model is initialized by pre-analyzing a small fraction of data, with $\delta = -1, \beta$ equal to the opposite value of the speed median in the fraction, $\psi_{i,j} = \frac{1}{K}, \phi_{j,k} = \frac{1}{M}$, and the topic number K is 2, denoting congested and normal. The dataset is divided into training and testing data by time stamps. In the training stage, the speed variables $v_{t,s}$ are observed to learn the model parameters; in the testing stage, the speeds are latent, the posterior distributions of which are inferred with fitted model parameters.

6.1.3 Accuracy Metrics

Two criteria are frequently used to measure the prediction error for numerical values, namely Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE).

In testing set, the number of time stamps is denoted as T_{test}, and the number of road links is denoted as M_{test}. RMSE and MAPE, which serve to aggregate the magnitudes of errors between predicted speeds value $\hat{v}_{t,s}$ and speed variable $v_{t,s}$ into a single measure of predictive power, are defined as

$$RMSE = \sqrt{\frac{1}{T_{test} \times M_{test}} \sum_{t=1}^{T_{test}} \sum_{s=1}^{M_{test}} (v_{t,s} - \hat{v}_{t,s})^2} \tag{24}$$

$$MAPE = \sqrt{\frac{1}{T_{test} \times M_{test}} \sum_{t=1}^{T_{test}} \sum_{s=1}^{M_{test}} \left(\frac{|v_{t,s} - \hat{v}_{t,s}|}{v_{t,s}} \right)}, \tag{25}$$

where RMSE is usually sensitive in predicting high speeds, that is, bad predictions of high values may easily cause huge RMSE, while MAPE is more sensitive in predicting low speeds, which means bad prediction when congestions happen may result in huge MAPE.

6.2 Model Validation

This part gives an overall performance analysis to TEGPAM, including the comparison between prediction accuracy and model efficiency, the predicted distribution analysis about fitting degree and uncertainty, and the predictive performance in terms of different degrees of speed sensor sparsity.

6.2.1 Trade-off between Accuracy and Efficiency

To show the overall performance of the proposed model, TEGPAM learned by the variational inference is compared with that learned by Gibbs sampling, which is named as M-Gibbs. Gibbs sampling is a wide-used Markov chain Monte Carlo (MCMC) algorithm for approximation from a probability distribution [52]. The comparison is conducted on prediction accuracy and model efficiency, the trade-off of which is made by the training set size of TEGPAM and the iteration time of M-Gibbs.

The result is presented in Fig. 7, with x-axis being the [training set ratio/the iteration time], y-axis-left representing the prediction error MAPE and y-axis-right representing the time cost. TEGPAM with more training data and M-Gibbs with more iterations achieve a better prediction accuracy, which becomes stable with the training set ratio over 0.5 and the iteration time over 14,000. And the error of M-Gibbs decreases faster than TEGPAM, which indicates its good convergence rate. The training time of both increases linearly, but TEGPAM is faster than M-Gibbs with a slower growth rate, which shows the efficiency of variational inference as it obtains an analytical solution for parameter updates. What's more, TEGPAM is scalable for future works on adding more information and features to improve the accuracy. The efficiency of TEGPAM can be further optimized by parallelization, while M-Gibbs as a type of sequential algorithm is difficult to be parallelized.

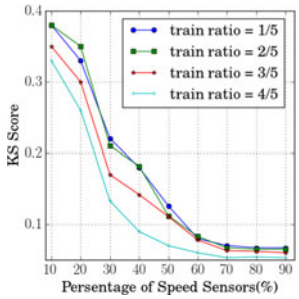


Fig. 4. Average KS score on datasets in two cities.

6.2.2 Predicted Distribution Analysis

To verify how our proposed method can capture the *distribution* of traffic speeds, we first quantify the fitting degree of the predicted distribution, then we analyze the uncertainty of the prediction.

Distribution Fitting Degree. To measure the fitting degree between the predictive distribution and the true empirical speed distribution, we adopt the Kolmogorov-Smirnov (KS) test [53], which is widely used in statistic community to evaluate the fitting degree of two distributions by quantifying the distance between them.

For each road link s, let $F_T(v^s)$ denote the empirical cumulative distribution function for T i.i.d. speed random variables v^s, and V_t^s is the traffic speed of link s at time t. The function is

$$F_T(v^s) = \frac{1}{T} \sum_{t=1}^{T} I_{V_t^s \leq v^s}, \tag{26}$$

where $I_{V_t^s \leq v^s}$ is an indicator function which equals to 1 if $V_t^s \leq v^s$, and 0 otherwise.

Denote the predictive cumulative distribution function as $F(v^s)$. The KS score for these two cumulative distribution function $F_T(v^s)$ and $F(v^s)$ is

$$Z_T(s) = \sup_{v^s} |F_T(v^s) - F(v^s)|, \tag{27}$$

the supremum over all distances between two functions. For all the unobserved speeds $\mathcal{S}_m$, the average KS score is

$$Z_T = \frac{\sum_{s=1}^{|\mathcal{S}_m|} Z_T(s)}{|\mathcal{S}_m|}. \tag{28}$$

TEGPAM is trained and KS score is tested, with the fraction of training data as $1/5, \ldots, 4/5$, and the percentages of speed sensors as $p_v = 10, \ldots, 90$ percent. A lower KS score indicates a better fitting, the results is shown in Fig. 4. We observe that with more training data, a better model fit is achieved, for example, $4/5$ training data with only

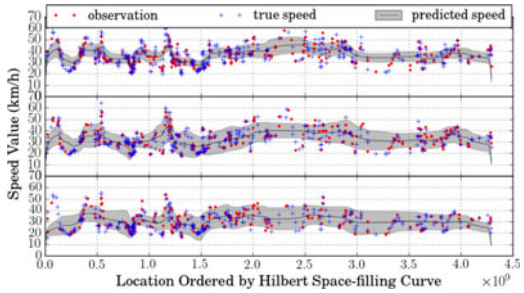


Fig. 5. The prediction profiling on the true and predicted speed in terms of different locations.

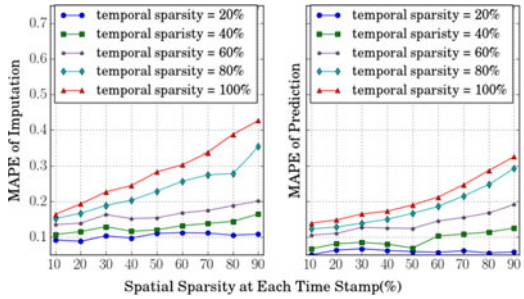


Fig. 6. Imputation and prediction error w.r.t degrees of spatial and temporal sparsity.

30 percent observations obtains a similar fit with 60 percent observations under $1/5$ training data. Typically, a KS score less than 0.1 shows a strong distribution fit, which is achieved when more than 70 percent links are sensed even when the fraction of training data is 10 percent.

Prediction Uncertainty. GPs provide the uncertainty of prediction. To exploit the advantage when using GP to capture the spatial correlation of traffic speed, we profile the prediction result by drawing true speed and the predicted posterior mean and variance, for all links.

Since the links spread 2-dimensionally, a space-filling Hilbert curve [54] is used to map between 2D and 1D space. If (x, y) are the coordinates of a location, and $\hat{d}$ is the distance along the curve when it reaches that location, then locations that have nearby $\hat{d}$ values will also have nearby (x, y) values. The links are mapped to 1D by a Hilbert curve, and are ordered by the distance along the curve. Then the prediction profile, including true speed, predicted posterior mean and covariance, can be drawn in the link order.

Prediction profiles at 00:00, 08:00 and 17:00 on a Monday are shown in Fig. 5. We observe that the variances (shadow area) of nearby locations are generally similar, and the wave patterns at different time are similar, which indicates the stability of spatial correlation. Meanwhile, the variance of rush hour (08:00 and 17:00) is bigger than that of plain hour (00:00), which shows the uncertainty of rush hour since the traffic condition is more dynamic.

6.2.3 Data Sparsity Analysis

To analyze the impact of data sparsity, consider two types of sparsity: 1) *spatial sparsity* represents the missing percentage of speed-unobserved links at a time stamp; 2) *temporal sparsity* represents the missing percentage of speed-unobserved time stamps for a link. To achieve a good predictive performance for extremely sparse datasets, we impute missing values through tensor factorization[44], which can exploit spatial and temporal correlations. The incomplete data is represented by a tensor $G = \mathbb{R}^{d \times p \times w}$, where d, p and w denoted the number of time stamps a day, links and same days of week. We use canonical (CP) factorization to obtain low rank representation for the tensor to impute data.

Fig. 6 draws error curves of links grouping by different degrees of temporal sparsity as $\{20\%, 40\%, \ldots, 100\%\}$, and each point of a curve denotes that these links are tested on time stamps with different degrees of spatial sparsity as $\{10\%, 20\%, \ldots, 90\%\}$. For example, the red curve refers to average MAPE values when testing links with completely no historical observations, on time stamps with different degrees of spatial sparsity. We observe that the predicting on links with lower sparsity degree has smaller error, and our model fusing more data performs better than purely imputing the data by tensor decomposition, however imputation will improve the prediction since it provides more data to perform Eq. (12).

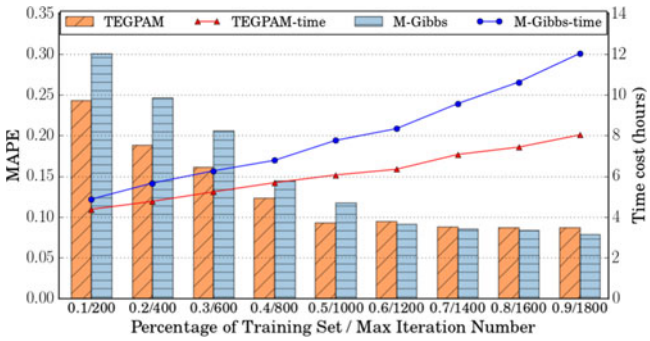


Fig. 7. Prediction accuracy and time cost of TEGPAM inferred by variational inference and Gibbs sampling.

6.3 Effectiveness of Each Data Source

To validate the effectiveness of each data source, we first give an overall comparison on prediction accuracy of six sub models tackling different data combinations to show the improvement of data fusion, then we separately validate the effectiveness of each data source by comparing the prediction accuracy with/without using this data and with different percentage of this data to give a finer analysis on the impact of this data.

6.3.1 Overall Comparison

To show the improvement of fusing more data, we compare the sub-models M-1, M-2, M-3 fusing two data sources and M-23, M-13 based on one data source in the baseline methods, note that M-12 is not added to this set of experiments because of the insufficiency of only using twitter data. The percentage of speed sensor and trajectory are all set as 50 percent, and the fraction of testing data ranges from $1/6$ to $5/6$.

The result is shown in Fig. 8. We observe that TEGPAM using 3 data sources performs steadily the best, while M-1, M-2 and M-3 fusing 2 data sources take second place and M-23 and M-13 using only one data source performs the worst, which validates the intuition that speed prediction combining more information can improve the accuracy. Meanwhile, comparing the error of 2 data based models, M-1 is the worst, so excluding speed sensing data impacts

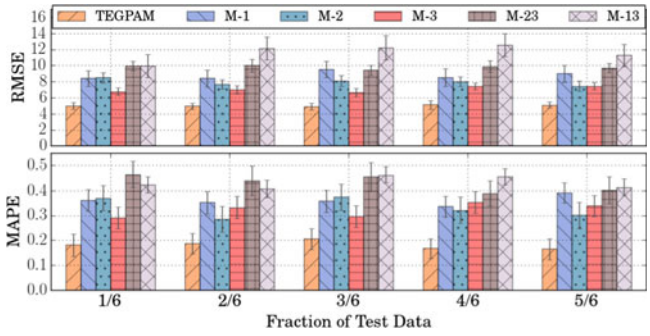


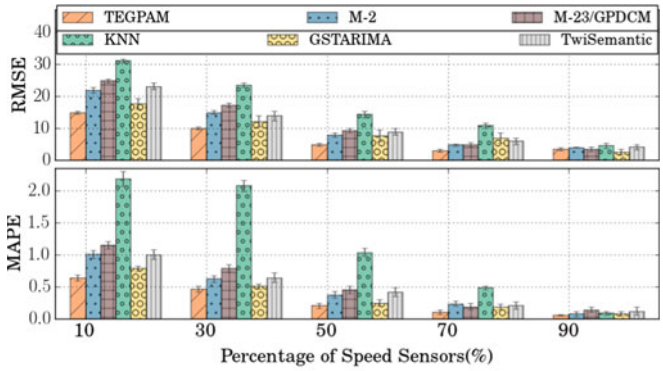
Fig. 8. Comparison on different data combinations.

the prediction most, which implies that speed sensing data might be more affective than trajectory data while trajectory is better than twitter data. Observing M-23 better than M-13 also proves the indication.

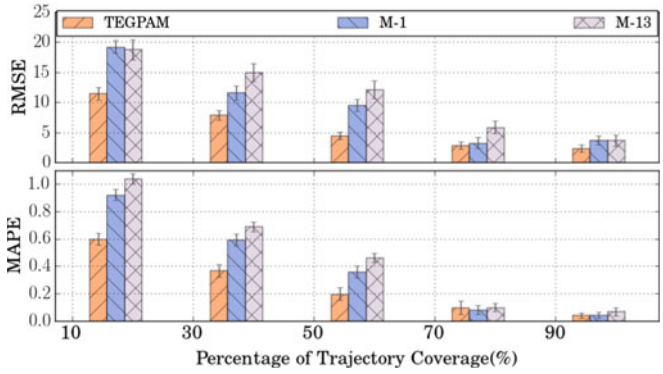
6.3.2 Effectiveness of Traffic Speed Data

We first compare the performance of models with or without using speed sensors, to demonstrate the effectiveness of speed data. Fix the percentage of speed sensor p_v and the percentage of trajectory coverage p_p as 50 percent, then test integrated TEGPAM, speed based M-23(GPDCM) and speed excluded M-1 with fraction of testing data as $\{1/6, \ldots, 5/6\}$. The results are shown in Fig. 9a. The performance of TEGPAM is steadily better than the others, while only using speed sensors (M-23) is insufficient and limited, which again demonstrates the benefit of multi-source data.

To answer the questions: with other data sources, how sparse the historical traffic speed data can be to predict current traffic speed? We set the percentage of speed sensors as $p_v = 10, 30, \ldots, 90\%$, under the fraction of testing data and the percentage of path coverage p_p as 50 percent. The traffic speed based models, TEGPAM, M-2 and M-23 (GPDCM) of our approach, and KNN, GSTARIMA, are applied on the training set. The results are shown in Fig. 10a. The score decrease trend of each model shows that with more current or recent observations, the missing speeds will be better predicted. However, when fewer than 70 percent speed sensors, TEGPAM fusing multi-source data performs better than the traffic speed based model (M-23/GPDCM, KNN and GSTARIMA), especially, the RMSE of TEGPAM is nearly 40, 50 and 15 percent less than that of M-23/GPDCM, KNN and GSTARIMA when only 10 percent links are observed. The results show the impact of traffic speed data



(a) Speed and twitter based models.



(b) Trajectory based methods.

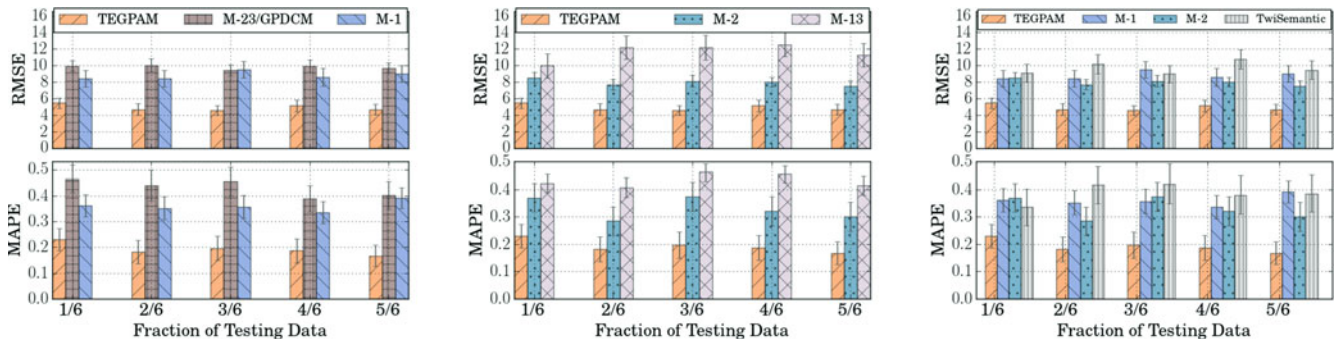
Fig. 10. Comparisons with changing the percentage of data.

and prove the effectiveness of TEGPAM when speed sensors are largely unavailable.

6.3.3 Effectiveness of Trajectory Data

Trajectory data, with time and link information, has a direct relationship with road speeds. In a trajectory with available travel time, more information about the speed of those road links in this path is contained. If the travel time is big, we will be more confident to infer that some road links in the trajectory are congested and the speeds of them must be low. This section validate the effectiveness of trajectory data in predicting unobserved traffic speeds.

We first compare the prediction accuracy of models with/without using trajectory information. Three models are applied on the fraction of testing data as $\{1/6, \ldots, 5/6\}$: integrate TEGPAM, trajectory based M-13 and trajectory excluded M-2. The percentages of speed p_v and trajectory



(a) Speed data validation. (b) Trajectory data validation. (c) Twitter data validation.

Fig. 9. Comparisons between models with/without using one data source.

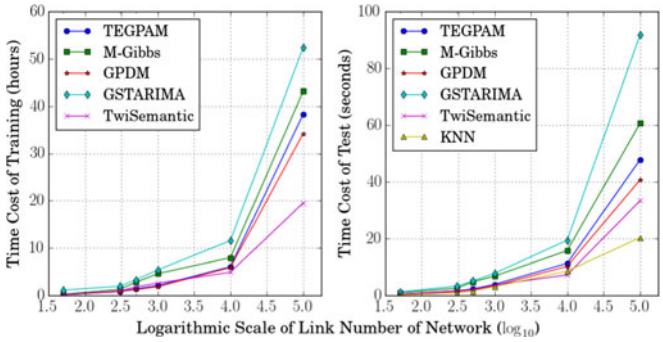


Fig. 11. Computation time for different network sizes.

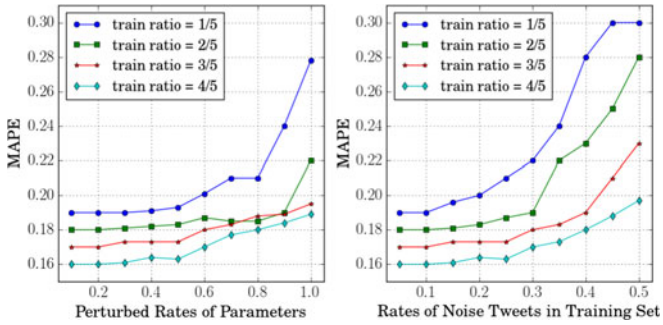


Fig. 12. (a) Parameter sensitivity and (b) noise reliability.

coverage p_p remain 50 percent. The RMSE and MAPE scores are shown in Fig. 9b. The performance of M-2 is better than M-13, which implies that trajectory data alone is also not good enough to predict traffic speeds. Meanwhile, comparing M-2 here with M-1, and M-13 with M-23 in Fig. 9a, we observe that speed based model M-23 has a slight advantage to trajectory based model M-13, which validates the effectiveness of speed data in some degree.

To explore how much trajectory data can help predict speed, we set the percentage of trajectory sensor as $p_p = 10, 30, \ldots, 90$ percent, under the fraction of testing data as 50 percent and the percentage of speed sensor p_v as 50 percent. The trajectory based models, TEGPAM, M-1 and M-13 are applied on the training set. From Fig. 10b, an obvious decrease of error with the increase of path coverage shows the effectiveness of trajectory data. Meanwhile, compared with Fig. 10a when p_p keeps 50 percent and $p_v < 50$ percent, here when p_v keeps 50 percent and $p_p < 50$ percent, the error score of TEGPAM is slightly lower, which indicates that traffic speed data, even sparse, can still act more efficiently than trajectory data for speed prediction.

6.3.4 Effectiveness of Twitter Data

The traffic related information of twitter data is very dynamic, this section is designed to answer the question: what role does twitter data play in predicting current traffic speeds, a strong predictor or a good supplement to other data sources?

To answer the question, we apply twitter based models, TEGPAM, M-1, M-2 of our approach and TwiSemantic on the settings of 50 percent speed sensor percentage and trajectory coverage. From Fig. 9c, we observe that the integrate model TEGPAM performs steadily good, and with the same data source of speed and twitter, M-2 gains less error than TwiSemantic, which demonstrate the effectiveness of our model, especially the Traffic Topic model. Meanwhile, comparing model with and without using twitter, e.g., M-1 and M-13 in Fig. 10b, M-2 and M-23 in Fig. 10a, we observe that when the percentage is less than 50 percent, the models including twitter data (M-1 and M-2) perform better than those excluding twitter (M-13 and M-23). The results indicate that when observed speed percentage is low, Twitter data is a strong complement to speed sensing data.

6.4 Model Efficiency and Stability

Model efficiency is shown in Section 6.4.1. Then Sections 6.4.2 and 6.4.3 validates two factors of model stability: 1) sensitivity to parameters and 2) reliability on noisy tweets.

6.4.1 Model Efficiency

This set of experiments compare the time costs for training and testing, incurred by TEGPAM and the baseline methods. We conducted these experiments on two city datasets and the average values maintain as the results. For each time stamp, we randomly choose 50 percent links to be unobserved and need to be predicted.

We introduced inducing variables to approximate the full-rank GP as mentioned in Section 5.3. Fig. 11 compares the training and testing time cost at a 5-minute rate for network with different link numbers, which is logarithmized by 10 in x-axis. The computation complexity is reduced from $O(S^3)$ to $O(U^2S)$, and the exponential increase in terms of $\log_{10} S$ is appropriately showed in the figure. It is worth pointing out that to handle million-size datasets, GP could be further distributed (e.g., [55], [56], [57], [58], [59]). Overall, KNN is fast but performs bad when speed sensors are sparse, and GSTARIMA has a good performance but is time consuming. Our model costs a reasonable training time while achieves much higher accuracy, and the efficiency can be further improved by parallelization in our future work.

6.4.2 Sensitivity Analysis of Parameters

Parameters mainly come from the traffic topic model, where β, δ determine the Logistic function in the Bernoulli distribution of traffic status, $\boldsymbol{\psi}$ determines the multinomial distribution of topics, and $\boldsymbol{\phi}$ determines the multinomial distribution of words. These four parameters are perturbed based on white noises, with means equal to 0, and variances equal to $\{k_\beta, k_\delta, \boldsymbol{k_\psi}, \boldsymbol{k_\phi}\}$ respectively. We set the perturbed rate σ as 0.1 to 1.0, then the variances are $\{k_\beta = \sigma, k_\delta = \sigma, \boldsymbol{k_\psi} = [\frac{\sigma}{K}]^K, \boldsymbol{k_\phi} = [\frac{\sigma}{M}]^M\}$. The results shown in Fig. 12a indicate that our model performs stable with perturbed rate less than 0.6.

6.4.3 Reliability Analysis of Noisy Tweets

Noise in social media is a common phenomenon. This section studies the reliability of TEGPAM under noisy tweets that present contents inconsistent with real traffic speeds (e.g., saying *"Rd.5 is really congested!"* while the speed of Rd.5 is actually normal). We generate simulated noisy tweets, which contain words *"congested, slow, stuck, crash"*, and inject those noisy tweets into real dataset randomly with the empirical distributions of time stamps and road links in the training Twitter data. The rates of noisy tweets are $noise = 0.05, 0.1, \ldots, 0.5$. The performance of TEGPAM

based on the contaminated training set shown in Fig. 12b indicates the robustness of our model, especially the traffic topic model, when the rate of noisy tweets is less than 0.25.

7 Conclusion

This paper proposes a novel probabilistic framework to predict road traffic speed with multiple cross-domain data. Existing works are mainly based on speed sensing data, which suffers data sparsity and low coverage. In our work, we handle the challenges arising from fusing multi-source data, including location uncertainty, language ambiguity and data heterogeneity, using Location Disaggregation Model, Traffic Topic model and Traffic Speed Gaussian Process Model. Experiments on real data demonstrate the effectiveness and efficiency of our model. For Future work, we plan to implement kernel-based and distributive GP, so the traffic prediction framework can be applied into a real-time large traffic network.

Acknowledgments

This work is supported by the China 973 Fundamental R&D Program (No. 2014CB340300), NSFC program (No. 61472022, 61421003), SKLSDE-2016ZX-11, and the Beijing Advanced Innovation Center for Big Data and Brain Computing.

References

[1] X. Yu and P. D. Prevedouros, "Performance and challenges in utilizing non-intrusive sensors for traffic data collection," *Advances Remote Sens.*, vol. 2, pp. 45–50, 2013.
[2] S. Clark, "Traffic prediction using multivariate nonparametric regression," *J. Transp. Eng.*, vol. 129, pp. 161–168, 2003.
[3] B. Williams, P. Durvasula, and D. Brown, "Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models," *Transp. Res. Rec.*, vol. 1644, pp. 132–141, 1998.
[4] M. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an Urban network: Comparison of multivariate and univariate approaches," *Transp. Res. Rec.*, vol. 1857, pp. 74–84, 2004.
[5] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transp. Res.*, vol. 19, pp. 606–616, 2011.
[6] S. M. Turner, W. L. Eisele, R. J. Benz, and D. J. Holdener, *Travel Time Data Collection Handbook*. Office Highway Inf. Manage., Federal Highway Administration, US Dept. Transportation, Washington, DC, USA, 1998.
[7] B. Abdulhai, H. Porwal, and W. Recker, "Short-term traffic flow prediction using neuro-genetic algorithms," *J.-Intell. Transp. Syst.*, vol. 7, no. 1, pp. 3–41, 2002.
[8] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res.*, vol. 10, pp. 303–321, 2002.
[9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal stochastic time series process," No. LTVA/29242/CE99/103, 1999.
[10] Y. Kamarianakis and P. Prastacos, "Space-time modeling of traffic flow," *Computers & Geosciences*, vol. 31, no. 2, pp. 119–133, 2005.
[11] R. Giacomini and C. W. Granger, "Aggregation of space-time processes," *J. Econometrics*, vol. 118, pp. 7–26, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0304407603001325
[12] M. Xinyu and H. Jianming, "Urban traffic network modeling and short-term traffic flow forecasting based on GSTARIMA model," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 19–22.
[13] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 246–254, Jun. 2009.
[14] J. Guo and B. M. Williams, "Real-time short-term traffic speed level forecasting and uncertainty quantification using layered kalman filters," *Transp. Res. Rec.*, vol. 2175, pp. 28–37, 2010.
[15] M. Lippi, M. Bertini, and P. Frasconi, "Collective traffic forecasting," *Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, pp. 259–273, 2010.
[16] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 2, pp. 246–254, Jun. 2009.
[17] S. Huang and A. W. Sadek, "A novel forecasting approach inspired by human memory: The example of short-term traffic volume forecasting," *Transp. Res.*, vol. 17, pp. 510–525, 2009.
[18] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," *Procedia-Social Behavioral Sci.*, vol. 96, pp. 653–662, 2013.
[19] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach," *Transp. Res.*, vol. 13, pp. 211–234, 2005.
[20] A. Khosravi, E. Mazloumi, S. Nahavandi, D. Creighton, and J. W. C. V. Lint, "A genetic algorithm-based method for improving quality of travel time prediction intervals," *Transp. Res.*, vol. 19, pp. 1364–1376, 2011.
[21] S. Jinxing and L. Wenquan, "Sensitivity analysis of wavelet neural network model for short-term traffic volume prediction," *Appl. Math.*, vol. 2013, 2013, Art. no. 953548.
[22] J. Shang, Y. Zheng, W. Tong, E. Chang, and Y. Yu, "Inferring gas consumption and pollution emission of vehicles throughout a city," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 1027–1036. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=217455
[23] S. Liu, Y. Yue, and R. Krishnan, "Adaptive collective routing using gaussian process dynamic congestion models," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 704–712.
[24] S. Liu, Y. Yue, and R. Krishnan, "Non-myopic adaptive route planning in uncertain congestion environments," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2438–2451, Sep. 2015.
[25] S. Thajchayapong, J. A. Barria, and E. Garcia-Trevino , "Lane-level traffic estimations using microscopic traffic variables," in *Proc. Int. IEEE Conf. Intell. Transp. Syst.*, 2010, pp. 1189–1194.
[26] E. Mai and R. Hranac, "Twitter interactions as a data source for transportation incidents," in *Proc. 92nd Ann. Meeting Transp. Res. Board*, 2013, pp. 1613–1636.
[27] A. Schulz and P. Ristoski, "The car that hit the burning house: Understanding small scale incident related information in microblogs," in *Proc. 7th Int. Conf. Weblogs Social Media*, vol. 13, 2013, pp. 11–14.
[28] B. Pan, Y. Zheng, D. Wilkie, and C. Shahabi, "Crowd sensing of traffic anomalies based on human mobility and social media," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2013, pp. 344–353.
[29] P. Raphiphan, A. Zaslavsky, P. Prathombutr, and P. Meesad, "Context aware traffic congestion estimation to compensate intermittently available mobile sensors," in *Proc. 10th Int. Conf. Mobile Data Manage.: Syst. Serv. Middleware*, 2009, pp. 405–410.
[30] T. H. Maze, M. Agarwai, and G. Burchett, "Whether weather matters to traffic demand, traffic safety, and traffic operations and flow," *Transp. Res.*, vol. 1948, pp. 170–176, 2006.
[31] R. Alfelor, H. S. Mahmassani, and J. Dong, "Incorporating weather impacts in traffic estimation and prediction systems," US Dept. Transportation, Washington, DC, USA, Tech. Rep. FHWA-JPO-09-065, EDL# 14497, 2009.
[32] S. Liu, S. Wang, K. Jayarajah, A. Misra, and R. Krishnan, "TODMIS: Mining communities from trajectories," in *Proc. ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 2109–2118.
[33] Y. Li, Y. Zheng, H. Zhang, and L. Chen, "Traffic prediction in a bike-sharing system," in *Proc. 23rd SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2015, Art. no. 33.
[34] M. Shao, J. Li, F. Chen, H. Huang, S. Zhang, and X. Chen, "An efficient approach to event detection and forecasting in dynamic multivariate social media networks," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 1631–1639.
[35] W. Yu, J. Li, M. Z. A. Bhuiyan, R. Zhang, and J. Huai, "Ring: Real-time emerging anomaly monitoring system over text streams," *IEEE Trans. Big Data*, to be published. doi: 10.1109/TBDATA.2017.2672672.
[36] S. Liu and Q. Qu, "Dynamic collective routing using crowdsourcing data," *Transp. Res. Part B: Methodological*, vol. 93, pp. 450–469, 2016.
[37] J. He, W. Shen, P. Divakaruni, L. Wynter, and R. Lawrence, "Improving traffic prediction with tweet semantics," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1387–1393.

[38] J. Luttinen and A. Ilin, "Efficient Gaussian process inference for short-scale spatio-temporal modeling," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2012, pp. 741–750.
[39] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. Cambridge, MA, USA: MIT Press, 2005.
[40] A. Schulz, P. Ristoski, and H. Paulheim, "I see a car crash: Real-time detection of small scale incidents in microblogs," in *Proc. Extended Semantic Web Conf.*, 2013, pp. 22–33.
[41] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[42] P.-T. Chen, F. Chen, and Z. Qian, "Road traffic congestion monitoring in social media with hinge-loss Markov random fields," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 80–89.
[43] 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 Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 901–921, Jul. 2015.
[44] M. T. Asif, N. Mitrovic, L. Garg, J. Dauwels, and P. Jaillet, "Low-dimensional models for missing data imputation in road networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2013, pp. 3527–3531.
[45] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "Introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, 1999.
[46] D. M. Blei and J. D. Lafferty, "Correlated topic models," in *Proc. Advances Neural Inf. Process. Syst.*, 2005, pp. 147–154.
[47] C. Wang and D. M. Blei, "Variational inference in nonconjugate models," *J. Mach. Learn. Res.*, vol. 14, no. 4, pp. 1005–1031, 2012.
[48] M. Seeger, C. K. I. Williams, and N. D. Lawrence, "Fast forward selection to speed up sparse Gaussian process regression," in *Proc. Int. Workshop Artif. Intell. Statist.*, 2003, pp. 205–212.
[49] E. Snelson and Z. Ghahramani, "Sparse Gaussian process using pseudo-inputs," in *Proc. Advances Neural Inf. Process. Syst.*, 2010, pp. 1257–1264.
[50] [Online]. Available: http://inrix.com/publicsector.asp
[51] D. B. Johnson, "Efficient algorithms for shortest paths in sparse networks," *J. ACM*, vol. 24, no. 1, pp. 1–13, 1977.
[52] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. New York, NY, USA: Springer-Verlag, 2006.
[53] A. J. Koning and P. Liang, "Goodness-of-fit tests for a heavy tailed distribution," *J. Statist. Planning Inference*, vol. 138, no. EI 2005–44, pp. 3960–3981, 2005.
[54] D. Voorhies, "Space-filling curves and a measure of coherence," *Graph. Gems II*, vol. 13, no. 12, 1991, Art. no. 485C486.
[55] Q. M. Hoang, T. N. Hoang, and K. H. Low, "A generalized stochastic variational Bayesian hyperparameter learning framework for sparse spectrum gaussian process regression," in *Proc. 30th AAAI Conf. on Artif. Intell.*, 2016, pp. 2007–2014.
[56] Y. Gal, M. V. D. Wilk, and C. E. Rasmussen, "Distributed variational inference in sparse Gaussian process regression and latent variable models," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 3257–3265.
[57] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A distributed variational inference framework for unifying parallel sparse Gaussian process regression models," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 382–391.
[58] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," *arXiv preprint arXiv:1309.6835*, 2013.
[59] T. N. Hoang, Q. M. Hoang, and B. K. H. Low, "A unifying framework of anytime sparse Gaussian process regression models with stochastic variational inference for big data," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 569–578.

Lu Lin received the BS degree from the School of Computer Science and Engineering, Beihang University, Beijing, China, in 2014. She is currently working toward the MSc degree at Beihang University. Her research interests include urban computing and graphical model.

Jianxin Li received the PhD degree from Beihang University, in 2008. He is a professor in the School of Computer Science and Engineering, Beihang University. He was a visiting scholar in the Machine Learning Department at CMU, in 2015, and a visiting researcher at MSRA in 2011. His current research interests include data analysis and processing, distributed systems, and system virtualization.

Feng Chen received the PhD degree in computer science from the Virginia Polytechnic Institute and State University, Virginia, in 2012. He is an assistant professor with the University at Albany, State University of New York, Albany. He has published 30 referred articles in major data mining venues. His research focuses on the detection of emerging events and other relevant patterns in the mobile context and/or data mining of spatial temporal, textual, or social media data.

Jieping Ye received the PhD degree in computer science from the University of Minnesota, Twin Cities, in 2005. He is an associate professor in the Department of Computational Medicine and Bioinformatics and Department of Electrical Engineering and Computer Science, University of Michigan. His research interests include machine learning, data mining, and biomedical informatics. He was a senior program committee/area chair of many conferences including NIPS, KDD, IJCAI, ICDM, SDM, and ACML. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is a senior member of the IEEE.

Jinpeng Huai is a professor with Beihang University. He is a member of the Consulting Committee of the Central Government Information Office, and Chairman of the Expert Committee in both the National e-Government Engineering Taskforce and the National e-Government Standard office. His research interests include middleware, grid computing, trustworthiness, and security. He served as the program chair of many notable conferences, including IEEE, SRDS2007, WWW2008, and ICWS2008.