Traffic estimation and prediction via Online Variational Bayesian Subspace Filtering

Charul*, Uttkarsha Bhatt†, Pravesh Biyani* and Ketan Rajawat†
* IIIT-Delhi, India, † IIT-Kanpur, India

***Abstract*—With the increased proliferation of smart devices, the transit passengers of today expect higher quality of service in the form of real-time traffic updates, accurate expected time-of-arrival (ETA) predictions. Providing these services requires the public transit agencies and private transportation players to maintain full situational awareness of the city-wide traffic. However, most such agencies and companies are resource-constrained and do not have access to city-wide traffic data. The availability of sparsely sampled and outlier-corrupted traffic data renders the resulting traffic maps patchy and unreliable, and necessitates the use of sophisticated real-time traffic interpolation and prediction algorithms. Moreover, since the traffic data is measured and collected in a sequential manner, the estimations must also be generated online. Thankfully, the traffic matrices are spatially and temporally structured, allowing the use of time-series and matrix/tensor completion algorithms. This work puts forth a generative model for the traffic density and subsequently use variational Bayesian formalism to learn the parameters of the model. Specifically, we consider low-rank traffic matrices whose subspace evolves according to a state-space model with possible sparse outliers. Different from most matrix/tensor completion algorithms, the proposed model is equipped with automatic relevance determination priors that allows it to learn the parameters in a completely data-driven manner. A forward-backward algorithm is proposed that enables the updates to be carried out at low-complexity. Simulations carried out on real traffic data demonstrate that that the proposed algorithm better predicts the future traffic densities and the ETA as compared to the state-of-the-art matrix/tensor completion algorithms.**

I. Introduction

Real-world data collected from various sensors are often incomplete as well as noisy with the possibility of outliers. As a consequence, such data naturally demands filling the missing entries and removal/segregation of outliers to perform tasks in a meaningful manner. Examples of applications that benefit from imputing of missing data includes road traffic estimation and prediction, air quality measurement across the city network, collaborative filtering and even video foreground-background separation etc. The collected data in these applications are naturally represented in the form of a matrix with missing entries that need to be inferred. Many approaches to impute the missing entries model the data as belonging to an underlying low-dimensional subspace that can subsequently be recovered via matrix completion [1], [2], [3], [4], robust principal component analysis (PCA) [5], [6], or their tensor counterparts [7]. Such techniques approach the problem from a static perspective. Specifically, matrix or tensor completion is applied to data collected as a whole to impute the missing entries. In contrast, many real-world applications including the ones mentioned above follow a common theme of a) inherently dynamic, consisting of sequentially arriving data that must be dealt in an online manner, b) while the data lies in the underlying low-rank subspace that also slowly evolves over time. Offline algorithms based on matrix completion often involve tuning parameters with very limited possibility of altering them in an online fashion. On the other hand Bayesian approaches offer a higher degree of flexibility of modeling the data in the above stated scenarios. For instance, unlike approaches for subspace tracking, it is possible to capture the temporal evolution in the data through a state space model. PB Comment: We have to add our review for other subspace tracking and related work here.

This work considers the first low-rank robust subspace filtering approach for online matrix imputation and prediction. Different from the existing matrix and tensor completion formulations, we consider low-rank matrices whose underlying subspace evolves according to a state-space model. As incomplete columns of the data matrix arrive sequentially over time, the low rank components as well as the state-space model are learned in an online fashion using the variational Bayes formalism. In particular, component distributions are chosen to allow automatic relevance determination (ARD) and unlike the matrix or tensor completion works, the algorithm parameters such as rank, noise powers, and state noise powers need not be specified or tuned. A low-complexity forward-backward algorithm is also proposed that allows the updates to be carried out efficiently. Enhancements to the proposed algorithm, capable of learning time-varying state-transition matrices, operating with a fixed lag, and robust to outliers, are also detailed. Our approach is general and we demonstrate its efficacy on various settings. In particular, we discuss the traffic estimation problem in detail and show that the variational Bayesian approach can be used to learn the model in online setting which estimates the missing link speed entries when only a fraction of speed entries are available. We also use the learned model to predict the link speed in the future time intervals. Additionally, the applicability of the proposed algorithm on the air quality sensor and video datasets is also shown. In summary, the contributions of the present work are as follows:

1) We present the variational Bayesian subspace filtering (VBSF) approach for online subspace filtering. The underlying subspace is modeled as comprising of a low rank subspace evolving according to a linear dynamical model. The proposed method is capable of automatically

assessing the relevance of each parameter.

2) Robust version of the VBSF algorithm is proposed for outlier removal and data cleansing. Different from the existing works that utilize variational Bayesian PCA or robust PCA for solving static problems, the proposed approach is specifically designed for dynamic settings.
3) The proposed algorithms are tested on real traffic data collected over a large area and the resulting imputation and ETA estimation accuracy is studied. Comparisons made with state-of-the-art algorithms reveal the efficacy of the proposed algorithms.

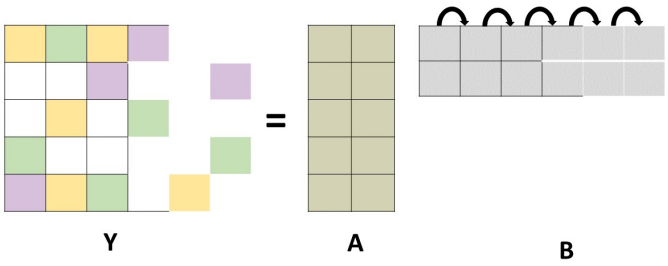

Fig. 1: Online Variational Bayesian Filtering

A. Related work

Variational Bayesian approaches for matrix completion and robust principal component analysis are well known [3], [8], [9], [10], [11], [12], [13], [4], [14], [15]. One of the first works considered the measured matrix to be expressible as a product of low-rank matrices, associated with appropriate ARD priors [3] while faster algorithms for similar settings were proposed in [8], [9], [15]. More recently, other approaches towards modeling the measured matrices have also been proposed [10], [11].Moreover, variational Bayesian approaches have also been applied to road traffic estimation; see e.g. [12]. However, these approaches do not explicitly model the evolution of the underlying subspace. Likewise, none of the existing variational Bayesian approaches for low rank matrix completion model the evolution of the subspace [3], [4], [14], [15]. In contrast to these, the state-space modeling in our work is inspired from [13], where the low-complexity updates were first proposed in the context of linear dynamical models. The VBSF algorithm in the current work extends and generalizes that in [13] to incorporate low-rank structure and outliers.

Several non bayesian algorithms have been proposed to addressthe online subspace estimation problem from incomplete observations[2], [16], [17], [18]. GROUSE is one of the algorithm that uses a update on the Grassmannian manifold to estimate the subspace [2]. GRASTA or robust GROUSE enhances the robustness by incorporating the l1 norm cost function[16]. OP-RPCA[17], another algorithm which uses alternating minimization to compute the outliers and the subspace. This algorithm also uses a stochastic gradient updates of an l2 regularized least squares cost to track the subspace. ROSETA[18] is another variant which uses the similar l1 cost function as that of GRASTA and adding another constraint to track the time-varying nature of the low dimensional subspace. However, our algorithm have a advantage that it can model the columns of the data in a generative manner using a auto regressive model. Also, bayesian learning does not require any parameter tuning as rank,noise powers, and state noise powers.

B. Applications:

Related work + details of application

Traffic estimation and prediction are the central components of any urban traffic congestion management system [19]. Comprehensive traffic density maps not only aid in the discovery of traffic flow patterns, but are also invaluable for city planners. With the advent of smartphones, public transportation services as well as private on-demand transportation companies are increasingly relying on the availability of real-time traffic maps for resource allocation and logistics [20]. These on-demand commuter shuttle services such as Shuttl (India) and Chariot (US and India), as well as cab companies such as Lyft, Uber, and Ola, often employ in-house traffic estimation and prediction algorithms to make routing decisions. Unlike cellphone application developers like Google and Microsoft however, these transportation providers lack the resources to collect accurate city-wide traffic densities and thus end up working with incomplete and noisy data sets.

Such providers rely on probe vehicles — GPS enabled and possibly crowd-sourced agents that upload speed measurements and corresponding location tags at sporadic times. Since traffic densities are inferred from speed measurements, they are often ridden with outliers, e.g., corresponding to random velocity changes unrelated to traffic. Once such outliers have been identified and removed, the traffic estimation problem entails estimating traffic densities at locations and times where no measurements are available. Finally, prediction of traffic in the near future is necessary to calculate expected-time-of-arrival (ETA), fastest route, and other related quality of service metrics for road users. More generally, all three problems should be solved jointly, since generating accurate traffic estimates requires knowing the outliers, and likewise traffic predictions may depend on traffic estimates. It is remarked that the currently available tools such as those available via Google Maps, apart from being costly, do not provide traffic predictions with reasonable accuracy [21]. The future traffic prediction problem becomes particularly challenging in regions with diverse modes of transport, such as in India, where ETA calculations must account for the multimodal nature of traffic [22], [23]. For instance the ETA calculations for buses should not only use traffic data meant for cars.

Construction of such spatio-temporal traffic maps from noisy and incomplete data is an ill-posed problem. In reality however, traffic densities are highly correlated across space and time [12]. A class of pertinent approaches have sought to visualize the traffic data as an incomplete matrix or tensor, and exploited this correlation to fill-in the missing entries [24], [25], [26], [12]. These approaches model the traffic matrix as belonging to an underlying low-dimensional subspace that can subsequently be recovered via matrix completion,

robust principal component analysis (PCA), or their tensor counterparts. Such techniques approach the problem from a static perspective. Specifically, data over one or more days is collected, and matrix or tensor completion is applied in order to impute the missing entries. In contrast, the traffic prediction problem is inherently dynamic, consisting of sequentially arriving traffic density measurements that must be handled in an online manner, and an underlying low-rank subspace that also evolves over time.

Complementary to these approaches, time-series modeling focuses on learning the temporal dynamics of traffic and generate predictions in an online manner [27]. While recent variants have incorporated spatial correlations as well, these techniques are generally unable to handle missing data or outliers. This work considers the first low-rank robust subspace filtering approach for online traffic imputation and prediction. Different from the existing matrix and tensor completion formulations, we consider low-rank traffic matrices whose underlying subspace evolves according to a state-space model. As columns of the traffic matrix arrive sequentially over time, the low rank components as well as the state-space model are learned in an online fashion using the variational Bayes formalism. In particular, component distributions are chosen to allow automatic relevance determination (ARD) and unlike the matrix or tensor completion works, the algorithm parameters such as rank, noise powers, and state noise powers need not be specified or tuned. A low-complexity forward-backward algorithm is also proposed that allows the updates to be carried out efficiently.. Enhancements to the proposed algorithm, capable of learning time-varying state-transition matrices, operating with a fixed lag, and robust to outliers, are also detailed.

The proposed algorithm is tested on traffic data collected over 200 square km. area within the city of New Delhi, India. The resulting matrix with more than 500 measurements per time instant is used for comparing the performance of the proposed algorithm with various state-of-the-art algorithms such as GROUSE [2] and LRTC [7]. The results show that modeling the evolution of the underlying subspace leads to accurate predictions and the low-complexity updates make the algorithm ideal for real-time applications. In summary, the contributions of the present work are as follows:

C. Related Work

Road traffic density modeling, estimation and prediction is of research interest to the transportation community since many years. Authors in [28] and [29] use auto regressive models to characterise the temporal variations in the traffic data. Likewise, the spatial correlation in the traffic data is captured using multivariate time series techniques in [30], [31]. Further, methods that exploit both temporal and spatial correlations were designed in [24], [32], [33]. Bayesian network based forecasting of traffic is proposed in [34], [35]. In the category of non-linear models, k-nearest-neighbour [36] and local least square (LLS) [37] have also been suggested to estimate the missing traffic data. Recently neural networks based approaches are also proposed in [38], [39], [40], [41].

Closer to our work, authors in [27] and [42] adopt an online approach to traffic-flow prediction. Specifically, authors in [27] use an adaptive Kalman filter by constructing state space models and perform short term traffic forecasting. However, both works [27] and [42] do not deal with the case of missing entries which is one of the main focus of this work. Missing data problem was tackled by [12] where the authors employ various matrix completion techniques to fill the missing entries in the traffic matrix, while [7], [26] use tensor completion algorithm to fill the missing data. Among many techniques, the authors in [12] use variational Bayesian principal component analysis (VBPCA), which is suitable for large-scale road networks.

Variational Bayesian approaches for matrix completion and robust principal component analysis are well known [3], [8], [9], [10], [11], [12], [13], [4], [14], [15]. One of the first works considered the measured matrix to be expressible as a product of low-rank matrices, associated with appropriate ARD priors [3] while faster algorithms for similar settings were proposed in [8], [9], [15]. More recently, other approaches towards modeling the measured matrices have also been proposed [10], [11].Moreover, variational Bayesian approaches have also been applied to road traffic estimation; see e.g. [12]. However, these approaches do not explicitly model the evolution of the underlying subspace. Likewise, none of the existing variational Bayesian approaches for low rank matrix completion model the evolution of the subspace [3], [4], [14], [15]. In contrast to these, the state-space modeling in our work is inspired from [13], where the low-complexity updates were first proposed in the context of linear dynamical models. The VBSF algorithm in the current work extends and generalizes that in [13] to incorporate low-rank structure and outliers.

This paper is organized as follows. Section II presents the Online Variational Bayesian subspace filtering method for traffic estimation and prediction. Section III presents the Online Robust Variational Bayesian subspace filtering method for traffic estimation and prediction in case of outliers. Results and findings for traffic estimation and prediction are discussed in section IV followed by conclusion in section V.

Notation: Scalars are denoted by letters in regular font, while vectors (matrices) are denoted by bold face (capital) letters. For a matrix $\mathbf{A}$, its transpose and trace are denoted by $\mathbf{A}^T$ and $\mathrm{tr}\,(\mathbf{A})$, respectively. The (i,j)-th element of a matrix $\mathbf{A}$ is denoted by a_{ij}, the i-th column by $\mathbf{a}_i$ or $[\mathbf{A}]_{\cdot i}$, and the i-th row by $\mathbf{a}_{i\cdot}^T$ or $[\mathbf{A}]_{i\cdot}^T$. The all-one vector of size $n \times 1$ is represented by $\mathbf{1}_n$, while $\mathbf{I}_n$ denotes identity matrix of size $n \times n$. The Frobenius norm for a matrix $\mathbf{A}$ and the Euclidean norm for a vector $\mathbf{a}$ are denoted by $\|\mathbf{A}\|$ and $\|\mathbf{a}\|$, respectively. The multivariate Gaussian probability density function (pdf) with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ evaluated at $\mathbf{x} \in \mathbb{R}^n$ is denoted by $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Likewise, $\mathrm{Ga}(x, a, b)$ denotes the Gamma pdf with parameters a_x and b_x evaluated at $x \in \mathbb{R}_+$. The expectation operator is symbolized by $\mathbb{E}$ while the pdf is generically denoted by $p(\cdot)$. Given data $\mathbf{D}$, we denote

Fig. 2: Region where traffic data is collected

$\hat{\mathbf{x}} := \mathbb{E}[\mathbf{x} \mid \mathbf{D}]$.

II. Variational Bayesian Subspace Filtering

The road network can be modeled using a directed graph where each edge represents a road segment and nodes represent intersections. Multiple lanes on the same road may be represented by different edges between two nodes. The average speed of vehicles on a particular segment is used as a proxy for the traffic density: a higher speed translates to a lower traffic and vice-versa. Traffic data is collected for the area shown in Fig. 2 using the Google map APIs. Unlike the complete data available from the API, real-world data may have missing entries. For instance, over the smaller area shown in Fig. 3, speed measurements may be available on the blue edges but not on the red ones. Prediction of such missing entries is one of the main goals of this paper.

Traffic data for m road segments is collected into the matrix $\mathbf{Y} \in \mathbb{R}^{m\times t}$, where t denotes the number of time instances over which measurements are made. For instance, if h measurements are made per day, we have that $t = dh$ after the end of d days. More generally, $\mathbf{Y}$ is an incomplete and growing matrix whose columns arrive sequentially over time. Specifically, for each column $\mathbf{y}_\tau$ with $1 \le \tau \le t$, only entries from the index set $\Omega_\tau \subset \{1, \ldots, m\}$ are observed. The algorithms developed here will seek to achieve the following two goals:

- *Imputation,* which yields $\{\hat{y}_{i\tau}\}_{i\notin\Omega_\tau}$ for $1 \le \tau \le t$
- *Prediction,* which yields $\{\hat{\mathbf{y}}_{t+\tau}\}_{\tau=1}^{T_p}$ where T_p is the prediction horizon.

The next subsection develops a variational Bayesian algorithm for achieving the aforementioned goals.

A. Hierarchical Bayesian Model

We begin with detailing a generative model for the traffic matrix $\mathbf{Y}$. The proposed model will not only capture the rank deficient nature of $\mathbf{Y}$ [43] but also the temporal correlation

Fig. 3: Map with red as missing and blue as known traffic entries

between successive columns of $\mathbf{Y}$ [44]. Recall that the standard low-rank parametrization of the full traffic matrix $\mathbf{Y}$ takes the form $\mathbf{Y} = \mathbf{AB}$ where $\mathbf{A} \in \mathbb{R}^{m\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times t}$. Classical non-negative matrix completion approaches seek to obtain such a factorization. In such algorithms, the choice of r is critical to avoiding underfitting or overfitting.

Within the Bayesian setting however, the measurements are modeled as arising from a noise distribution with unknown parameter, while various parameters are assigned different prior distributions. The Bayesian framework allows the use of ARD, wherein associating appropriate priors to the problem parameters leads to pruning of the redundant features [43]. This work uses pdfs from the exponential family that allow for tractable forms of the posterior pdf but are also flexible enough to adequately model the data.

Specifically, the entries of $\mathbf{Y}$ are generated from the following probability density function (pdf)

$$p(y_{i\tau} \mid \mathbf{a}_{i\cdot}, \mathbf{b}_\tau, \beta) = \mathcal{N}(\mathbf{y}_{i\tau} \mid \mathbf{b}_\tau^T \mathbf{a}_{i\cdot}, \beta^{-1}) \qquad i \in \Omega_\tau \tag{1}$$

for all $\tau \ge 1$, where $\mathbf{A} \in \mathbb{R}^{m\times r}$, $\mathbf{B} \in \mathbb{R}^{r\times t}$, and $\beta \in \mathbb{R}_{++}$ are the (hidden) problem parameters. Unlike the deterministic setting however, the rank parameter r is not critical to the imputation or prediction accuracy, but is only required to chosen according to computational considerations. The temporal evolution of the entries of $\mathbf{Y}$ is modeled by making the columns of $\mathbf{B}$ adhere to the following first order autoregressive model:

$$p(\mathbf{b}_\tau \mid \mathbf{J}, \mathbf{b}_{\tau-1}) = \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \qquad 2 \le \tau \le t \tag{2}$$

for $\tau \ge 2$, where $\mathbf{J} \in \mathbb{R}^{r\times r}$ is again a problem parameter. It follows from (2) that the conditional pdf of $\mathbf{b}_\tau$ given $\mathbf{J}$ is given by

$$p(\mathbf{B} \mid \mathbf{J}) = \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^{t} \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \tag{3}$$

The ARD priors to ensure that r can be learned in a data driven fashion are given by

$$p(\mathbf{A} \mid \boldsymbol{\gamma}) = \prod_{i=1}^{r} \mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1}\mathbf{I}_m) \tag{4}$$

$$p(\mathbf{J} \mid \boldsymbol{\upsilon}) = \prod_{i=1}^{r} \mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1}\mathbf{I}_r) \tag{5}$$

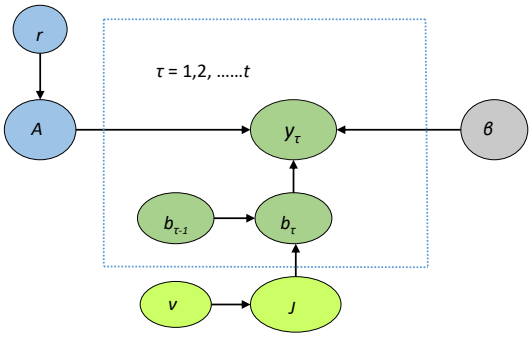

Fig. 4: Hierarchical Bayesian Model for Matrix Completion

where the precisions $\boldsymbol{\gamma}$ and $\boldsymbol{\upsilon}$ are problem parameters. Many of the precisions $\{\gamma_i, \upsilon_i\}$ will generally assume large values during the inference, effectively removing the corresponding columns from $\mathbf{A}$ and $\mathbf{J}$ respectively. Note that although the number of relevant columns in $\mathbf{A}$ and $\mathbf{J}$ should be the same, the non-zero precisions γ_i and υ_i may still be different from each other.

Finally, the three precision variables are selected to have have non-informative Jeffrey's priors

$$p(\beta) = \frac{1}{\beta}, \qquad p(\gamma_i) = \frac{1}{\gamma_i}, \qquad p(\upsilon_i) = \frac{1}{\upsilon_i} \tag{6}$$

for $1 \leq i \leq r$. Let $\mathbf{y}_\Omega$ denote the collection of measurements $\{y_{i\tau}\}_{i\in\Omega_\tau,\tau=1}^{t}$. Collecting the hidden variables into $\mathcal{H} := \{\mathbf{A}, \mathbf{B}, \mathbf{J}, \beta, \boldsymbol{\gamma}, \boldsymbol{\upsilon}\}$, the joint distribution of $\{\mathbf{y}_\Omega, \mathcal{H}\}$ can be written as

$$\begin{aligned} p(\mathbf{y}_\Omega, \mathcal{H}) &= p(\mathbf{y}_\Omega|\mathbf{A}, \mathbf{B}, \beta)p(\mathbf{A}|\boldsymbol{\gamma})p(\mathbf{B}|\mathbf{J})p(\mathbf{J}|\boldsymbol{\upsilon})p(\beta)p(\boldsymbol{\upsilon})p(\boldsymbol{\gamma}) \\ &= \prod_{\tau=1}^{t}\prod_{i\in\Omega_\tau} \mathcal{N}(y_{i\tau} \mid \mathbf{b}_\tau^T \mathbf{a}_{i\cdot}, \beta^{-1}) \\ &\quad \times \prod_{i=1}^{r} \left[\mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1}\mathbf{I}_m)\mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1}\mathbf{I}_r)\right] \\ &\quad \times \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^{t} \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \frac{1}{\beta} \prod_{i=1}^{r} \frac{1}{\gamma_i \upsilon_i} \end{aligned} \tag{7}$$

The full hierarchical Bayesian model adopted here is summarized in Fig. 4.

B. Variational Bayesian Inference

Since exact full Bayesian inference is intractable, we make use of the mean field approximation, wherein the posterior distribution $p(\mathcal{H} \mid \mathbf{y}_\Omega)$ factorizes as

$$p(\mathcal{H} \mid \mathbf{y}_\Omega) \approx q(\mathcal{H}) = q_{\mathbf{A}}(\mathbf{A}) q_{\mathbf{B}}(\mathbf{B}) q_{\mathbf{J}}(\mathbf{J}) q_{\boldsymbol{\upsilon}}(\boldsymbol{\upsilon}) q_\beta(\boldsymbol{\beta}) q_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}). \tag{8}$$

Under the mean-field approximation, the variational lower bound can be maximized via coordinate ascent iterations [45]. Indeed, thanks to the choice of conjugate priors for the parameters, it can be shown that the individual factors in (8) take the following forms:

$$q_{\mathbf{B}}(\mathbf{B}) = \mathcal{N}(\mathrm{vec}\,(\mathbf{B}) \mid \boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{B}}) \tag{9a}$$
$$q_{\mathbf{a}_{i\cdot}} = \mathcal{N}(\mathbf{a}_{i\cdot} \mid \boldsymbol{\mu}_i^{\mathbf{A}}, \boldsymbol{\Xi}_i^{\mathbf{A}}) \tag{9b}$$
$$q_{\mathbf{j}_{i\cdot}} = \mathcal{N}(\mathbf{j}_{i\cdot} \mid \boldsymbol{\mu}_i^{\mathbf{J}}, \boldsymbol{\Xi}_i^{\mathbf{J}}) \tag{9c}$$
$$q_\beta(\beta) = \mathrm{Ga}(\beta; a^\beta, b^\beta) \tag{9d}$$
$$q_{\gamma_i}(\gamma_i) = \mathrm{Ga}(\gamma_i; a_i^\gamma, b_i^\gamma) \tag{9e}$$
$$q_{\upsilon_i}(\upsilon_i) = \mathrm{Ga}(\upsilon_i; a_i^\upsilon, b_i^\upsilon) \tag{9f}$$

where $\boldsymbol{\mu}^{\mathbf{B}} \in \mathbb{R}^{rt}$, $\boldsymbol{\Xi}^{\mathbf{B}} \in \mathbb{R}^{rt\times rt}$, $\boldsymbol{\mu}_i^{\mathbf{A}} \in \mathbb{R}^r$, $\boldsymbol{\Xi}_i^{\mathbf{A}} \in \mathbb{R}^{r\times r}$, $\boldsymbol{\mu}_i^{\mathbf{J}} \in \mathbb{R}^r$, $\boldsymbol{\Xi}_i^{\mathbf{J}} \in \mathbb{R}^{r\times r}$, and $a^\beta, b^\beta, a_i^\gamma, b_i^\gamma, a_i^\upsilon, b_i^\upsilon \in \mathbb{R}_{++}$. Consequently, each iteration of coordinate ascent simply involves updating the variables $\{\boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{B}}, \{\boldsymbol{\mu}_i^{\mathbf{A}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{A}}\}, \{\boldsymbol{\mu}_i^{\mathbf{J}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{J}}\}, a^\beta, b^\beta, \{a_i^\gamma\}, \{b_i^\gamma\}, \{a_i^\upsilon\}, \{b_i^\upsilon\}\}$ in a cyclic manner.

In the present case, not all variables need to be updated explicitly and the updates may be written in a compact form. Let us denote $\omega_\tau := |\Omega_\tau|$ and let $\omega := \sum_\tau \omega_\tau$ be the total number of observations made. Then, the updates for hyperparameters $\{\boldsymbol{\upsilon}, \boldsymbol{\gamma}\}$ take the following form

$$\hat{\upsilon}_i = \frac{m}{\sum_{k=1}^{m}\left([\boldsymbol{\mu}_k^{\mathbf{J}}]_i^2 + [\boldsymbol{\Xi}_k^{\mathbf{J}}]_{ii}\right)} \tag{10a}$$
$$\hat{\gamma}_i = \frac{m}{\sum_{k=1}^{m}\left([\boldsymbol{\mu}_k^{\mathbf{A}}]_i^2 + [\boldsymbol{\Sigma}_k^{\mathbf{A}}]_{ii}\right)} \tag{10b}$$

Subsequently, let $\hat{\boldsymbol{\upsilon}}$ and $\hat{\boldsymbol{\gamma}}$ be the vectors that collect $\{\hat{\upsilon}_i\}$ and $\{\hat{\gamma}_i\}$, respectively. Since $\mathbf{b}_\tau$ denotes the τ-th column of $\mathbf{B}^T$, its posterior distribution may be written as $q_{\mathbf{b}_\tau}(\mathbf{b}_\tau) = \mathcal{N}(\mathbf{b}_\tau \mid \boldsymbol{\mu}_\tau^{\mathbf{B}}, \boldsymbol{\Xi}_\tau^{\mathbf{B}})$, where $\boldsymbol{\mu}_\tau^{\mathbf{B}}$ and $\boldsymbol{\Xi}_\tau^{\mathbf{B}}$ comprise of the corresponding elements of $\boldsymbol{\mu}^{\mathbf{B}}$ and $\boldsymbol{\Xi}^{\mathbf{B}}$, respectively. Also define the posterior covariance matrices

$$\boldsymbol{\Sigma}_{\tau,\iota}^{\mathbf{B}} := \boldsymbol{\mu}_\tau^{\mathbf{B}}(\boldsymbol{\mu}_\iota^{\mathbf{B}})^T + \boldsymbol{\Xi}_{\tau,\iota}^{\mathbf{B}} \tag{11}$$
$$\boldsymbol{\Sigma}_i^{\mathbf{J}} := \boldsymbol{\mu}_i^{\mathbf{J}}(\boldsymbol{\mu}_i^{\mathbf{J}})^T + \boldsymbol{\Xi}_i^{\mathbf{J}} \tag{12}$$
$$\boldsymbol{\Sigma}_i^{\mathbf{A}} := \boldsymbol{\mu}_i^{\mathbf{A}}(\boldsymbol{\mu}_i^{\mathbf{A}})^T + \boldsymbol{\Xi}_i^{\mathbf{A}} \tag{13}$$

Subsequently, the update for $\hat{\beta}$ becomes

$$\hat{\beta} = \frac{\omega}{\sum_{\tau=1}^{t}\sum_{i\in\Omega_\tau}\left[y_{i\tau}^2 - 2y_{i\tau}(\boldsymbol{\mu}_i^{\mathbf{A}})^T\boldsymbol{\mu}_\tau^{\mathbf{B}} + \mathrm{tr}\left(\boldsymbol{\Sigma}_i^{\mathbf{A}}\boldsymbol{\Sigma}_{\tau,\tau}^{\mathbf{B}}\right)\right]} \tag{14}$$

Next, the updates for the factors $\mathbf{J}$ and $\mathbf{A}$ take the following form

$$\boldsymbol{\mu}_i^{\mathbf{J}} = [\boldsymbol{\Xi}_i^{\mathbf{J}}\boldsymbol{\Sigma}_{\tau,\tau-1}^{\mathbf{B}}]_{\cdot i} \tag{15a}$$
$$\boldsymbol{\Xi}_i^{\mathbf{J}} = \left(\mathrm{Diag}\,(\hat{\boldsymbol{\upsilon}}) + \sum_{\tau=1}^{t-1}\boldsymbol{\Sigma}_{\tau,\tau-1}^{\mathbf{B}}\right)^{-1} \tag{15b}$$
$$\boldsymbol{\mu}_i^{\mathbf{A}} = \hat{\beta}\boldsymbol{\Xi}_i^{\mathbf{A}}\sum_{\tau\in\Omega_i'}\boldsymbol{\mu}_\tau^{\mathbf{B}} y_{i\tau} \tag{15c}$$
$$\boldsymbol{\Xi}_i^{\mathbf{A}} = \left(\hat{\gamma}_i\mathbf{I}_r + \hat{\beta}\sum_{\tau\in\Omega_i'}\boldsymbol{\Sigma}_{\tau,\tau}^{\mathbf{B}}\right)^{-1} \tag{15d}$$

where $\Omega_i' := \{\tau \mid i \in \Omega_\tau\}$. Observe from the updates that the rows of $\mathbf{J}$ are independent identically distributed under the

mean field approximation. The update for $\boldsymbol{\mu}^{\mathbf{B}}$ can be written as

$$\boldsymbol{\mu}^{\mathbf{B}} = \boldsymbol{\Xi}^{\mathbf{B}} \begin{bmatrix} \hat{\beta} \sum_{i\in\Omega_1} y_{i1} \boldsymbol{\mu}_i^{\mathbf{A}} + \boldsymbol{\Lambda}_1^{-1} \boldsymbol{\mu}_1 \\ \hat{\beta} \sum_{i\in\Omega_2} y_{i2} \boldsymbol{\mu}_i^{\mathbf{A}} \\ \vdots \\ \hat{\beta} \sum_{i\in\Omega_t} y_{it} \boldsymbol{\mu}_i^{\mathbf{A}} \end{bmatrix} \tag{16}$$

Finally, $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1}$ a block-tridiagonal matrix. Defining $\hat{\mathbf{J}} := \mathbb{E}[\mathbf{J} \mid \mathbf{y}_\Omega]$ as the matrix whose i-row is given by $(\boldsymbol{\mu}_i^{\mathbf{J}})^T$, $\boldsymbol{\Sigma}_{(\tau)}^{\mathbf{A}} = \sum_{i\in\Omega_\tau'} \boldsymbol{\Sigma}_i^{\mathbf{A}}$, and $\boldsymbol{\Sigma}^{\mathbf{J}} := \sum_{i=1}^r \boldsymbol{\Sigma}_i^{\mathbf{J}}$, the updates take the form:

$$\left[\boldsymbol{\Xi}^{\mathbf{B}}\right]^{-1} = \hat{\beta}\text{Diag}\left(\boldsymbol{\Xi}_{(1)}^{\mathbf{A}}, \ldots, \boldsymbol{\Xi}_{(t)}^{\mathbf{A}}\right) + \\ + \begin{bmatrix} \boldsymbol{\Lambda}_1^{-1} & -\hat{\mathbf{J}} & \ldots & 0 \\ -\hat{\mathbf{J}} & \mathbf{I}_r + \boldsymbol{\Sigma}^{\mathbf{J}} & -\hat{\mathbf{J}} & \ldots \\ \vdots & \vdots & & \vdots \\ \ldots & 0 & -\hat{\mathbf{J}} & \mathbf{I}_r \end{bmatrix}. \tag{17}$$

It is remarked that although the $rt \times rt$ matrix $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1}$ is block-tridiagonal, the matrix $\boldsymbol{\Xi}^{\mathbf{B}}$ is dense, and direct inversion would be prohibitively costly. Moreover, the classical Rauch-Tung-Striebel (RTS) smoother cannot be directly applied as the evaluation since evaluating the conditional expectations under $q(\mathbf{B})$ is difficult and not amenable to the Matrix Inversion Lemma [46]. Interestingly, observe that the updates in (14) and (15) depend only on diagonal and super-diagonal blocks of $\boldsymbol{\Xi}^{\mathbf{B}}$, namely $\boldsymbol{\Xi}_{\tau,\tau}^{\mathbf{B}}$ and $\boldsymbol{\Xi}_{\tau,\tau-1}^{\mathbf{B}}$, respectively. The next subsection details a low-complexity algorithm for carrying out the updates for these blocks as well as for $\boldsymbol{\mu}^{\mathbf{B}}$.

C. Low-complexity updates via LDL-decomposition

Thanks to the block-tridiagonal structure of $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1}$, it is possible to use the LDL decomposition to carry out the updates in an efficient manner. Decomposing $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, the key idea is that left multiplication with $\boldsymbol{\Xi}^{\mathbf{B}}$ is equivalent to left multiplication with $\mathbf{L}^{-T}\mathbf{D}^{-1}\mathbf{L}^{-1}$. Towards this end, we utilize the algorithm from [47], that comprises of two phases: the forward pass that carries out the multiplication with $\mathbf{D}^{-1}\mathbf{L}^{-1}$ and the backward pass that implements the multiplication with $\mathbf{L}^{-T}$. Let us define for $2 \leq \tau \leq t$,

$$\boldsymbol{\Psi}_\tau := \hat{\beta} \sum_{i\in\Omega_\tau} \boldsymbol{\Sigma}_{(i)}^{\mathbf{A}} + \mathbf{I}_r + \mathbf{1}_{\tau\neq t} \sum_{i=1}^r \boldsymbol{\Sigma}_i^{\mathbf{J}} \tag{18}$$

$$\mathbf{v}_\tau := \hat{\beta} \sum_{i\in\Omega_\tau} y_{i\tau} \boldsymbol{\mu}_i^{\mathbf{A}}. \tag{19}$$

The forward pass outputs intermediate variables $\breve{\boldsymbol{\Xi}}_{\tau,\tau}^{\mathbf{B}}$, $\breve{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}}$, and $\breve{\boldsymbol{\mu}}_\tau$, that are subsequently used in the backward pass. The updates take the following form:

1) Initialize $\hat{\boldsymbol{\Xi}}_{1,1}^{\mathbf{B}} = \boldsymbol{\Lambda}_1$ and $\hat{\boldsymbol{\mu}}_1^{\mathbf{B}} = \boldsymbol{\mu}_1 + \hat{\beta} \sum_{i\in\Omega_\tau} y_{i\tau} \boldsymbol{\Lambda}_1 \boldsymbol{\mu}_i^{\mathbf{A}}$
2) For $\tau = 1, \ldots, t-1$

$$\breve{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}} = -\hat{\boldsymbol{\Xi}}_{\tau,\tau}^{\mathbf{B}} \hat{\mathbf{J}} \tag{20a}$$

$$\breve{\boldsymbol{\Xi}}_{\tau+1,\tau+1}^{\mathbf{B}} = (\boldsymbol{\Psi}_{\tau+1} - (\breve{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}})^T \boldsymbol{\Psi}_{\tau,\tau+1}^{\mathbf{B}})^{-1} \tag{20b}$$

$$\breve{\boldsymbol{\mu}}_{\tau+1}^{\mathbf{B}} = \breve{\boldsymbol{\Xi}}_{\tau+1,\tau+1}^{\mathbf{B}} (\mathbf{v}_{\tau+1} - (\breve{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}})^T \breve{\boldsymbol{\mu}}_\tau^{\mathbf{B}}) \tag{20c}$$

3) For $\tau = t-1, \ldots, 1$

$$\boldsymbol{\Xi}_{\tau,\tau+1}^{\mathbf{B}} = -\breve{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}} \boldsymbol{\Xi}_{\tau+1,\tau+1}^{\mathbf{B}} \tag{20d}$$

$$\boldsymbol{\Xi}_{\tau,\tau}^{\mathbf{B}} = \breve{\boldsymbol{\Xi}}_{\tau,\tau}^{\mathbf{B}} - \hat{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}} (\boldsymbol{\Xi}_{\tau,\tau+1}^{\mathbf{B}})^T \tag{20e}$$

$$\boldsymbol{\mu}_\tau^{\mathbf{B}} = \breve{\boldsymbol{\mu}}_\tau^{\mathbf{B}} - \breve{\boldsymbol{\Xi}}_{\tau,\tau+1}^{\mathbf{B}} \boldsymbol{\mu}_{\tau+1}^{\mathbf{B}} \tag{20f}$$

4) Output $\{\boldsymbol{\Xi}_{\tau,\tau+1}^{\mathbf{B}}, \boldsymbol{\Xi}_{\tau,\tau}^{\mathbf{B}}, \boldsymbol{\mu}_\tau^{\mathbf{B}}\}_{\tau=2}^t$

Note that while $\boldsymbol{\Xi}_{i,j}^{\mathbf{B}} \neq 0$ for $|i-j| > 1$, these blocks are neither calculated in the forward and backward passes nor required in any of the variational updates.

Finally, the predictive distribution $p(y_{i\tau} \mid \mathbf{y}_\Omega)$ for $\tau \notin \Omega_i$ or $\tau \geq t+1$ is still not tractable in the present case. Instead, we simply use point estimates for estimating the missing entries. Specifically, for $\tau \notin \Omega_i$, the missing entries are imputed as

$$y_{\imath\tau} = (\boldsymbol{\mu}_\tau^{\mathbf{B}})^T \boldsymbol{\mu}_i^{\mathbf{A}}. \tag{21}$$

Likewise for $\tau \geq t+1$, the prediction becomes

$$y_{\imath\tau} = (\hat{\mathbf{J}}^{\tau-t} \boldsymbol{\mu}_t^{\mathbf{B}})^T \boldsymbol{\mu}_i^{\mathbf{A}}. \tag{22}$$

It can be seen that as compared to the updates in (16)-(17) that incur a complexity of $\mathcal{O}(t^3)$, the complexity incurred due to (20) is only $\mathcal{O}(t)$. Overall, the different parameters are updated cyclically until convergence for each $t = 1, 2, \ldots$.

D. EM Baysian Subspace Filtering

Different from the variational Bayesian framework used here, the EM algorithm treats $\mathcal{H}_h := \{\mathbf{A}, \mathbf{B}, \mathbf{J}\}$ as hidden variables (with posterior pdf $q_h(\mathcal{H}_h) := q_{\mathbf{B}}(\mathbf{B}) q_{\mathbf{A}}(\mathbf{A}) q_{\mathbf{J}}(\mathbf{J})$) and uses maximum a posteriori (MAP) estimates for the precision variables $\mathcal{H}_p := \{\boldsymbol{\upsilon}, \boldsymbol{\gamma}, \beta\}$. Consequently, the EM algorithm for Bayesian subspace tracking starts with an initial estimate $\mathcal{H}_p^{(0)}$ and uses the following updates at iteration $\iota \geq 1$,

- **E-step:** evaluate

$$Q(\mathcal{H}_p, \mathcal{H}_p^{(\iota)}) := \mathbb{E}_{q_h(\mathcal{H}_h)} \left[\log p(\mathbf{y}_\Omega, \mathcal{H}_h, \mathcal{H}_p^{(\iota)})\right] \tag{23}$$

- **M-step:** maximize

$$\mathcal{H}_p^{(\iota+1)} = \arg\max_{\mathcal{H}_p} Q(\mathcal{H}_p, \mathcal{H}_p^{(\iota)}) \tag{24}$$

Interestingly, the updates resulting from the E-step take the same form as those in (15) and (20). On the other hand, the updates obtained from solving the M step take the slightly different form:

$$\hat{\upsilon}_i = \frac{m-2}{\sum_{k=1}^m \left([\boldsymbol{\mu}_k^{\mathbf{J}}]_i^2 + [\boldsymbol{\Xi}_k^{\mathbf{J}}]_{ii}\right)} \tag{25a}$$

$$\hat{\gamma}_i = \frac{m-2}{\sum_{k=1}^m \left([\boldsymbol{\mu}_k^{\mathbf{A}}]_i^2 + [\boldsymbol{\Sigma}_k^{\mathbf{A}}]_{ii}\right)} \tag{25b}$$

$$\hat{\beta} = \frac{\omega - 2}{\sum_{\tau=1}^t \sum_{i\in\Omega_\tau} \left[y_{i\tau}^2 - 2y_{i\tau} (\boldsymbol{\mu}_i^{\mathbf{A}})^T \boldsymbol{\mu}_\tau^{\mathbf{B}} + \text{tr}\left(\boldsymbol{\Sigma}_i^{\mathbf{A}} \boldsymbol{\Sigma}_\tau^{\mathbf{B}}\right)\right]} \tag{25c}$$

The slight differences arise due to the difference between the mean and mode of the Gamma distribution. Specifically, for $p(x) = \text{Ga}(x|a,b)$, it holds that $\mathbb{E}[X] = a/b$ while $\max_x \text{Ga}(x|a,b) = \frac{a-1}{b}$.

E. Handling time-varying dynamics

For time-series data that are not collected in a continuous manner, not all pairs of measurements may exhibit the same amount of correlation. For instance, in the present context, traffic data is only collected from 07:00 hrs to 23:00 hrs. That is, although the measurement vector $\mathbf{y}_\tau$ collected at 23:00 hrs on a certain day d are placed just before the measurement data $\mathbf{y}_{\tau+1}$ collected at 07:00 on day $d+1$, the correlation between them may be small. The present framework however allows for a time-varying dynamical model $\mathbf{b}_{\tau+1} = \mathbf{J}_\tau \mathbf{b}_\tau$ where we set

$$\mathbf{J}_\tau = \begin{cases} \mathbf{J}_1 & \tau \neq dh \\ \mathbf{J}_2 & \tau = dh \end{cases} \tag{26}$$

where $\mathbf{J}_1, \mathbf{J}_2$ are problem parameters while h is the number of samples taken per day. Note however that such a modification increases the size of the problem. Further, accurate estimation of $\mathbf{J}_2$ requires the data to have sufficient number of such transitions so the that appropriate transition matrix may be learnt. Associating similar prior distributions as in (28c), the updates for this case can be derived in a similar manner and are omitted for the sake of brevity.

F. Fixed-lag tracking

Algorithm 1 can be viewed as an offline algorithm that must be run for every t. In practical settings, it may be impractical to remember and process the entire history of measurements at each t. Moreover, given traffic at time t, traffic estimates may only be required starting from $t-\Delta$ for some $\Delta < h$. Towards this end, we consider a sliding window of measurements. Since $\mathbf{A}_t$ and $\mathbf{J}_t$ may be seen as transition matrices for the latent states and between latent state and observations, we initialize the next sliding-window with inferred approximate distributions on the transition matrices of the current window. One way to exploit periodicity in the data is to use the inferred approximate distribution for a day as the prior distribution for the coming days. That is the distributions for $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{J}$ for a day and sliding window can be initialized with the approximate distribution of previous month's data.

Algorithm 1: Variational Bayesian Subspace Filtering

Initialize $\boldsymbol{\gamma}, \boldsymbol{\beta}, \mathbf{v}$,
$sub = 1$, Ω_τ, Ω'_i, $\boldsymbol{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^A, \boldsymbol{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^B, \boldsymbol{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^J \boldsymbol{\Lambda}_1, \mu_1$,
$\hat{\mathbf{Y}} = \boldsymbol{\mu}^A (\boldsymbol{\mu}^{\mathbf{B}})^T$
while $Y_{conv} < 10^{-5}$ **do**
  $\mathbf{Y_{old}} = \hat{\mathbf{Y}}$
  $\boldsymbol{\Gamma} = diag(\boldsymbol{\gamma})$
  if $sub == 1$ **then**
    Update using (20)
    $sub = 2$
    Update using (10a), (11), (15a), (15b) $\forall\ 1 \leq i \leq r$
  else if $sub == 2$ **then**
    Update using (13), (15c), (15d), (10b) $\forall 1 \leq i \leq m$
    $sub = 1$
  end
  $\hat{\mathbf{Y}} = \boldsymbol{\mu}^A (\boldsymbol{\mu}^{\mathbf{B}})^T$
  Update using (14)
  $Y_{conv} = \frac{\|\mathbf{Y} - \mathbf{Y_{old}}\|_F}{\|\mathbf{Y_{old}}\|_F}$
end
return $(\hat{\mathbf{Y}}, \boldsymbol{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^A, \boldsymbol{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^B, \boldsymbol{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^J)$

III. Robust Variational Bayesian Subspace Filtering

In this section we consider the robust version of the variational Bayesian subspace filtering problem in Sec. II. Within this context, in addition to the missing entries in $\mathbf{Y}$, some entries of $\mathbf{Y}$ are also contaminated with outliers. Unlike the missing entries however, the location of these outliers is not known. In the traffic prediction problem, such entries arise due to sensor malfunctions, communication errors, and impulse noise. The robust subspace filtering problem is more difficult as the removal of such outliers entails estimating their magnitudes as well as locations.

Within the deterministic robust PCA framework, the traffic matrix is modeled as taking the form $\mathbf{Y} = \mathbf{AB} + \mathbf{E}$ where $\mathbf{A} \in \mathbb{R}^{m \times r}$, $\mathbf{B} \in \mathbb{R}^{r \times t}$ are low-rank matrices as before. Additionally, we also need to estimate the sparse outlier matrix $\mathbf{E} \in \mathbb{R}^{m \times t}$. As before, both r and the level of sparsity in $\mathbf{E}$ are tuning parameters that must generally be carefully selected.

Here, we put forth the variational Bayesian subspace filtering algorithm that makes use of ARD priors to prune the redundant features. Consider the measurement matrix $\mathbf{Y}$, whose entries are generated from the following pdf:

$$p(y_{i\tau} \mid \mathbf{a}_{i\cdot}, \mathbf{b}_\tau, e_{i\tau}, \beta) = \mathcal{N}(\mathbf{y}_{i\tau} \mid \mathbf{b}_\tau^T \mathbf{a}_{i\cdot} + e_{i\tau}, \beta^{-1}) \quad i \in \Omega_\tau \tag{27}$$

for all $\tau \geq 1$, and apart from the matrices $\mathbf{A}$ and $\mathbf{B}$ defined earlier, we also have $\{e_{i\tau}\}_{\tau=1, i\in\Omega_\tau}^t$ as the additional (hidden) problem parameter that captures the outliers. The generative models for $\mathbf{A}$ and $\mathbf{B}$ are the same as before, i.e.,

$$p(\mathbf{B} \mid \mathbf{J}) = \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^{t} \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \tag{28a}$$

$$p(\mathbf{A} \mid \boldsymbol{\gamma}) = \prod_{i=1}^{r} \mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1}\mathbf{I}) \tag{28b}$$

$$p(\mathbf{J} \mid \boldsymbol{\upsilon}) = \prod_{i=1}^{r} \mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1}\mathbf{I}) \tag{28c}$$

for $\tau \geq 2$, and $\boldsymbol{\gamma}$ and $\boldsymbol{\upsilon}$ are problem parameters. Additionally, we also associate an ARD prior to the outliers, i.e.,

$$p(e_{i\tau}) = \mathcal{N}(e_{i\tau} \mid 0, \alpha_{i\tau}^{-1}) \qquad i \in \Omega_\tau \tag{29}$$

for $1 \leq \tau \leq t$, where the precision $\alpha_{i\tau}$ is a hidden variable, that would be driven to infinity whenever e_{ij} is zero. It is remarked that the prior for $e_{i\tau}$ is only specified for the measurements, i.e., for $i \in \Omega_\tau$ and no predictions are made

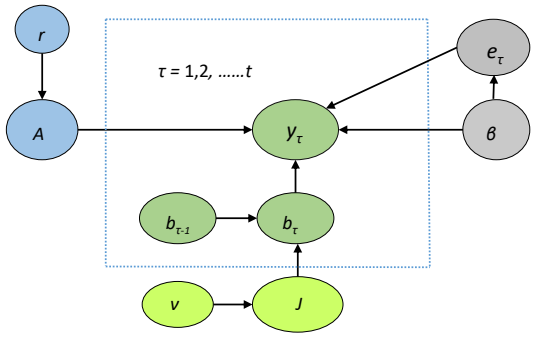

Fig. 5: Hierarchical Bayesian Model for Robust Matrix Completion

for the outliers. As before, we associate Jeffery's prior to the precisions β, $\{\gamma_i\}$, $\{\upsilon_i\}$, and $\{\alpha_{i\tau}\}$.

$$p(\beta) = \frac{1}{\beta}, \quad p(\gamma_i) = \frac{1}{\gamma_i}, \quad p(\upsilon_i) = \frac{1}{\upsilon_i}, \quad p(\alpha_{i\tau}) = \frac{1}{\alpha_{i\tau}} \tag{30}$$

Let the vectors $\mathbf{e} \in \mathbb{R}^{\omega}$and $\boldsymbol{\alpha} \in \mathbb{R}^{\omega}$ collect the variables $\{e_{i\tau}\}$ and $\{\alpha_{i\tau}\}$, respectively. Likewise, defining all the hidden variables as $\mathcal{H} := \{\mathbf{A}, \mathbf{B}, \mathbf{J}, \mathbf{e}, \beta, \boldsymbol{\gamma}, \boldsymbol{\upsilon}\}$, the joint distribution of $\{\mathbf{y}_\Omega, \mathcal{H}\}$ can be written as

$$\begin{aligned}
&p(\mathbf{y}_\Omega, \mathcal{H}) \\
&= p(\mathbf{y}_\Omega|\mathbf{A}, \mathbf{B}, \beta)p(\mathbf{A}|\boldsymbol{\gamma})p(\mathbf{B}|\mathbf{J})p(\mathbf{J}|\boldsymbol{\upsilon})p(\mathbf{e}|\boldsymbol{\alpha})p(\beta)p(\boldsymbol{\upsilon})p(\boldsymbol{\gamma}) \\
&= \prod_{\tau=1}^{t}\prod_{i\in\Omega_\tau} \mathcal{N}(y_{i\tau} \mid \mathbf{b}_\tau^T \mathbf{a}_{i\cdot}, \beta^{-1})\mathcal{N}(e_{i\tau} \mid 0, \alpha_{i\tau}^{-1})\frac{1}{\alpha_{i\tau}} \\
&\quad \times \prod_{i=1}^{r}\left[\mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1}\mathbf{I})\mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1}\mathbf{I})\right] \\
&\quad \times \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1)\prod_{\tau=2}^{t}\mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I})\frac{1}{\beta}\prod_{i=1}^{r}\frac{1}{\gamma_i\upsilon_i}
\end{aligned} \tag{31}$$

The full hierarchical Bayesian model adopted here is summarized in figure 5.

A. Variational Bayesian Inference

Utilizing the mean field approximation, the posterior distribution $p(\mathcal{H} \mid \mathbf{y}_\Omega)$ factorizes as

$$p(\mathcal{H} \mid \mathbf{y}_\Omega) \approx q(\mathcal{H}) = q_{\mathbf{A}}(\mathbf{A})q_{\mathbf{B}}(\mathbf{B})q_{\mathbf{J}}(\mathbf{J})q_{\mathbf{e}}(\mathbf{e})q_{\boldsymbol{\upsilon}}(\boldsymbol{\upsilon})q_{\beta}(\boldsymbol{\beta})q_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}) \tag{32}$$

where the individual factors take the same forms as in (9), in addition to

$$q_{\mathbf{e}}(\mathbf{e}) = \prod_{\tau=1}^{t}\prod_{i\in\Omega_\tau}\mathcal{N}(e_{i\tau}|\mu_e^{i\tau}, \Xi_e^{i\tau}) \tag{33}$$

As before, the variational inference problem can be solved by updating the variables $\{\boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{B}}, \{\boldsymbol{\mu}_i^{\mathbf{A}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{A}}\}, \{\boldsymbol{\mu}_i^{\mathbf{J}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{J}}\}, \{\mu_e^{i\tau}\}, \{\Xi_e^{i\tau}\}, a^\beta, b^\beta, \{a_i^\gamma\}, \{b_i^\gamma\}, \{a_i^\upsilon\}, \{b_i^\upsilon\}\}$ in a cyclic manner. However, a more compact form for the updates may be derived as follows.

Specifically, the updates for $\{\hat{\upsilon}_i, \hat{\gamma}_i\}$ remain the same as in (10). However, the update for $\hat{\beta}$ takes the form:

$$\hat{\beta} = \frac{\omega}{\sum_{\tau=1}^{t}\sum_{i\in\Omega_\tau}\nu_{i\tau}} \tag{34}$$

where,

$$\begin{aligned}
\nu_{i\tau} :=& y_{i\tau}^2 - 2(y_{i\tau} - \mu_e^{i\tau})(\boldsymbol{\mu}_i^{\mathbf{A}})^T\boldsymbol{\mu}_\tau^{\mathbf{B}} - 2y_{i\tau}\mu_e^{i\tau} \\
&+ (\mu_e^{i\tau})^2 + \Xi_e^{i\tau} + \mathrm{tr}\left(\boldsymbol{\Sigma}_i^{\mathbf{A}}\boldsymbol{\Sigma}_{\tau,\tau}^{\mathbf{B}}\right)
\end{aligned} \tag{35}$$

Further, the parameters $\mu_e^{i\tau}$ and $\Xi_e^{i\tau}$ are updated as

$$\Xi_e^{i\tau} = \frac{1}{\hat{\beta} + (\mu_e^{i\tau})^2 + \Xi_e^{i\tau}} \tag{36a}$$

$$\mu_e^{i\tau} = \hat{\beta}\Xi_e^{i\tau}(y_{i\tau} - (\boldsymbol{\mu}_i^{\mathbf{A}})^T\boldsymbol{\mu}_\tau^{\mathbf{B}}) \tag{36b}$$

Proceeding similarly, the updates for $\{\boldsymbol{\mu}_i^{\mathbf{J}}\}$, $\{\boldsymbol{\Xi}_i^{\mathbf{J}}\}$, and $\{\boldsymbol{\Xi}_i^{\mathbf{A}}\}$ remain the same as in (15), while the updates for $\{\boldsymbol{\mu}_i^{\mathbf{A}}\}$ become:

$$\boldsymbol{\mu}_i^{\mathbf{A}} = \hat{\beta}\boldsymbol{\Xi}_i^{\mathbf{A}}\sum_{\tau\in\Omega_i'}\boldsymbol{\mu}_\tau^{\mathbf{B}}(y_{i\tau} - \mu_e^{i\tau}) \tag{37}$$

Finally, the updates for $\boldsymbol{\Xi}^{\mathbf{B}}$ remain the same but the updates of $\boldsymbol{\mu}^{\mathbf{B}}$ change. Specifically, the low complexity updates via LDL-decomposition remain mostly the same, except for the modified definition of $\mathbf{v}_\tau$ in (19) which now looks like

$$\mathbf{v}_\tau = \hat{\beta}\sum_{i\in\Omega_\tau}(y_{i\tau} - \mu_e^{i\tau}) \tag{38}$$

The full robust subspace filtering algorithm is summarized in Algorithm 2. The predictions for $y_{i\tau}$ for $i \notin \Omega_\tau$ and for $\tau \geq t+1$ are obtained as in (21) and (22), respectively.

IV. Results

We now discuss the performance of the proposed VBSF algorithm for the twin tasks of real time traffic estimation as well as future traffic prediction in a road network. Future traffic prediction can be used to estimate the expected time of arrival (ETA) for which we provide simulation results as well. To evaluate the VBSF algorithm, we use the partial road network of the city of New Delhi with an area of 200 square kms consisting of $m = 519$ edges (shown in Fig. 2). To estimate the model parameters and for testing purposes, speed data was collected using the Google Maps API for nearly 3 months across all the 519 edges. Taking advantage of the slow varying nature of the speed in the network edges, we sample the traffic data at the rate of one sample every $t_s = 15$ minutes. Note that our algorithm is agnostic of the sampling rate and would work for higher sampling rates as well.

In order to evaluate the VBSF algorithm for both real-time traffic estimation as well as the future traffic prediction problems, an incomplete data set is created by randomly sampling a fraction p of the measurements. In our evaluations we consider three different cases with 75%, 50%, and 25% of missing data. To evaluate the VBSF algorithm for the current (or real-time) traffic estimation task, for a selected time interval, we select previous h = 30 time intervals for the same day. Finally, we use the traffic data for the last 3 months to

Algorithm 2: Robust Variational Bayesian Subspace Filtering

Initialize $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \mathbf{v}$, $sub = 1$, Ω_τ, Ω'_i, $\boldsymbol{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^A, \boldsymbol{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^B, \boldsymbol{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^J \boldsymbol{\Lambda}_1, \mu_1$,
$\hat{\mathbf{Y}} = \boldsymbol{\mu}^A(\boldsymbol{\mu}^{\mathbf{B}})^T$
while $Y_{conv} < 10^{-5}$ **do**
  $\mathbf{Y_{old}} = \hat{\mathbf{Y}}$
  $\boldsymbol{\Gamma} = diag(\boldsymbol{\gamma})$
  if $sub == 1$ **then**
    Update using (20)
    $sub = 2$
    Update using (10a), (11), (15a), (15b) $\forall\ 1 \leq i \leq r$
  else if $sub == 2$ **then**
    Update using (13), (15c), (15d), (10b) $\forall 1 \leq i \leq m$ $sub = 3$
  end
  else
    Update using (36a), (36b) $\forall\ 1 \leq i \leq m,\ \forall\ 1 \leq \tau \leq t$
    $sub = 1$
  end
  $\hat{\mathbf{Y}} = \boldsymbol{\mu}^A(\boldsymbol{\mu}^{\mathbf{B}})^T$
  Update using (34)
  $Y_{conv} = \frac{\|\mathbf{Y} - \mathbf{Y_{old}}\|_F}{\|\mathbf{Y_{old}}\|_F}$
end
return $(\hat{\mathbf{Y}}, \boldsymbol{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^A, \boldsymbol{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^B, \boldsymbol{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^J)$

test the VBSF algorithm, wherein we use the first month data is used to estimate the priors, while the subsequent months data is used to evaluate the accuracy of the estimation and prediction tasks.

A. Performance Index

To measure the effectiveness of our algorithm and for the comparison with other relevant algorithms, we use mean relative error (MRE) as the performance index. For any time instance τ, the MRE denoted by MRE_τ is defined as:

$$\text{MRE}_\tau = \frac{1}{z} \sum_{k=1}^{z} \frac{\| \hat{y}_{\tau,k} - y_{\tau,k} \|_2}{\| y_{\tau,k} \|_2}. \tag{39}$$

where $y_{\tau,k}$ and $\hat{y}_{\tau,k}$ are the ground truth and estimated data for k^{th} day and τ^{th} time instance. Since the value for the known data may be modified post estimation, we compute the MRE over the whole column for a given time instance. MRE is calculated for each day time of the day separately. For calculating the overall accuracy of prediction for a day, we calculate the MRE over z days. The value of z is taken as 50 for weekdays and 10 for the weekends.

B. Real Time Traffic Estimation

We now discuss simulation results for the current traffic estimation based on the current and past missing data using the VBSF algorithm. For a typical day, Fig. 6a shows the heatmap of the actual traffic data. The x axis of each of the heatmap represents time instances while the y axis represents the edges. Each pixel of a heatmap indicates the speed, where higher speed is represented by a lighter colour. Figures 6b, 6e and 6h are heatmaps with missing entries of varying degrees as well as the corresponding completed matrices using VBSF algorithm are shown in Figs. 6c, 6f, and 6i. Since the proposed VBSF is an online method that 'completes' one column at a time given the incomplete data from previous columns, the corresponding heatmaps are also generated in an online fashion. In other words, in spirit of the online methodology, window of $h+1$ incomplete columns are used to complete the last column followed by moving the window by one column. Finally, all the completed columns form a matrix represented in these heatmaps. Unsurprisingly, the heatmaps show that the performance of VBSF increases as the size of missing data decreases.

The MRE values for real time traffic estimation using VBSF for weekends is shown in Fig. 7a and for weekdays in Fig. 7b. It is observed that the prediction error is higher during the peak traffic time (in the evening) vis-a-vis non peak time intervals. This may be due to a greater variance in traffic during the peak-time intervals. However, the difference between the MRE values for 50% and 25% missing data case is only about 0.15 in the worst case. Equivalently, the average error of estimation of speed is only around 2 km/hr during the peak-time when the average speed is 15 km/hr even with 75% missing data. Similarly, for non-peak hours, even though the observed speed are higher (around 30-40 km/hr), the MRE values for $p = 50\%$ and $p = 25\%$ is around 0.1, which in other words indicate an average error of 3-4 km/hr in the estimation of speed.

Finally, we compare our algorithm with other methods that potentially solve the current traffic estimation problem in the missing data scenario. We use low rank tensor completion (LRTC) [7], Grassmannian Rank-One Update Subspace Estimation (GROUSE) [2] algorithm and finally the historic mean for comparison purposes. The historic mean is simply the mean of edge speed values at a given time instance calculated using the data. Table I presents the overall results. Further Figs. 8a and 8b show the comparison of our algorithm for different percentage of missing traffic data. It is observed that for low missing rate of traffic data (25%), the LRTC (low rank tensor completion) [7] and VBSF obtain similar performance. But as the missing data increases, VBSF outperforms the LRTC method. Also, for all the cases, VBSF performs better than GROUSE. This difference in performance can be attributed to the fact that the VBSF framework captures the temporal dependencies as well as the latent factors in the traffic matrix better than other methods. In terms of running time, VBSF is faster than LRTC and is comparable to GROUSE as shown in Table II.

C. Future Traffic Prediction Problem

We also test the VBSF algorithm for speed prediction during the future time intervals assuming randomly sampled data from the current and previous time intervals. We predict traffic

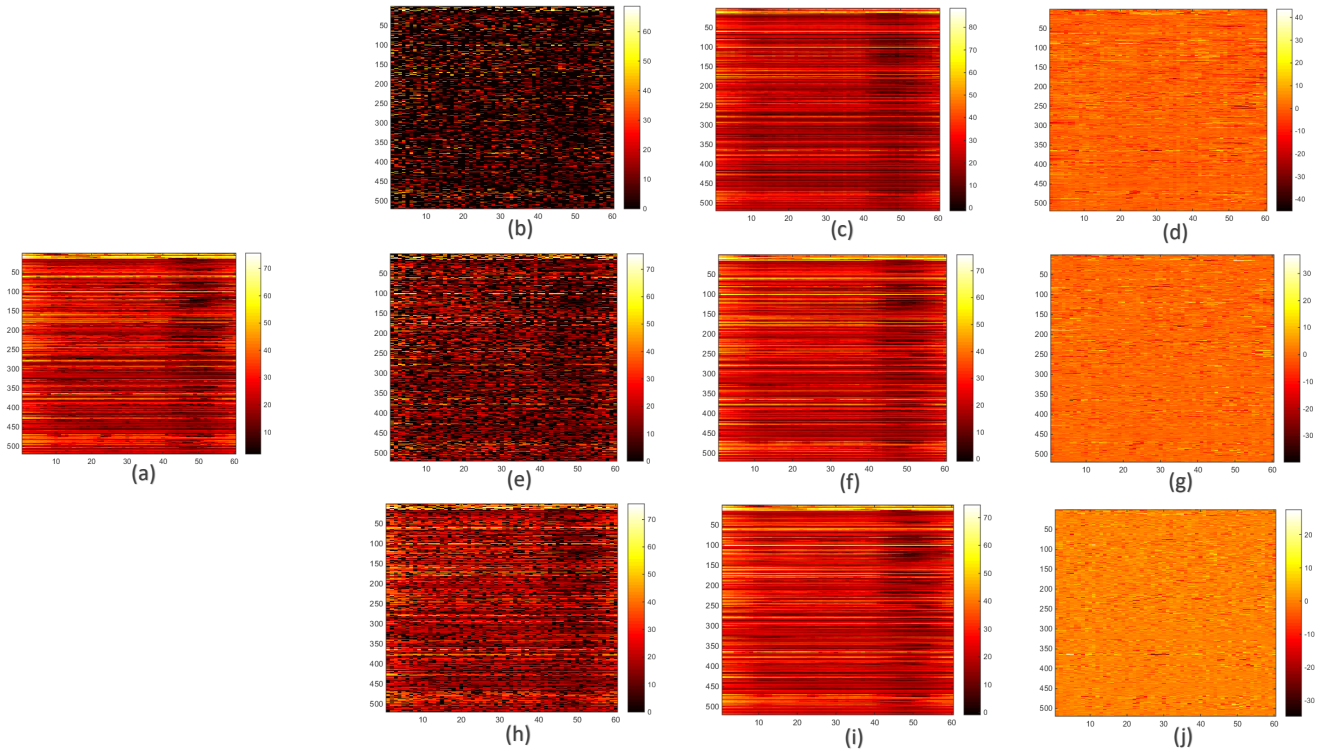

Fig. 6: Estimation of traffic data for different percentage of missing entries
(a) Actual Traffic data , (b) Traffic data with 25% entries, (c) Estimated Traffic with 25% known data, (d) Residual error for estimation with 25% data , (e) Traffic data with 50% entries , (f) Estimated Traffic with 50% known data, (g) Residual error for estimation with 50% data, (h) Traffic data with 75% entries , (i) Estimated Traffic with 75% known data,
(j) Residual error for estimation with 75% data

	$p = 0.25$ MRE	$p = 0.50$ MRE	$p = 0.75$ MRE
VBSF	0.1439	0.11277	0.09336
GROUSE	0.372	0.3446	0.3085
LRTC	0.1921	0.1418	0.09578
Mean	0.2083	0.2083	0.2083

TABLE I: Performance comparison for real time traffic estimation

	$p = 0.25$ time(sec)	$p = 0.50$ time(sec)	$p = 0.75$ time(sec)
VBSF	0.7001	0.8685	0.9675
GROUSE	0.7935	0.85324	0.923960
LRTC	2.92	4.32	6.23

TABLE II: Comparison of running time for different algorithms

data up to 5 sampling intervals, that is 15 to 75 minutes in future. We test our algorithm for 50% and 75% of the missing entries in the traffic data The MRE plots for traffic prediction is shown in Figs. 7c, 7d and 7e. The MRE error difference for 50% and 75% missing data is not significant. Similar to observations from the current traffic estimation simulations, it is seen that the error increases from 5:30 to 8:00 pm. As one would expect, the prediction accuracy decreases as we predict further in future. Interestingly, it is observed that the MRE for real-time traffic estimation with 75% missing entries case and for future prediction with 50% missing entries are comparable as can be seen in Fig. 7f.

The performance of the proposed VBSF algorithm is compared with that of LRTC in Table III. The VBSF performs better than the LRTC as shown in Fig. 8c. While predicting the speed for outlier edges (the edges which significantly deviate from their usual speed) VBSF performs better than LRTC as seen in Fig. 8d.

	$p = 0.50$ $15\,mins$	$p = 0.50$ $30\,mins$
VBSF	0.15362	0.17434
LRTC	0.15843	0.1812
Mean	0.2082	0.2073

TABLE III: Performance comparison for traffic prediction

D. Robust Traffic Estimation

The GPS data that is collected using probe vehicles may be corrupted by noise and may often contain outliers which need to be removed before further processing is performed. To mitigate the performance degradation due to outliers,

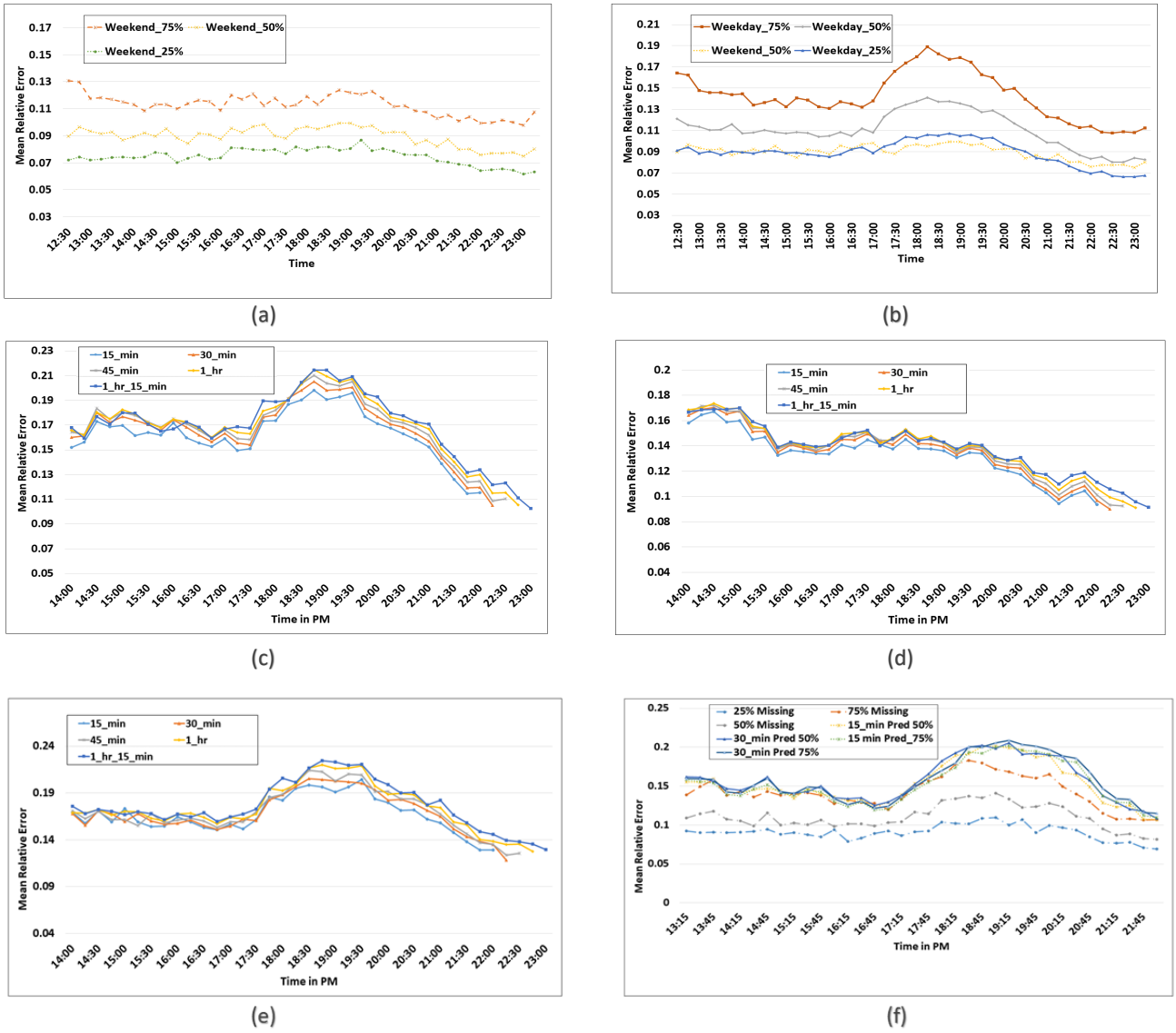

Fig. 7: Real time Traffic Estimation and Prediction for different missing entries
(a) Real time traffic estimation for different missing entries (Weekend), (b) Weekday Prediction 50% missing entries (Weekday), (c) Weekday Prediction 50% missing entries, (d) Weekend Prediction 50% missing entries, (e) Weekday Prediction 75% missing entries, (f) Overall Prediction

we employ the robust variational bayesian subspace filtering (RVBSF) that models the presence of outliers in the data in the sparse outlier matrix $\mathbf{E}$. To test the RVBSF algorithm, on a given day, we randomly sample a certain p_o percentage of the already sampled traffic data $y_{i,\tau}$ and replace these values with $o_{i,\tau}$ as follows:

$$o_{i,\tau} = \max\left(y_{i,\tau-1}, y_{i,\tau+1}\right) + c\,\mu_t. \tag{40}$$

In other words, the outlier is created by adding a large value $c\,\mu_t$ to the maximum of $y_{i,\tau-1}$ and $y_{i,\tau+1}$. Here, mu_t is the mean of observed entries at time t and c is a scaling parameter. The RVBSF algorithm is then applied to solve the real time traffic estimation problem. The detected artificial outliers are those points residing in the matrix $\mathbf{E}$.

The accuracy of outlier detection depends on the outlier value as shown in Fig. 9d. The value of c for simulations is chosen from the set $[0.75, 1, 1.25, 1.5, 1.75]$. We compare the robust VBSF (termed as RVBSF) with VBSF for two scenarios. First, when no outliers are added (VBSF), second, when outliers are present in the data but only VBSF was used (VBSF_with_outliers). Table IV summarises the overall performance of the RVBSF algorithm. Understandably, RVBSF improves over VBSF when outliers are present, but is still worse than the MRE of VBSF for the case when no outliers were present. For 25% missing entries, $p_o = 5\%$ and $c = 0.75$, the plots in Fig. 9a illustrate the performance of the RVBSF

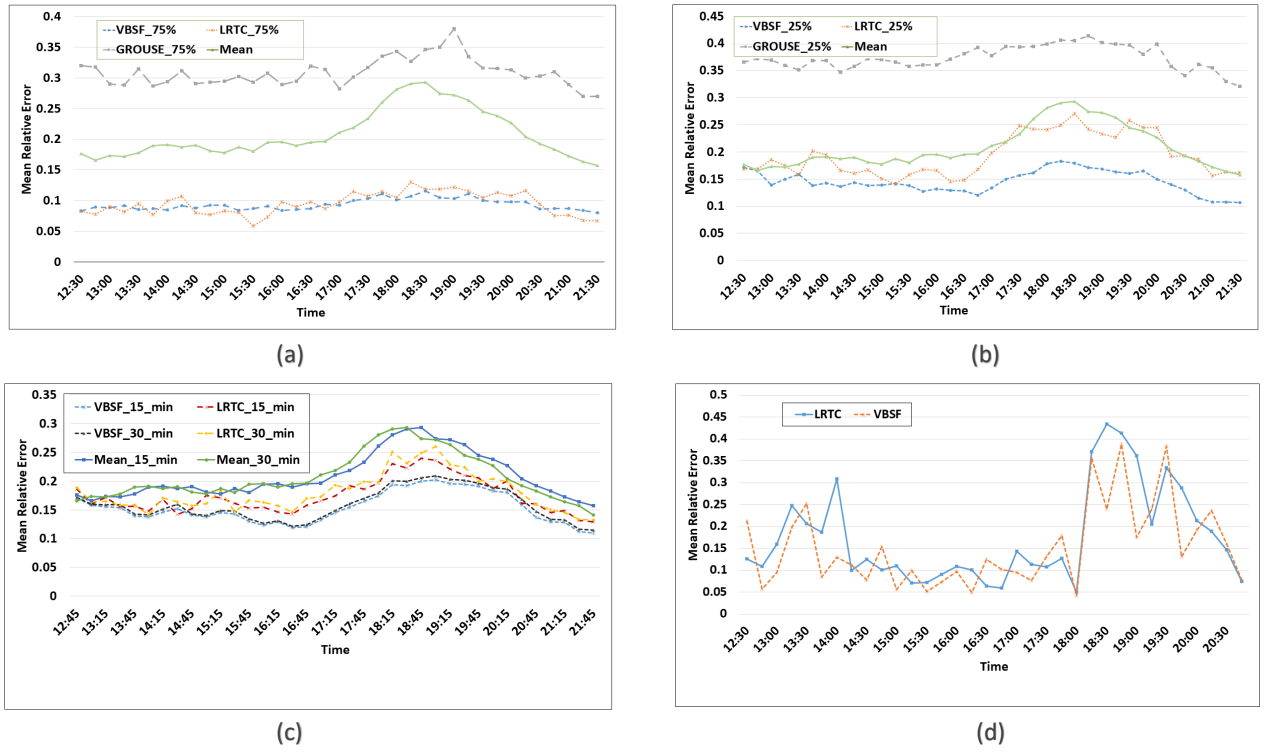

Fig. 8: Comparison between VBSF and Low rank Tensor Completion (LRTC) and Matrix Completion Algorithm (GROUSE) (a) Real Time Traffic Estimation for 25% percentage of Missing Data, (b) Real time traffic estimation for 75% percentage of missing data, (c) Traffic prediction for 50% of missing data, (d) Traffic prediction for outliers

algorithm. Similarly for 75% of missing entries, $p_o = 2\%$ the results are shown in Fig. 9b. When $p_o = 5\%$ and $c = 0.75$, we observe that RVBSF detects outliers reasonably well vis-a-vis VBSF_with_outliers. Similar observation holds when outlier values increase as shown in Fig. 9c and Fig. 9d.

	$c = 0.75$ $p_o = 5\%$	$c = 0.75$ $p_o = 2\%$	$c = 1.5$ $p_o = 2\%$
VBSF	0.09462	0.09457	0.09434
VBSF_outlier	0.13406	0.11643	0.15318
RVBSF	0.11741	0.1127	0.10912

TABLE IV: RVBSF: overall performance

The performance of the proposed RVBSF algorithm is compared with that of OP-RPCA[**?**] GRASTA[16] and ROSETA[18] in Table V. Our algorithm performs better than GRASTA[16].

A possible limitation of the suggested robust traffic estimation framework is following. While there may be outliers present due to an erroneous speed estimation, there might be cases when the so called outlier value may actually be a real value. The current method may not be able to distinguish between such cases. Hence, a sudden drop in speed along an edge may be treated as an outlier and its possible impact on the traffic of nearby edges be be ignored by the model.

	$c = 0.75$ $p_o = 5\%$	$c = 0.75$ $p_o = 2\%$	$c = 1.5$ $p_o = 2\%$
OP-RPCA	0.2594	0.2298	0.2165
ROSETA	0.1859	0.1819	0.1723
GRASTA	0.1493	0.1507	0.1492
RVBSF	0.11741	0.1127	0.10912

TABLE V: Performance Comparison for Robust Traffic Estimation

E. Expected Time of Arrival estimation

We test the efficacy of the traffic prediction using VBSF for estimating the ETA of an imaginary vehicle plying in the network. To this end, we create three routes of various lengths. The routes consist of a sequences of connected edges and the three selected routes are shown in Fig. 10. The network is randomly sampled with 25%, 50% and 75% of values for a window size of 30 columns. We then use the VBSF algorithm to perform speed prediction in the edges of all three routes for a future duration of 15, 30 and 45 minutes, followed by the computation of the time taken to traverse the routes. The results are tabulated in Table VI. The columns of Table VI indicate distance, average time required to travel each route and the average error in ETA upto 3 time intervals ahead for various missing data percentage values. The average error in ETA is calculated by averaging it over 6 days with different sampling matrices and for 30 time instances between 13:00 pm and 20:15 pm. Negative entries in the table indicate that

the estimated ETA is less than the actual ETA for a route. Similarly, positive entries depicts that the actual ETA is less than the predicted ETA. We see that ETA for route 1 and 3 turn out be sufficiently accurate while route 2 showed more error than the other two routes.

F. Electricity Load Prediction

We use the online electricity dataset available[48]. The data contains the hourly power consumption of 370 consumers, sampled every 15 min in kW. The data is recorded from Jan. 1, 2012 to Jan. 1, 2015. We run our algorithm online on this data. We sample p percentage of data from the dataset and run our algorithm to predict the real time and 1 step ahead electricity load prediction. The result for real time prediction is shown in table VII

Further, we predict the 1 step electricity load prediction and compared it with the algorithm presented in paper[49]. From table VIII, we can conclude that our algorithm performs better than the algorithm in paper[49] for electricity load dataset.

V. Conclusion

Real time traffic estimation and future traffic prediction in road networks is a problem with an inherently online flavour. A stream of incomplete traffic data arrives sequentially and the transit agencies need to estimate the traffic density/speed in the remaining edges along with an accurate prediction of the future traffic density. The VBSF algorithm presented in the paper models the traffic matrix as a low rank subspace whose temporal evolution is characterised by a state space model. Simulation experiments quantify that the suggested model can be deployed to estimate the missing traffic data with a reasonable accuracy even with a fraction of random traffic measurements in the network. Moreover, one can also predict the future traffic which in return can be used to increase the reliability of the public transport even in places with multiple modes of transport.

References

[1] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[2] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE, 2010, pp. 704–711.

[3] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

[4] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas, "Online sparse and low-rank subspace learning from incomplete data: A bayesian view," *Signal Processing*, vol. 137, pp. 199–212, 2017.

[5] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[6] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3419–3430, 2011.

[7] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.

[8] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passingpart i: Derivation," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5839–5853, 2014.

[9] ——, "Bilinear generalized approximate message passingpart ii: Applications," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5854–5867, 2014.

[10] B. Xin, Y. Wang, W. Gao, and D. Wipf, "Exploring algorithmic limits of matrix rank minimization under affine constraints," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 4960–4974, 2016.

[11] L. Yang, J. Fang, H. Duan, H. Li, and B. Zeng, "Fast low-rank bayesian matrix completion with hierarchical gaussian prior models," *IEEE Transactions on Signal Processing*, 2018.

[12] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1816–1825, 2016.

[13] J. Luttinen, "Fast variational bayesian linear state-space model," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 305–320.

[14] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational bayesian matrix factorization for bounded support data," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 4, pp. 876–889, 2015.

[15] L. Yang, J. Fang, H. Duan, H. Li, and B. Zeng, "Fast low-rank bayesian matrix completion with hierarchical gaussian prior models," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2804–2817, 2018.

[16] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," *arXiv preprint arXiv:1109.3827*, 2011.

[17] H. Xu, C. Caramanis, and S. Sanghavi, "Robust pca via outlier pursuit," in *Advances in Neural Information Processing Systems*, 2010, pp. 2496–2504.

[18] H. Mansour and X. Jiang, "A robust online subspace estimation and tracking algorithm," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4065–4069.

[19] M. Papageorgiou, C. Diakaki, V. Dinopoulou, A. Kotsialos, and Y. Wang, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043–2067, 2003.

[20] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, vol. 114, no. 3, pp. 462–467, 2017.

[21] StackExchange, "How accurate is google maps for travel times," 2014. [Online]. Available: \url{https://travel.stackexchange.com/questions/39354/how-accurate-is-google-maps-for-travel-times}

[22] D. Mohan, "Moving around in indian cities," *Economic and Political Weekly*, vol. 48, no. 48, 2013.

[23] R. Goel and G. Tiwari, "Access–egress and other travel characteristics of metro users in delhi and its satellite cities," *IATSS Research*, vol. 39, no. 2, pp. 164–172, 2016.

[24] L. Qu, L. Li, Y. Zhang, and J. Hu, "Ppca-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Transactions on intelligent transportation systems*, vol. 10, no. 3, pp. 512–522, 2009.

[25] L. Qu, Y. Zhang, J. Hu, L. Jia, and L. Li, "A bpca based missing value imputing method for traffic flow volume data," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 985–990.

[26] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 8, pp. 2123–2133, 2016.

[27] J. Guo, W. Huang, and B. M. Williams, "Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[28] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.

[29] L. Li, S. He, J. Zhang, and B. Ran, "Short-term highway traffic flow prediction based on a hybrid strategy considering temporal–spatial information," *Journal of Advanced Transportation*, vol. 50, no. 8, pp. 2029–2040, 2016.

[30] S. Dunne and B. Ghosh, "Regime-based short-term multivariate traffic condition forecasting algorithm," *Journal of Transportation Engineering*, vol. 138, no. 4, pp. 455–466, 2011.

[31] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation*

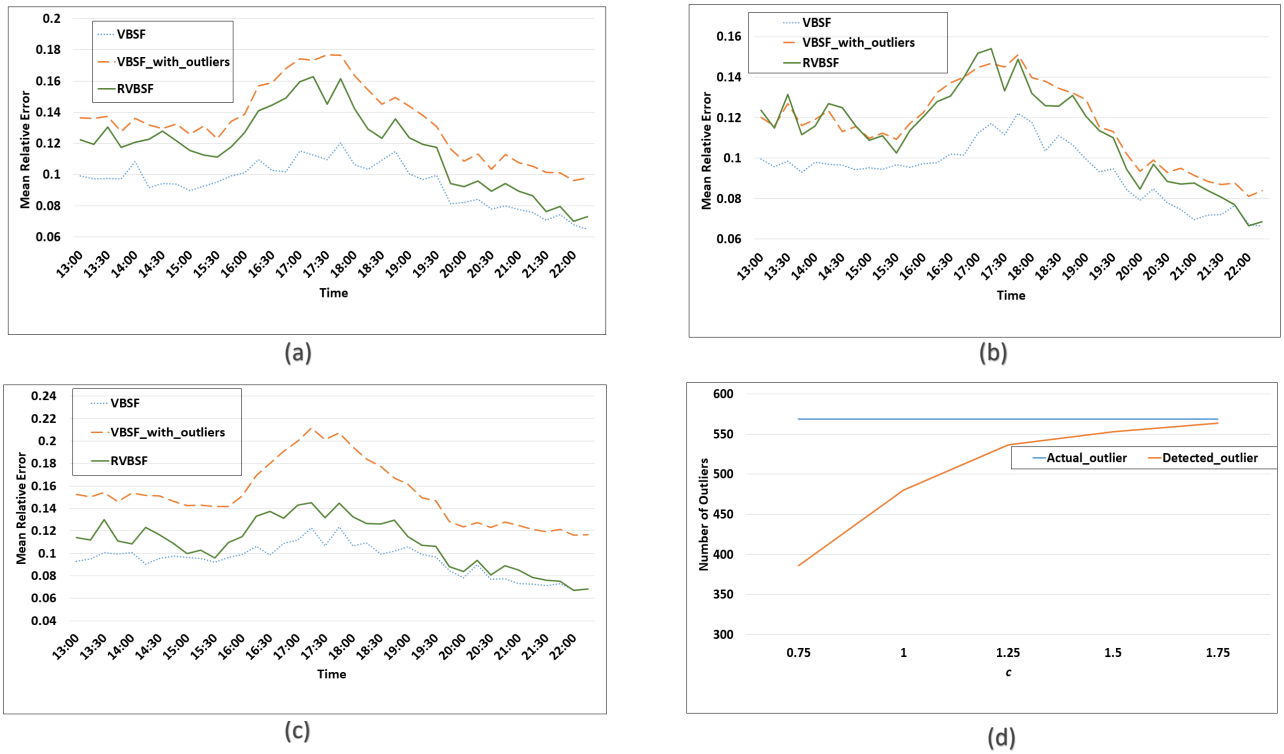

Fig. 9: Robust Bayesian subspace filtering for traffic data
(a) Comparison for VBSF and RVBSF with 5% outliers and c = 0.75, (b) Comparison for VBSF and RVBSF with 2% outliers and c = 0.75 (c) Comparison of VBSF and RVBSF for c = 1.25, (d) Number of outliers detected for different outlier values

	Distance km	Avg Time $(mins)$	15 $mins$ 75%	15 $mins$ 50%	15 $mins$ 25%	30 $mins$ 75%	30 $mins$ 50%	30 $mins$ 25%	45 $mins$ 75%	45 $mins$ 50%	45 $mins$ 25%
Route1	1.733	6.505	-0.816	-0.882	-0.940	-0.958	0.940	0.9621	0.990	0.9813	0.9808
Route2	4.473	15.735	-3.16	-3.4018	-3.48	-3.45	-3.626	-3.895	-3.745	-4.090	-3.952
Route3	2.370	15.5	0.0598	-0.0934	-0.089	-0.033	0.20	0.33	0.135	0.424	0.58

TABLE VI: Average error in ETA for the three routes in minutes

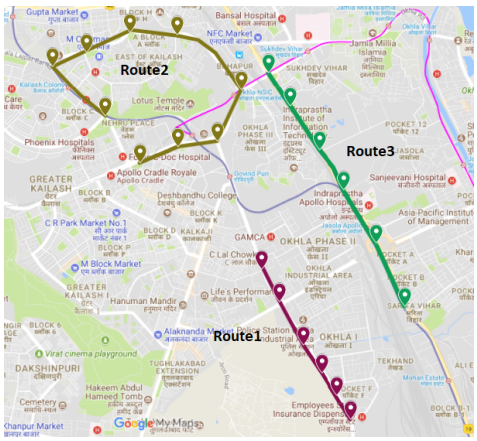

Fig. 10: Different routes for ETA

	$p = 0.25\%$	$p = 0.5\%$	$p = 0.75\%$
MRE	0.1789	0.101	0.0987
MAE(kW)	96.83	66.67	53.95

TABLE VII: Electricity real time load prediction

	$p = 0.25\%$	$p = 0.5\%$	$p = 0.75\%$
MAE(kW)	98.38	75.995	65.70

TABLE VIII: Electricity 1 step ahead load prediction

Research Part C: Emerging Technologies, vol. 11, no. 2, pp. 121–135, 2003.

[32] L. Li, Y. Li, and Z. Li, "Efficient missing data imputing for traffic flow by considering temporal and spatial dependence," *Transportation research part C: emerging technologies*, vol. 34, pp. 108–120, 2013.

[33] D. Ni and J. Leonard II, "Markov chain monte carlo multiple imputation using bayesian networks for incomplete intelligent transportation systems data," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1935, pp. 57–67, 2005.

[34] C. Zhang, S. Sun, and G. Yu, "A bayesian network approach to time series forecasting of short-term traffic flows," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004, pp. 216–221.

[35] B. Ghosh, B. Basu, and M. OMahony, "Bayesian time-series model for short-term traffic flow forecasting," *Journal of transportation engineer-*

ing, vol. 133, no. 3, pp. 180–189, 2007.

[36] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for dna microarrays," *Bioinformatics*, vol. 17, no. 6, pp. 520–525, 2001.

[37] G. Chang, Y. Zhang, and D. Yao, "Missing data imputation for traffic flow based on improved local least squares," *Tsinghua Science and Technology*, vol. 17, no. 3, pp. 304–309, 2012.

[38] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg–marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.

[39] K. Y. Chan and T. S. Dillon, "On-road sensor configuration design for traffic flow prediction using fuzzy neural networks and taguchi method," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 1, pp. 50–59, 2013.

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

[41] Y. Duan, Y. Lv, W. Kang, and Y. Zhao, "A deep learning based approach for traffic data imputation," in *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 912–917.

[42] X. Liu, S. I. Chien, and M. Chen, "An adaptive model for highway travel time prediction," *Journal of Advanced Transportation*, vol. 48, no. 6, pp. 642–654, 2014.

[43] S. D. Babacan, M. Luessi, R. Molina, and A. K. Katsaggelos, "Sparse bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

[44] M. T. Asif, N. Mitrovic, L. Garg, J. Dauwels, and P. Jaillet, "Low-dimensional models for missing data imputation in road networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3527–3531.

[45] C. M. Bishop, "Pattern recognition and machine learning (information science and statistics) springer-verlag new york," *Inc. Secaucus, NJ, USA*, 2006.

[46] M. J. Beal *et al.*, *Variational algorithms for approximate Bayesian inference*. university of London London, 2003.

[47] J. Luttinen, "Fast variational bayesian linear state-space model," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2013, pp. 305–320.

[48] UCI, "Electricity load diagrams 20112014," 2014. [Online]. Available: \url{https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014}

[49] S. Gultekin and J. Paisley, "Online forecasting matrix factorization," *CoRR*, vol. abs/1712.08734, 2017. [Online]. Available: http://arxiv.org/abs/1712.08734