Traffic Estimation and Prediction via Online Variational Bayesian Subspace Filtering

Charul Paliwal[1], *Student Member*, *IEEE*, Uttkarsha Bhatt[2] , Pravesh Biyani[1], *Member*, *IEEE* and Ketan Rajawat[3], *Member*, *IEEE*

***Abstract*—With the increased proliferation of smart devices, the transit passengers of today expect a higher quality of service in the form of real-time traffic updates, accurate expected time-of-arrival (ETA) predictions. Providing these services requires 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 speed data demonstrate 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

Traffic estimation and prediction are the central components of any urban traffic congestion management system [1]. 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 [2]. 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. Transportation providers like Google and Microsoft lack the resources to collect accurate city-wide traffic densities and thus end up working with incomplete and noisy data sets.

[1]The authors are with the Department of Electronics and Communication Engineering, Indraprastha Institute of Information Technology, Delhi, India.

[3]K. Rajawat is with the Department of Electrical Engineering, Indian Institute of Technology, Kanpur, UP, India.

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, the 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 [3]. 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 multi-modal nature of traffic [4], [5]. 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 [6]. A class of pertinent approaches has sought to visualize the traffic data as an incomplete matrix or tensor, and exploited this correlation to fill-in the missing entries [6]–[9]. 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 [10]. 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 speed 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 [11] and LRTC [12]. 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:

1) A variational Bayesian subspace filtering (VBSF) approach is proposed, where the traffic matrix is modeled as comprising of an underlying 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 speed 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.

A. Related Work

Road traffic density modeling, estimation, and prediction are of research interest to the transportation community for many years. Authors in [13] and [14] use autoregressive 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 [15], [16]. Further, methods that exploit both temporal and spatial correlations were designed in [7], [17], [18]. Bayesian network based forecasting of traffic is proposed in [19], [20].

In the category of non-linear models, k-nearest-neighbour [21] and local least square (LLS) [22] have also been suggested to estimate the missing traffic data. Recently neural networks based approaches are also proposed in [23]–[26].

Gaussian process regression is used in literature for traffic prediction [27], [28]. However, these methods do not deal with the multivariate time series and random missing entries. Univariate time series models are difficult to scale for more than 500 time series used in the paper. Closer to our work, authors in [10] and [29] adopt an online approach to traffic-flow prediction. Specifically, authors in [10] use an adaptive Kalman filter by constructing state space models and perform short term traffic forecasting. However, both works [10] and [29] 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 [6], where the authors employ various matrix completion techniques to fill the missing entries in the traffic matrix, while [12], [9] use tensor completion algorithm to fill the missing data. Among many techniques, the authors in [6] 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 [6], [30]–[38]. One of the first works considered the measured matrix to be expressible as a product of low-rank matrices, associated with appropriate ARD priors [30] while faster algorithms for similar settings were proposed in [31], [32], [38]. More recently, other approaches towards modeling the measured matrices have also been proposed [33], [38]. Moreover, variational Bayesian approaches have also been applied to road traffic estimation; see e.g., [6]. 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 [30], [36]–[38]. In contrast to these, the state-space modeling in our work is inspired from [35], 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 [35] to incorporate low-rank structure and outliers.

Several non-Bayesian algorithms have been proposed to address the online subspace estimation problem from incomplete observations [11], [39]–[41]. GROUSE [11] is one of the early approaches that uses an update on the Grassmannian manifold to estimate the subspace. The robust variant of GROUSE, namely GRASTA , handles outliers by incorporating the l_1 norm cost function [39]. OP-RPCA [40] is a robust subspace estimation technique that uses alternating minimization to compute the outliers and the underlying subspace. A number of online subspace tracking algorithms, such as ROSETA [41], have since been proposed. The proposed approach is compared with some of these algorithms in Sec. IV.

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 the 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 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, $\text{Ga}(x, a, b)$ 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 $\mathbf{D}$, the posterior mean is given by $\hat{\mathbf{x}} := \mathbb{E}[\mathbf{x} \mid \mathbf{D}]$.

II. Variational Bayesian Subspace Filtering

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 $1 \leq \tau \leq t$, 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 $1 \leq \tau \leq t$, and
- *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.

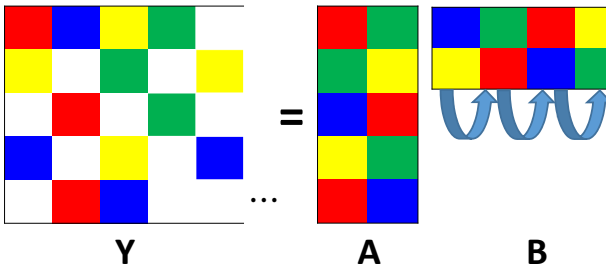


Fig. 1: Online Variational Bayesian Filtering

A. Hierarchical Bayesian Model

We begin with detailing a generative model for the matrix $\mathbf{Y}$. The proposed model will not only capture the rank deficient nature of $\mathbf{Y}$ [42] but also the temporal correlation between successive columns of $\mathbf{Y}$ [43]. Recall that the standard low-rank parametrization of the full matrix $\mathbf{Y}$ takes the form $\mathbf{Y} = \mathbf{AB}$ where $\mathbf{A} \in \mathbb{R}^{m \times r}$ and $\mathbf{B} \in \mathbb{R}^{r \times t}$ where r is the rank of matrix $\mathbf{Y}$. 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 distribution with unknown hyper-parameters, while various components or parameters are assigned different prior distributions. The Bayesian framework allows the use of ARD, wherein associating appropriate priors to the model parameters leads to pruning of the redundant features [42].

Specifically, for matrix completion to capture the low rankness in the data, the entries of $\mathbf{Y}$ are generated as

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

for all $\tau \geq 1$, 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 hyper-parameter r is not critical to the imputation or prediction accuracy, but is only required to be 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:

$$\mathbf{b}_\tau = \mathbf{J}\mathbf{b}_{\tau-1} + noise \tag{2}$$

In Bayesian setting, we model the autoregressive model as

$$p(\mathbf{b}_\tau \mid \mathbf{J}, \mathbf{b}_{\tau-1}) = \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \qquad 2 \leq \tau \leq t \tag{3}$$

for $\tau \geq 2$, where $\mathbf{J} \in \mathbb{R}^{r \times r}$ is again a problem parameter. Here, $\mathbf{J}$ captures the temporal structure of the underlying subspace, and is learned from the data itself. The scaling ambiguity present in matrix factorization allows the transition matrix $\mathbf{J}$ to capture both slow and fast variations in $\mathbf{b}_\tau$ without the need to explicitly model the state noise variance. It follows from (3) that the conditional pdf of $\mathbf{b}_\tau$ given $\mathbf{J}$ is given by

$$p(\mathbf{B} \mid \mathbf{J}) = \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^{t} \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r). \tag{4}$$

Observe that the model complexity depends on the rank r, which is also the number of columns in $\mathbf{A}$ and $\mathbf{J}$. In order to ensure the value of r is learned in a data-driven fashion, the columns of $\mathbf{A}$ and $\mathbf{J}$ are assigned multivariate Gaussian priors with column-specific precisions, i.e.,

$$p(\mathbf{A} \mid \boldsymbol{\gamma}) = \prod_{i=1}^{r} \mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1} \mathbf{I}_m) \tag{5}$$

$$p(\mathbf{J} \mid \boldsymbol{\upsilon}) = \prod_{i=1}^{r} \mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1} \mathbf{I}_r) \tag{6}$$

where the precisions $\boldsymbol{\gamma}$ and $\boldsymbol{\upsilon}$ are problem parameters. It can be seen that if any of γ_i or υ_i are large, the corresponding columns will be close to zero and consequently irrelevant. Indeed, the priors in (5)-(6) aid in automatic relevance determination since the subsequent optimization process may drive some of the precisions to infinity, yielding a low-rank factorization.

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

$$p(\beta) = \frac{1}{\beta}, \qquad p(\gamma_i) = \frac{1}{\gamma_i}, \qquad p(\upsilon_i) = \frac{1}{\upsilon_i} \tag{7}$$

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

$$p(\mathbf{y}_\Omega, \mathcal{H}) = p(\mathbf{y}_\Omega | \mathbf{A}, \mathbf{B}, \beta) p(\mathbf{A} | \boldsymbol{\gamma}) p(\mathbf{B} | \mathbf{J}) p(\mathbf{J} | \boldsymbol{\upsilon}) p(\beta) p(\boldsymbol{\upsilon}) p(\boldsymbol{\gamma})$$

The full hierarchical Bayesian model adopted here is summarized in Fig. 2(a).

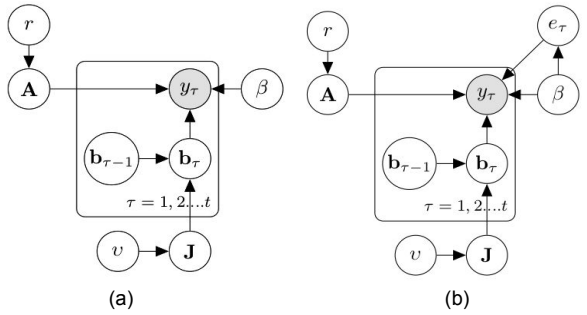


Fig. 2: (a) Hierarchical Bayesian Model for Matrix Completion (b) Robust Hierarchical Bayesian Model for Matrix Completion

B. Variational Bayesian Inference

Having specified the generative model for the data, the goal is to determine the posterior distribution $p(\mathcal{H}|\mathbf{y}_\Omega)$ given the likelihood probability of $\mathbf{y}_\Omega$ and prior distribution of $\mathcal{H}$. However, the posterior distribution $p(\mathcal{H}|\mathbf{y}_\Omega)$ is intractable as $p(\mathbf{y}_\Omega)$ needs to be marginalized over all parameters ($\mathcal{H}$). Therefore, we utilize the mean-field approximation, wherein the posterior distribution factorizes as:

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

In other words, the posterior is now restricted to a family of distributions that adhere to (8). The factors $q_{\mathbf{A}}$, $q_{\mathbf{B}}$, $q_{\mathbf{J}}$, q_υ, q_β, and q_γ can be determined by minimizing the Kullback–Leibler divergence of $p(\mathcal{H}|\mathbf{y}_\Omega)$ from $q(\mathcal{H})$, usually via an alternating minimization approach [44]. 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 [35]:

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

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^β, b^β, a_i^γ, b_i^γ, a_i^υ, $b_i^\upsilon \in \mathbb{R}_{++}$.

We use EM Algorithm to derive the posterior distribution in (9). 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 $\mathcal{H}_p := \{\boldsymbol{\upsilon}, \boldsymbol{\gamma}, \beta\}$.

C. Update Equations

Each iteration of EM Algorithm simply involves updating the variables $\{\boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{B}}, \{\boldsymbol{\mu}_i^{\mathbf{A}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{A}}\}, \{\boldsymbol{\mu}_i^{\mathbf{J}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{J}}\}, \hat{\upsilon}, \hat{\gamma}, \hat{\beta}\}$ in a cyclic manner. 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

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

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 as

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

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

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

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

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

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

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

$$\tag{18}$$

where $\Omega_i' := \{\tau \mid i \in \Omega_\tau\}$, $\hat{\mathbf{J}} := \mathbb{E}[\mathbf{J} \mid \mathbf{y}_\Omega]$ as the matrix whose i-row is given by $(\boldsymbol{\mu}_i^{\mathbf{J}})^T$, and $\boldsymbol{\Sigma}^{\mathbf{J}} := \sum_{i=1}^r \boldsymbol{\Sigma}_i^{\mathbf{J}}$ It is remarked that although the $rt \times rt$ matrix $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1}$ is

block-tridiagonal, the matrix $\mathbf{\Xi}^{\mathbf{B}}$ is dense, and direct inversion would be prohibitively costly. Moreover, the classical Rauch-Tung-Striebel (RTS) smoother cannot be directly applied since evaluating the conditional expectations under $q(\mathbf{B})$ is difficult and not amenable to the Matrix Inversion Lemma [45]. Interestingly, observe that the updates in (14) and (15) depend only on diagonal and super-diagonal blocks of $\mathbf{\Xi}^{\mathbf{B}}$, namely $\mathbf{\Xi}^{\mathbf{B}}_{\tau,\tau}$ and $\mathbf{\Xi}^{\mathbf{B}}_{\tau,\tau-1}$, respectively. The next subsection details a low-complexity algorithm for carrying out the updates for these blocks as well as for $\boldsymbol{\mu}^{\mathbf{B}}$.

D. Low-complexity updates via LDL-decomposition

Thanks to the block-tridiagonal structure of $[\mathbf{\Xi}^{\mathbf{B}}]^{-1}$, it is possible to use the LDL decomposition to carry out the updates in an efficient manner. Decomposing $[\mathbf{\Xi}^{\mathbf{B}}]^{-1} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, the key idea is that left multiplication with $\mathbf{\Xi}^{\mathbf{B}}$ is equivalent to left multiplication with $\mathbf{L}^{-T}\mathbf{D}^{-1}\mathbf{L}^{-1}$. Towards this end, we utilize the algorithm from [35], 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 $2 \leq \tau \leq t$,

$$\mathbf{\Psi}_\tau := \hat{\beta} \sum_{i \in \Omega_\tau} \mathbf{\Sigma}^{\mathbf{A}}_{(i)} + \mathbf{I}_r + \mathbf{1}_{\tau \neq t} \sum_{i=1}^{r} \mathbf{\Sigma}^{\mathbf{J}}_i \tag{19}$$

$$\mathbf{v}_\tau := \hat{\beta} \sum_{i \in \Omega_\tau} y_{i\tau} \boldsymbol{\mu}^{\mathbf{A}}_i. \tag{20}$$

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

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

$$\breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau+1} = -\hat{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau} \hat{\mathbf{J}} \tag{21a}$$

$$\breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau+1,\tau+1} = (\mathbf{\Psi}_{\tau+1} - (\breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau+1})^T \mathbf{\Psi}^{\mathbf{B}}_{\tau,\tau+1})^{-1} \tag{21b}$$

$$\breve{\boldsymbol{\mu}}^{\mathbf{B}}_{\tau+1} = \breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau+1,\tau+1} (\mathbf{v}_{\tau+1} - (\breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau+1})^T \breve{\boldsymbol{\mu}}^{\mathbf{B}}_\tau) \tag{21c}$$

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

$$\mathbf{\Xi}^{\mathbf{B}}_{\tau,\tau+1} = -\breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau+1} \mathbf{\Xi}^{\mathbf{B}}_{\tau+1,\tau+1} \tag{21d}$$

$$\mathbf{\Xi}^{\mathbf{B}}_{\tau,\tau} = \breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau} - \hat{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau+1} (\mathbf{\Xi}^{\mathbf{B}}_{\tau,\tau+1})^T \tag{21e}$$

$$\boldsymbol{\mu}^{\mathbf{B}}_\tau = \breve{\boldsymbol{\mu}}^{\mathbf{B}}_\tau - \breve{\mathbf{\Xi}}^{\mathbf{B}}_{\tau,\tau+1} \boldsymbol{\mu}^{\mathbf{B}}_{\tau+1} \tag{21f}$$

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

Note that while $\mathbf{\Xi}^{\mathbf{B}}_{i,j} \neq 0$ 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

$$y_{i\tau} = (\boldsymbol{\mu}^{\mathbf{B}}_\tau)^T \boldsymbol{\mu}^{\mathbf{A}}_i. \tag{22}$$

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

$$y_{i\tau} = (\hat{\mathbf{J}}^{\tau-t} \boldsymbol{\mu}^{\mathbf{B}}_t)^T \boldsymbol{\mu}^{\mathbf{A}}_i. \tag{23}$$

It can be seen that as compared to the updates in (16)-(18) that incur a complexity of $\mathcal{O}(t^3)$, the complexity incurred due to (21) is only $\mathcal{O}(t)$. Overall, the different parameters are updated cyclically until convergence for each $t = 1, 2, \ldots$.

1) Remarks on the Convergence of VBSF: The VB framework used in the present work is a special case of a more general mean field approximation approach. The convergence of the VB algorithm is well-known; see e.g. [46], [47]. Intuitively, the variational approximation renders the evidence lower bound convex in individual factors, and thus amenable to coordinate ascent iterations. Since the lower bound is also differentiable with respect to each factor, the coordinate ascent iterations converge to a stationary point; see [48] for a more general result. However, convergence to the global optimum is not guaranteed.

Algorithm 1: Variational Bayesian Subspace Filtering

Initialize $\boldsymbol{\gamma}, \boldsymbol{\beta}, \mathbf{v}$, $sub = 1$, Ω_τ, Ω'_i, $\mathbf{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^{\mathbf{A}}, \mathbf{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^{\mathbf{B}}, \mathbf{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^{\mathbf{J}}, \mathbf{\Lambda}_1, \mu_1$,
$\hat{\mathbf{Y}} = \boldsymbol{\mu}^A (\boldsymbol{\mu}^{\mathbf{B}})^T$
while $Y_{conv} < 10^{-5}$ **do**
  $\mathbf{Y}_{\mathbf{old}} = \hat{\mathbf{Y}}$
  $\mathbf{\Gamma} = diag(\boldsymbol{\gamma})$
  if $sub == 1$ **then**
    Update using (21)
    $sub = 2$
    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)
  $Y_{conv} = \frac{\|\mathbf{Y} - \mathbf{Y}_{\mathbf{old}}\|_F}{\|\mathbf{Y}_{\mathbf{old}}\|_F}$
end
return $(\hat{\mathbf{Y}}, \mathbf{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^{\mathbf{A}}, \mathbf{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^{\mathbf{B}}, \mathbf{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^{\mathbf{J}})$

E. 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 data at time t, estimates may only be required for entries at time $t - \Delta$ 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. For instance, within the context of traffic density prediction, the inferred approximate distribution for a day may be used as a prior 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 distributions obtained from the previous month's data.

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:

$$p(y_{i\tau} \mid \mathbf{a}_{i\cdot}, \mathbf{b}_\tau, e_{i\tau}, \beta) = \mathcal{N}(y_{i\tau} \mid \mathbf{b}_\tau^T \mathbf{a}_{i\cdot} + e_{i\tau}, \beta^{-1}) \quad i \in \Omega_\tau \tag{24}$$

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.,

$$p(\mathbf{B} \mid \mathbf{J}) = \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^t \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \tag{25a}$$

$$p(\mathbf{A} \mid \boldsymbol{\gamma}) = \prod_{i=1}^r \mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1}\mathbf{I}) \tag{25b}$$

$$p(\mathbf{J} \mid \boldsymbol{\upsilon}) = \prod_{i=1}^r \mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1}\mathbf{I}) \tag{25c}$$

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.,

$$p(e_{i\tau}) = \mathcal{N}(e_{i\tau} \mid 0, \alpha_{i\tau}^{-1}) \qquad i \in \Omega_\tau \tag{26}$$

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

$$p(\beta) = \frac{1}{\beta}, \quad p(\gamma_i) = \frac{1}{\gamma_i}, \quad p(\upsilon_i) = \frac{1}{\upsilon_i}, \quad p(\alpha_{i\tau}) = \frac{1}{\alpha_{i\tau}}. \tag{27}$$

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

$$\begin{aligned}
&p(\mathbf{y}_\Omega, \mathcal{H}) \\
&= p(\mathbf{y}_\Omega|\mathbf{A}, \mathbf{B}, \beta)p(\mathbf{A}|\boldsymbol{\gamma})p(\mathbf{B}|\mathbf{J})p(\mathbf{J}|\boldsymbol{\upsilon})p(\mathbf{e}|\boldsymbol{\alpha})p(\beta)p(\boldsymbol{\upsilon})p(\boldsymbol{\gamma}) \\
&= \prod_{\tau=1}^t \prod_{i\in\Omega_\tau} \mathcal{N}(y_{i\tau} \mid \mathbf{b}_\tau^T \mathbf{a}_{i\cdot}, \beta^{-1}) \mathcal{N}(e_{i\tau} \mid 0, \alpha_{i\tau}^{-1}) \frac{1}{\alpha_{i\tau}} \\
&\qquad \times \prod_{i=1}^r \left[\mathcal{N}(\mathbf{a}_i \mid 0, \gamma_i^{-1}\mathbf{I}) \mathcal{N}(\mathbf{j}_i \mid 0, \upsilon_i^{-1}\mathbf{I})\right] \\
&\qquad \times \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^t \mathcal{N}(\mathbf{b}_\tau \mid \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}) \frac{1}{\beta} \prod_{i=1}^r \frac{1}{\gamma_i \upsilon_i}.
\end{aligned} \tag{28}$$

The full hierarchical Bayesian model adopted here is summarized in figure 2(b).

A. Variational Bayesian Inference

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

$$\begin{aligned}
p(\mathcal{H} \mid \mathbf{y}_\Omega) &\approx q(\mathcal{H}) \\
&= q_{\mathbf{A}}(\mathbf{A}) q_{\mathbf{B}}(\mathbf{B}) q_{\mathbf{J}}(\mathbf{J}) q_{\mathbf{e}}(\mathbf{e}) q_{\boldsymbol{\upsilon}}(\boldsymbol{\upsilon}) q_\beta(\boldsymbol{\beta}) q_{\boldsymbol{\gamma}}(\boldsymbol{\gamma}).
\end{aligned} \tag{29}$$

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

$$q_{\mathbf{e}}(\mathbf{e}) = \prod_{\tau=1}^t \prod_{i\in\Omega_\tau} \mathcal{N}(e_{i\tau} | \mu_e^{i\tau}, \Xi_e^{i\tau}). \tag{30}$$

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.

B. Update Equations

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:

$$\hat{\beta} = \frac{\omega}{\sum_{\tau=1}^t \sum_{i\in\Omega_\tau} \nu_{i\tau}} \tag{31}$$

where,

$$\begin{aligned}
\nu_{i\tau} :=& y_{i\tau}^2 - 2(y_{i\tau} - \mu_e^{i\tau})(\boldsymbol{\mu}_i^{\mathbf{A}})^T \boldsymbol{\mu}_\tau^{\mathbf{B}} - 2y_{i\tau}\mu_e^{i\tau} \\
&+ (\mu_e^{i\tau})^2 + \Xi_e^{i\tau} + \operatorname{tr}\left(\boldsymbol{\Sigma}_i^{\mathbf{A}} \boldsymbol{\Sigma}_{\tau,\tau}^{\mathbf{B}}\right).
\end{aligned} \tag{32}$$

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

$$\Xi_e^{i\tau} = \frac{1}{\hat{\beta} + (\mu_e^{i\tau})^2 + \Xi_e^{i\tau}} \tag{33a}$$

$$\mu_e^{i\tau} = \hat{\beta}\Xi_e^{i\tau}(y_{i\tau} - (\boldsymbol{\mu}_i^{\mathbf{A}})^T \boldsymbol{\mu}_\tau^{\mathbf{B}}). \tag{33b}$$

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:

$$\boldsymbol{\mu}_i^{\mathbf{A}} = \hat{\beta}\boldsymbol{\Xi}_i^{\mathbf{A}} \sum_{\tau\in\Omega_i'} \boldsymbol{\mu}_\tau^{\mathbf{B}}(y_{i\tau} - \mu_e^{i\tau}). \tag{34}$$

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 (20) which now looks like

$$\mathbf{v}_\tau = \hat{\beta} \sum_{i \in \Omega_\tau} (y_{i\tau} - \mu_e^{i\tau}). \tag{35}$$

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 (22) and (23), respectively.

Algorithm 2: Robust Variational Bayesian Subspace Filtering

Initialize $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\beta}, \mathbf{v}$, $sub = 1$, Ω_τ, Ω'_i, $\boldsymbol{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^{\mathbf{A}}, \boldsymbol{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^{\mathbf{J}} \boldsymbol{\Lambda}_1, \mu_1$,
$\hat{\mathbf{Y}} = \boldsymbol{\mu}^{\mathbf{A}} (\boldsymbol{\mu}^{\mathbf{B}})^T$
while $Y_{conv} < 10^{-5}$ **do**
  $\mathbf{Y_{old}} = \hat{\mathbf{Y}}$
  $\boldsymbol{\Gamma} = diag(\boldsymbol{\gamma})$
  if $sub == 1$ **then**
    Update using (21)
    $sub = 2$
    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 (33a), (33b) $\forall\ 1 \leq i \leq m,\ \forall\ 1 \leq \tau \leq t$
    $sub = 1$
  end
  $\hat{\mathbf{Y}} = \boldsymbol{\mu}^{\mathbf{A}} (\boldsymbol{\mu}^{\mathbf{B}})^T$
  Update using (31)
  $Y_{conv} = \frac{\|\mathbf{Y} - \mathbf{Y_{old}}\|_F}{\|\mathbf{Y_{old}}\|_F}$
end
return $(\hat{\mathbf{Y}}, \boldsymbol{\Xi}^{\mathbf{A}}, \boldsymbol{\mu}^{\mathbf{A}}, \boldsymbol{\Xi}^{\mathbf{B}}, \boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{J}}_{diag}, \boldsymbol{\mu}^{\mathbf{J}})$

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 ETA for which we provide simulation experiment results as well. To evaluate the VBSF algorithm, we use the partial road network of Delhi with an area of 200 square kms consisting of $m = 519$ edges. We collect the traffic speed data in the form of the average speed of vehicles on a particular segment using the Google map APIs for nearly 3 months across 519 edges. Taking advantage of the slow varying nature of the speed in the network edges, we sample the traffic speed data at the rate of one sample every $t_s = 15$ minutes. Unlike the complete data available from the API, real-world data may have missing entries. For instance, the smaller area shown in Fig. 3, speed measurements may be available on the blue edges but not on the red ones. Finally, we evaluate our algorithm for the twin tasks of real time traffic estimation as well as future traffic prediction. We further evaluate our algorithm for robust traffic estimation, i.e., when the traffic data is corrupted by outliers.

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

In order to evaluate the VBSF algorithm, an incomplete data set is created by randomly sampling a fraction p of the measurements. In our evaluations we consider three different cases with 0.75,0.5 and 0.25 fraction of present data. We select previous h = 30 time intervals for traffic speed dataset. We compare our algorithm with other methods that potentially solve the current traffic estimation problem in the missing data scenario. The algorithms are

- Low rank tensor completion (LRTC) [12].
- Grassmannian Rank-One Update Subspace Estimation (GROUSE) [11].
- Bayesian augmented tensor factorization(BATF) [49].
- Historic mean.

For future traffic prediction we compare with the following methods:

- k Nearest Neighbour (KNN) [50].
- Gaussian Process Regression (GPR) [51].
- Historic mean.

For the robust VBSF, we compare our algorithm with corresponding robust matrix completion frameworks.

- Robust PCA via Outlier Pursuit (OP-RPCA) [40].
- Robust Online Subspace Estimation and Tracking Algorithm (ROSETA) [41].
- Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) [39].

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 the traffic speed data. For any time instance τ, the MRE denoted by MRE_τ is defined as:

$$\text{MRE}_\tau = \frac{1}{z} \sum_{k=1}^{z} \frac{\| \hat{\mathbf{y}}_{\tau,k} - \mathbf{y}_{\tau,k} \|_2}{\| \mathbf{y}_{\tau,k} \|_2}. \tag{36}$$

where $\mathbf{y}_{\tau,k}$ and $\hat{\mathbf{y}}_{\tau,k}$ are the ground truth and estimated data for k^{th} day and τ^{th} time instance. Since the value for the known data (sampled entries) may be modified post estimation, we compute the MRE over the whole column for a given time instance. For calculating the overall accuracy of prediction for

a day, we calculate MRE averaged over z days. The value of z is taken as 50 for weekdays and 10 for the weekends.

B. Online 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. 4a shows the heatmap of the actual traffic speed data. The x-axis of each 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 4b, 4e and 4h are heatmaps with missing entries of varying degrees. The corresponding completed matrices using the VBSF algorithm are shown in Figs. 4c, 4f, and 4i. 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 improves as the size of missing data decreases.

The MRE values for real time traffic estimation using VBSF for weekends are shown in Fig. 5a. 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. A similar pattern is observed in the morning as shown in Fig. 6

The performance of the proposed VBSF algorithm is compared with that of (LRTC) [12], (GROUSE) [11], (BATF) [49] and the historic mean. We used a grid search based approach for rank initialization in GROUSE and choose the rank that gives the least error. Table I presents the overall results. Further, Figs. 7a and 7b show the comparison of our algorithm for different percentages of missing traffic data. It is observed that for a high sampling rate of traffic data (p=75%), the LRTC and VBSF obtain similar performance. And for the low sampling rate of traffic data (p=25%), the BATF and VBSF obtain similar performance. 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, BATF and is comparable to GROUSE as shown in Table II.

	$p = 0.25$ 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
BATF	0.1352	0.12067	0.10142
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	7.001	8.685	9.675
GROUSE	7.935	8.5324	9.23960
BATF	9.388	9.5911	9.94917
LRTC	29.2	43.2	62.3

TABLE II: Comparison of running time for different algorithms[1]

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 speed 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 are shown in Figs. 5b and 5c. 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. 5d.

The performance of the proposed VBSF algorithm is compared with that of kNN, GPR, LRTC in Table III. Since kNN and GPR models deals with complete data, we first impute the data using the VBSF and then predict the future traffic time series. This is shown in Table III with kNN(vbsf) and GPR(vbsf). Also, we report the results with the grouth truth for kNN and GPR in Table III. The performance of kNN and GPR is comparable in both cases which illustrate the effective imputation of VBSF.

	$p = 0.50$ $15\,mins$	$p = 0.50$ $30\,mins$
VBSF	0.15362	0.17434
kNN(vbsf)	0.17705	0.17963
GPR(vbsf)	0.1695	0.1773
kNN(with ground truth)	0.1762	0.1812
GPR(with ground truth)	0.1673	0.17413
LRTC	0.15843	0.1812
Mean	0.2082	0.2073

TABLE III: Performance comparison for traffic prediction

[1]Experiments are conducted to evaluate average running time per column on Matlab using PC: Intel i5-6200U CPU 2.4 GHz.

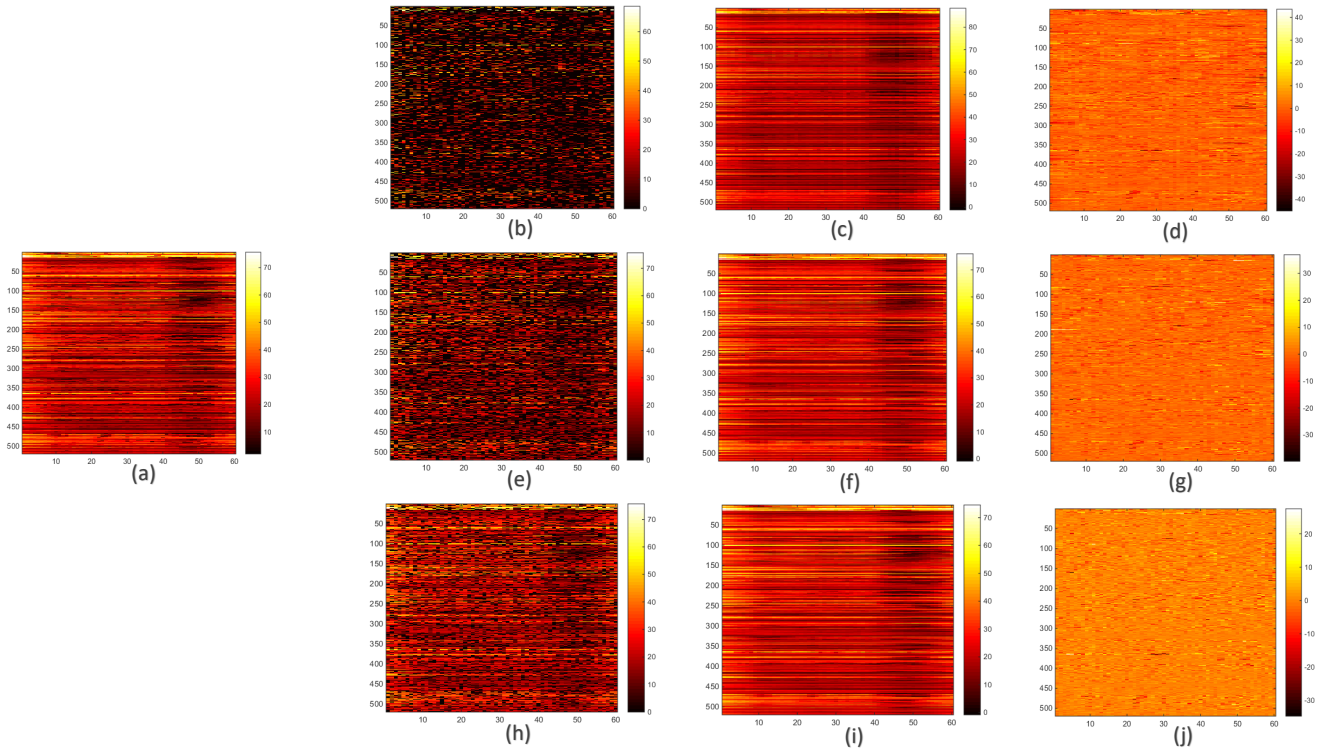


Fig. 4: Estimation of traffic speed for different percentage of missing entries
(a) Actual Traffic speed , (b) Traffic speed for $p = 0.25$, (c) Estimated Traffic for $p = 0.25$, (d) Residual error for estimation for $p = 0.25$, (e) Traffic speed for $p = 0.5$, (f) Estimated Traffic with for $p = 0.5$, (g) Residual error for estimation for $p = 0.5$, (h) Traffic speed for $p = 0.75$, (i) Estimated Traffic for $p = 0.75$, (j) Residual error for estimation for $p = 0.75$

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, 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 $\mathbf{y}_{i,\tau}$ and replace these values with $o_{i,\tau}$ as follows:

$$\mathbf{o}_{i,\tau} = \max\left(\mathbf{y}_{i,\tau-1}, \mathbf{y}_{i,\tau+1}\right) + c\,\mu_t. \tag{37}$$

In other words, the outlier is created by adding a large value $c\,\mu_t$ to the maximum of $\mathbf{y}_{i,\tau-1}$ and $\mathbf{y}_{i,\tau+1}$. Here, μ_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. 8d. 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, $p_o = 2\%$ and $c = 1.25$, the plots in Fig. 8a illustrates the performance of the RVBSF algorithm.

	$c = 0.75$ $p_o = 5\%$	$c = 0.75$ $p_o = 2\%$	$c = 1.5$ $p_o = 2\%$
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 [40] GRASTA [39] and ROSETA [41] in Table V. The RVBSF algorithm performs better than the subspace estimation and tracking algorithms. The difference in performance may be due to a better modeling of the temporal structure available in the data.

	$c = 0.75$ $p_o = 5\%$	$c = 0.75$ $p_o = 2\%$	$c = 1.5$ $p_o = 2\%$
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

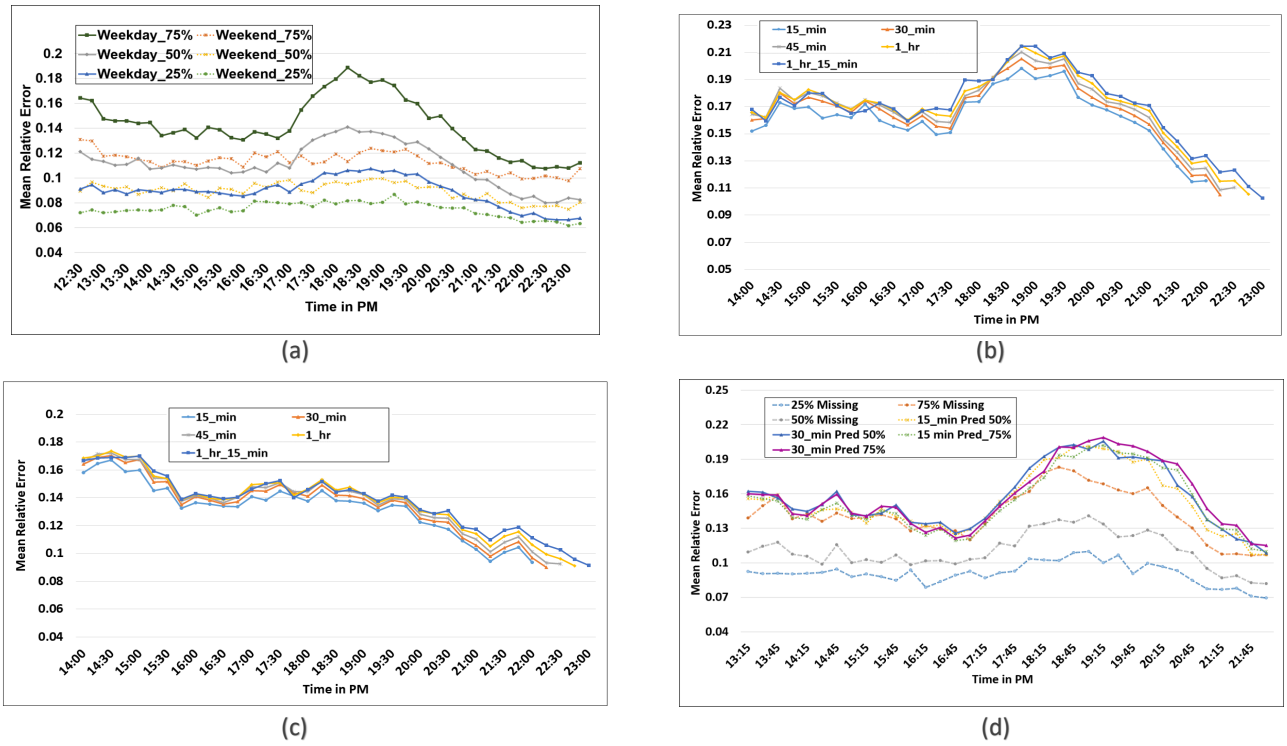


Fig. 5: Real time Traffic Estimation and Prediction for different missing entries
(a) Real time traffic estimation for different missing entries, (c) Weekday Prediction 50% missing entries, (d) Weekend Prediction 50% missing entries, (d) Overall Prediction

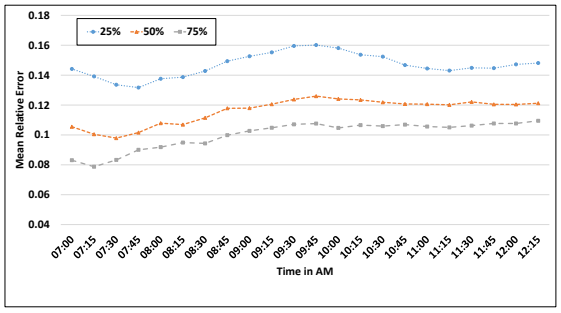


Fig. 6: Real time traffic estimation for different sampling percentage in the morning.

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 ignored by the model.

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 future traffic, which in return can be used to increase the reliability of both single occupancy vehicles as well as public transport.

References

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

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

[3] 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/}

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

[5] 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.

[6] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1816–1825, 2016.

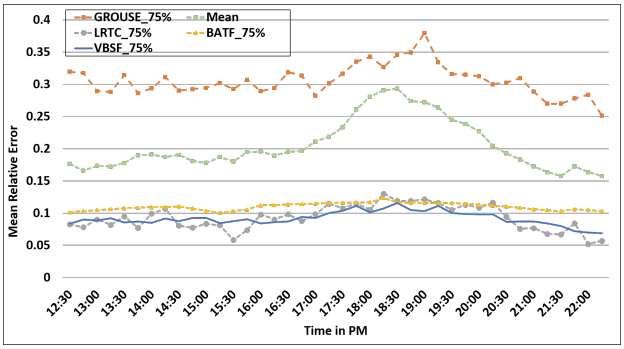
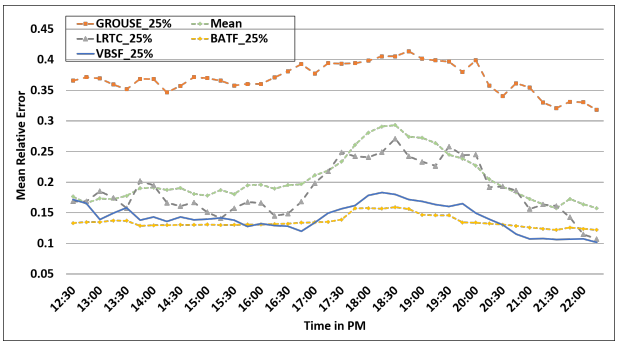
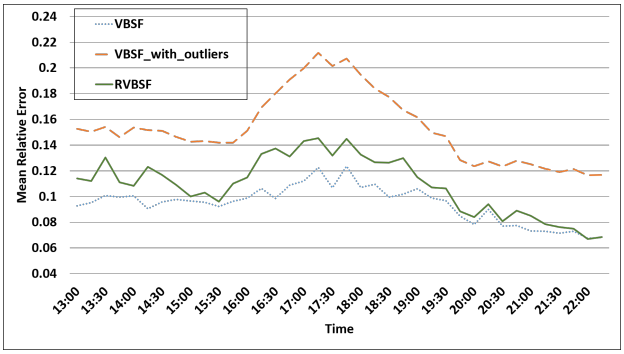


Fig. 7: a) Real time traffic estimation for $p = 0.75$,



b) Real time traffic estimation for $p = 0.25$.



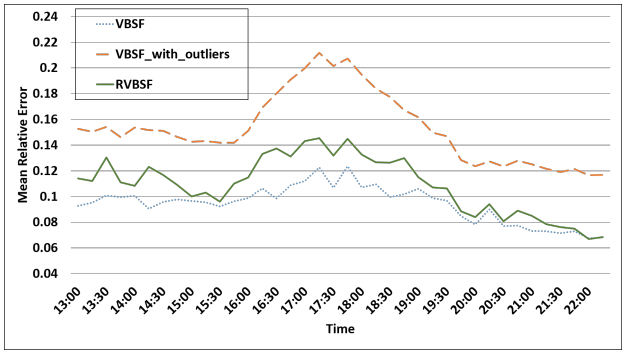


Fig. 8: a) Comparison of VBSF and RVBSF for $c = 1.25$ and $p_o = 2\%$, b) Number of outliers detected for different outlier values.

[7] L. Qu, L. Li, Y. Zhang, and J. Hu, "PPCA-based missing data imputation for traffic flow volume: A systematical approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 3, pp. 512–522, 2009.

[8] L. Qu, Y. Zhang, J. Hu, L. Jia, and L. Li, "A BPCA based missing value imputing method for traffic flow volume data," in *Proc. of the IEEE Symp. Intelligent Vehicles*, 2008, pp. 985–990.

[9] H. Tan, Y. Wu, B. Shen, P. J. Jin, and B. Ran, "Short-term traffic prediction based on dynamic tensor completion," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2123–2133, 2016.

[10] 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.

[11] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. of IEEE Allerton*, Sept. 2010, pp. 704–711.

[12] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 35, no. 1, pp. 208–220, 2013.

[13] 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.

[14] 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.

[15] 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.

[16] A. Stathopoulos and M. G. Karlaftis, "A multivariate state space approach for urban traffic flow modeling and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 2, pp. 121–135, 2003.

[17] 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.

[18] 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.

[19] 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.

[20] B. Ghosh, B. Basu, and M. O'Mahony, "Bayesian time-series model for short-term traffic flow forecasting," *Journal of transportation engineering*, vol. 133, no. 3, pp. 180–189, 2007.

[21] 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.

[22] 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.

[23] 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 Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 644–654, 2012.

[24] 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.

[25] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2015.

[26] 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.

[27] J. Zhao and S. Sun, "Variational dependent multi-output gaussian process dynamical systems," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 4134–4169, 2016.

[28] S. Sun and X. Xu, "Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 466–475, 2010.

[29] 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.

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

[31] J. T. Parker, P. Schniter, and V. Cevher, "Bilinear generalized approximate message passing Part I: Derivation," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5839–5853, 2014.

[32] ——, "Bilinear generalized approximate message passing Part II: Applications," *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, 2014.

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

[34] 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.

[35] 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.

[36] 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.

[37] Z. Ma, A. E. Teschendorff, A. Leijon, Y. Qiao, H. Zhang, and J. Guo, "Variational Bayesian matrix factorization for bounded support data," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 37, no. 4, pp. 876–889, 2015.

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

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

[40] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. of NIPS*, Vancouver, Canada, Dec. 2010, pp. 2496–2504.

[41] H. Mansour and X. Jiang, "A robust online subspace estimation and tracking algorithm," in *Proc. of the IEEE ICASSP*, Apr. 2015, pp. 4065–4069.

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

[43] M. T. Asif, N. Mitrovic, L. Garg, J. Dauwels, and P. Jaillet, "Low-dimensional models for missing data imputation in road networks," in *Proc. of the IEEE ICASSP*, May. 2013, pp. 3527–3531.

[44] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[45] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University of London London, 2003.

[46] D. G. Tzikas, A. C. Likas, and N. P. Galatsanos, "The variational approximation for Bayesian inference," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 131–146, 2008.

[47] M.-A. Sato, "Online model selection based on the variational Bayes," *Neural Computation*, vol. 13, no. 7, pp. 1649–1681, 2001.

[48] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[49] X. Chen, Z. He, Y. Chen, Y. Lu, and J. Wang, "Missing traffic data imputation and pattern discovery with a bayesian augmented tensor factorization model," *Transportation Research Part C: Emerging Technologies*, vol. 104, pp. 66–77, 2019.

[50] Y. Yin and P. Shang, "Forecasting traffic time series with multivariate predicting method," *Applied Mathematics and Computation*, vol. 291, pp. 266–278, 2016.

[51] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, p. 20110550, 2013.

Charul Paliwal received the B.Tech. degree from IGIT Delhi in 2016, the Mtech degree from IIIT Delhi, in 2018. She is currently pursuing the Ph.D. degree with Department of ECE, IIIT Delhi. Her research interests include transportation,spatio temporal signal processing and optimization

Uttkarsha Bhatt received the M.Tech degree from IIT Kanpur in 2017. She is currently Digital Design Engineer at Intel Corporation Bengaluru, Karnataka, India.

Pravesh Biyani received the B.Tech. degree from IIT Bombay in 2002, the M.S. degree from McMaster University, Canada, in 2004, and the Ph.D. degree from IIT Delhi in 2012. He is currently an Assistant Professor with the ECE Department at the Indraprastha Institute of Information Technology, New Delhi, India.

Ketan Rajawat received the B.Tech. and M.Tech. degrees in electrical engineering from the Indian Institute of Technology (IIT) Kanpur, India, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of Minnesota, Minneapolis, MN, USA, in 2012. He is currently an Assistant Professor with the Department of Electrical Engineering, IIT Kanpur. His research interests are in the broad areas of signal processing, robotics, and communications networks, with particular emphasis on distributed optimization. His current research interests include development and analysis of distributed and asynchronous optimization algorithms, online convex optimization algorithms, stochastic optimization algorithms, and the application of these algorithms to problems in machine learning, communications, and smart grid systems. He was a recipient of the 2018 INSA Medal for Young Scientists. He is currently an Associate Editor of the IEEE COMMUNICATIONS LETTERS.