

Online Variational Bayesian Subspace Filtering with Applications

Charul¹, *Student Member, IEEE*, Uttkarsha Bhatt², Pravesh Biyani¹, *Member, IEEE* and Ketan Rajawat³, *Member, IEEE*

Abstract—Matrix completion and robust principal component analysis have been widely used for the recovery of data suffering from missing entries or outliers. In many real-world applications however, the data is also time-varying, and the naive approach of per-snapshot recovery is both expensive and sub-optimal. This paper develops generative Bayesian models that fit sequential multivariate measurements arising from a low-dimensional time-varying subspace. A variational Bayesian subspace filtering approach is proposed that learns the underlying subspace and its state-transition matrix. Different from the plethora of deterministic counterparts, the proposed approach utilizes automatic relevance determination priors that obviate the need to tune key parameters such as rank and noise power. We also propose a forward-backward algorithm that allows the updates to be carried out at low complexity. Extensive tests over traffic and electricity data demonstrate the superior imputation, outlier rejection, and temporal prediction prowess of the proposed algorithm over the state-of-the-art matrix/tensor completion algorithms.

I. INTRODUCTION

Sensor measurements are often incomplete, noisy, and replete with outliers arising due to malfunctions or intermittent errors. Imputation of the missing entries and removal/segregation of the outliers is a critical first step that must be carried out prior to any data analytics. Examples of applications that benefit from such a pre-processing step include estimation/prediction of city-wide road traffic, regional air quality, electricity consumption in power distribution networks and foreground-background separation in videos. For most of these applications, the measurements can be arranged in form of a matrix, some of whose entries may be missing or contaminated with outliers. Pertinent approaches model the measurements as arising from a low-dimensional subspace whose recovery allows us to reject the noise and outliers, and impute the missing entries [1]–[6].

Many real-world applications, including the aforementioned ones, involve time-varying data that arrives in a sequential manner and must be processed as such. As a result, the data matrices arising in such applications comprise of low-dimensional subspaces that evolve over time. While the classical matrix completion or robust principal component analysis (RPCA) approaches are still applicable to each snapshot of the data, the performance can generally be improved by exploiting

the temporal correlations present in the measurements [2], [7]–[10]. State-of-the-art approaches for processing time-varying subspaces can mostly be classified into approaches based on tensor completion [7] and regularized matrix completion [11]. A common feature of these techniques is their static perspective and the resulting focus on batch processing. In contrast however, the data streaming from the sensors may be inherently dynamic, arising from subspaces that evolve over time. Theoretical guarantees for the dynamic setting have been studied in [12]. Different from these approaches and closer to the classical time-series modeling, an online forecasting matrix completion approach was proposed in [8] where the underlying subspace was assumed to follow a linear state-space model and must be learned in an online fashion. Approaches based on matrix completion often involve a number of tuning parameters that must be correctly set in order to avoid over-fitting. However determining these parameters via cross-validation is quite challenging with time-series data, especially in the online setting [8]. Alternatively, probabilistic learning algorithms have been proposed for the static matrix completion, and are generally free of tuning parameters. Such approaches entail constructing generative models that are not only capable of modeling the data but are also simple enough to allow low-complexity updates.

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 impute road traffic densities in an online fashion and from only a few observations. As the proposed models are generative, the resulting traffic density

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

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

predictions can also be used to obtain accurate expected time-of-arrival (ETA) estimates. Additionally, the applicability of the proposed algorithm on the electricity load estimation and prediction problem is also shown. The superior performance of our algorithm vis-a-vis other state of the art subspace tracking and online matrix factorisation algorithms may be attributed to the proposed state space model as well as the flexibility in the data modeling provided by the variational Bayesian approach. In summary, the contributions of the present work are as follows:

- 1) We present the variational Bayesian subspace filtering (VBSF) algorithm and demonstrate its ability to perform data modeling, imputation and temporal prediction in an online setting wherein the key algorithmic parameters are automatically tuned.
- 2) Robust version of the VBSF algorithm is also proposed for outlier removal and data cleansing.
- 3) Finally, we report a comprehensive comparison of our algorithm with various relevant (offline) matrix completion as well as online subspace estimation and tracking techniques, e.g, GROUSE [2], Low Rank Tensor Completion (LRTC) [7], GRASTA [9], ROSETA [10], OP-RPCA [13] and Online Forecasting Matrix Factorisation (OFMF) [14] over real-world traffic speed data as well as the electricity load data.

A. Related work

Variational Bayesian approaches for matrix completion and robust principal component analysis are well known [3], [4], [15]–[21]. 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 [15], [16]. More recently, other approaches towards modeling the measured matrices have also been proposed [17], [18]. Moreover, variational Bayesian approaches have also been applied to road traffic estimation; see e.g. [19]. 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], [18], [21]. In contrast to these, the state-space modeling in our work is inspired from [20], 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 [20] to incorporate low-rank structure and outliers.

On a related note, temporal evolution of the additive noise is modeled in [4] using a forgetting factor. Different from [4] however, we use a state-space model to capture the evolution of the underlying subspace. An online Bayesian matrix factorization model is also proposed in [13] wherein the time-stamps are directly incorporated as features. In contrast, the present model is more specific and suited to a slowly time-varying system.

Several non-Bayesian algorithms have been proposed to address the online subspace estimation problem from incomplete observations [2], [9], [10], [13]. GROUSE [2] 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 [9]. OP-RPCA [13] 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 [10], have since been proposed. The proposed approach is compared with some of these algorithms in Sec. IV.

B. Applications:

1) *Traffic Estimation and Prediction:* Traffic estimation and prediction are the central components of any urban traffic congestion management system [22]. 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 [23]. 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. 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 ETA, fastest route, and other related quality of service metrics for road users. 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 [24], [25]. For instance the ETA calculations for buses should not only use traffic data meant for cars. 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 [19], [26]–[28]. Complementary to these approaches, time-series modeling focuses on learning the temporal dynamics of traffic and generate predictions in an online manner [29]. While recent variants have incorporated spatial correlations as well, these techniques are generally unable to handle missing data or outliers. Finally, [14] presents the online forecasting matrix factorisation algorithm on the time series data that also handles the missing data scenario.

2) *Electricity Load Estimation and Prediction:* Similar to the traffic data, the electricity load data also exhibits the spatial and temporal structure that can be exploited to impute the missing data while simultaneously removing the noisy outliers. Due to the environmental disturbance, communication error or sensor fault, it is inevitable that load data may be lost during the collection process [30].

This paper is organized as follows. Sec. II presents the online variational Bayesian subspace filtering method for traffic estimation and prediction. Sec. III presents the online robust variational Bayesian subspace filtering method for traffic estimation and prediction in case of outliers. Results and findings for traffic prediction and electricity load prediction are discussed in Sec. IV followed by conclusion in Sec. 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 $\text{tr}(\mathbf{A})$, 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^T or $[\mathbf{A}]_i^T$. 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 $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma})$. 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 $\hat{\mathbf{x}} := \mathbb{E}[\mathbf{x} | \mathbf{D}]$.

II. VARIATIONAL BAYESIAN SUBSPACE FILTERING

We consider a scenario where the data with the missing entries is arriving in a sequential manner. The data can be considered in the form of the matrix $\mathbf{Y} \in \mathbb{R}^{m \times t}$, where t denotes the number of time instances over which measurements are made and m denotes the number of rows of the matrix \mathbf{Y} . More generally, \mathbf{Y} is an incomplete and growing matrix whose columns arrive sequentially over time. Specifically, for each column \mathbf{y}_τ with $1 \leq \tau \leq t$, only entries from the index set $\Omega_\tau \subset \{1, \dots, m\}$ 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 $\{\hat{y}_{t+\tau}\}_{\tau=1}^{T_p}$ where T_p is the prediction horizon.

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

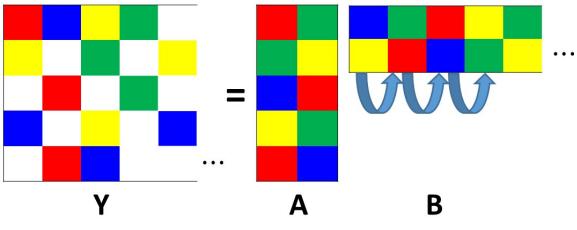


Fig. 1: Online Variational Bayesian Filtering

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} [31] but also the temporal correlation between successive columns of \mathbf{Y} [32]. 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 $\mathbf{B} \in \mathbb{R}^{r \times t}$. 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 [31]. 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 as

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

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:

$$p(\mathbf{b}_\tau | \mathbf{J}, \mathbf{b}_{\tau-1}) = \mathcal{N}(\mathbf{b}_\tau | \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \quad 2 \leq \tau \leq t \quad (2)$$

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}_τ without the need to explicitly model the state noise variance. It follows from (2) that the conditional pdf of \mathbf{b}_τ given \mathbf{J} is given by

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

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.,

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

$$p(\mathbf{J} | \boldsymbol{v}) = \prod_{i=1}^r \mathcal{N}(\mathbf{j}_i | 0, v_i^{-1} \mathbf{I}_r) \quad (5)$$

where the precisions $\boldsymbol{\gamma}$ and \boldsymbol{v} are problem parameters. It can be seen that if any of γ_i or v_i are large, the corresponding columns will be close to zero and consequently irrelevant. Indeed, the priors in (4)-(5) 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 non-informative Jeffrey's priors

$$p(\beta) = \frac{1}{\beta}, \quad p(\gamma_i) = \frac{1}{\gamma_i}, \quad p(v_i) = \frac{1}{v_i} \quad (6)$$

for $1 \leq i \leq r$. Let \mathbf{y}_Ω denote the collection of measurements $\{y_{i\tau}\}_{i \in \Omega_\tau, \tau=1}^t$. Collecting the hidden variables into

$\mathcal{H} := \{\mathbf{A}, \mathbf{B}, \mathbf{J}, \beta, \gamma, \mathbf{v}\}$, the joint distribution of $\{\mathbf{y}_\Omega, \mathcal{H}\}$ can be written as

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

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

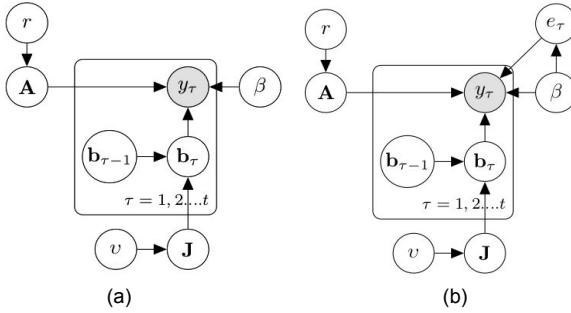


Fig. 2: (a) Hierarchical Bayesian Model for Matrix Completion
(b) Robust Hierarchical Bayesian Model for Matrix Completion

B. Variational Bayesian Inference

Having specified the generative model for the data, the goal is to determine the posterior distribution $p(\mathcal{H} | \mathbf{y}_\Omega)$, which would yield the corresponding point estimates and can be used for imputation and prediction tasks. However, exact full Bayesian inference is well-known to be intractable. Instead, we utilize the mean-field approximation, wherein the posterior distribution factorizes as:

$$p(\mathcal{H} | \mathbf{y}_\Omega) \approx q(\mathcal{H}) = q_{\mathbf{A}}(\mathbf{A}) q_{\mathbf{B}}(\mathbf{B}) q_{\mathbf{J}}(\mathbf{J}) q_{\mathbf{v}}(\mathbf{v}) q_{\beta}(\beta) q_{\gamma}(\gamma). \quad (8)$$

In other words, the posterior is now restricted to a family of distributions that adhere to (8). The factors $q_{\mathbf{A}}$, $q_{\mathbf{B}}$, $q_{\mathbf{J}}$, $q_{\mathbf{v}}$, 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 [33]. 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 [20]:

$$q_{\mathbf{B}}(\mathbf{B}) = \mathcal{N}(\text{vec}(\mathbf{B}) | \boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{B}}) \quad (9a)$$

$$q_{\mathbf{a}_{i\cdot}} = \mathcal{N}(\mathbf{a}_{i\cdot} | \boldsymbol{\mu}_i^{\mathbf{A}}, \boldsymbol{\Xi}_i^{\mathbf{A}}) \quad (9b)$$

$$q_{\mathbf{j}_{i\cdot}} = \mathcal{N}(\mathbf{j}_{i\cdot} | \boldsymbol{\mu}_i^{\mathbf{J}}, \boldsymbol{\Xi}_i^{\mathbf{J}}) \quad (9c)$$

$$q_{\beta}(\beta) = \text{Ga}(\beta; a^\beta, b^\beta) \quad (9d)$$

$$q_{\gamma_i}(\gamma_i) = \text{Ga}(\gamma_i; a_i^\gamma, b_i^\gamma) \quad (9e)$$

$$q_{v_i}(v_i) = \text{Ga}(v_i; a_i^v, b_i^v) \quad (9f)$$

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 $a^\beta, b^\beta, a_i^\gamma, b_i^\gamma, a_i^v, b_i^v \in \mathbb{R}_{++}$. Consequently, each iteration of alternating optimization simply involves updating the variables $\{\boldsymbol{\mu}^{\mathbf{B}}, \boldsymbol{\Xi}^{\mathbf{B}}, \{\boldsymbol{\mu}_i^{\mathbf{A}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{A}}\}, \{\boldsymbol{\mu}_i^{\mathbf{J}}\}, \{\boldsymbol{\Xi}_i^{\mathbf{J}}\}, a^\beta, b^\beta, \{a_i^\gamma\}, \{b_i^\gamma\}, \{a_i^v\}, \{b_i^v\}\}$ 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 $\{\mathbf{v}, \gamma\}$ take the following form

$$\hat{v}_i = \frac{m}{\sum_{k=1}^m ([\boldsymbol{\mu}_k^{\mathbf{J}}]_i^2 + [\boldsymbol{\Xi}_k^{\mathbf{J}}]_{ii})} \quad (10a)$$

$$\hat{\gamma}_i = \frac{m}{\sum_{k=1}^m ([\boldsymbol{\mu}_k^{\mathbf{A}}]_i^2 + [\boldsymbol{\Sigma}_k^{\mathbf{A}}]_{ii})}. \quad (10b)$$

Subsequently, let $\hat{\mathbf{v}}$ and $\hat{\gamma}$ be the vectors that collect $\{\hat{v}_i\}$ and $\{\hat{\gamma}_i\}$, respectively. Since \mathbf{b}_τ denotes the τ -th column of \mathbf{B}^T , its posterior distribution may be written as $q_{\mathbf{b}_\tau}(\mathbf{b}_\tau) = \mathcal{N}(\mathbf{b}_\tau | \boldsymbol{\mu}_\tau^{\mathbf{B}}, \boldsymbol{\Xi}_\tau^{\mathbf{B}})$, 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

$$\boldsymbol{\Sigma}_{\tau,\ell}^{\mathbf{B}} := \boldsymbol{\mu}_\ell^{\mathbf{B}} (\boldsymbol{\mu}_\ell^{\mathbf{B}})^T + \boldsymbol{\Xi}_{\tau,\ell}^{\mathbf{B}} \quad (11)$$

$$\boldsymbol{\Sigma}_i^{\mathbf{J}} := \boldsymbol{\mu}_i^{\mathbf{J}} (\boldsymbol{\mu}_i^{\mathbf{J}})^T + \boldsymbol{\Xi}_i^{\mathbf{J}} \quad (12)$$

$$\boldsymbol{\Sigma}_i^{\mathbf{A}} := \boldsymbol{\mu}_i^{\mathbf{A}} (\boldsymbol{\mu}_i^{\mathbf{A}})^T + \boldsymbol{\Xi}_i^{\mathbf{A}}. \quad (13)$$

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

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

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

$$\boldsymbol{\mu}_i^{\mathbf{J}} = [\boldsymbol{\Xi}_i^{\mathbf{J}} \boldsymbol{\Sigma}_{\tau,\tau-1}^{\mathbf{B}}]_{\cdot i} \quad (15a)$$

$$\boldsymbol{\Xi}_i^{\mathbf{J}} = \left(\text{Diag}(\hat{\mathbf{v}}) + \sum_{\tau=1}^{t-1} \boldsymbol{\Sigma}_{\tau,\tau-1}^{\mathbf{B}} \right)^{-1} \quad (15b)$$

$$\boldsymbol{\mu}_i^{\mathbf{A}} = \hat{\beta} \boldsymbol{\Xi}_i^{\mathbf{A}} \sum_{\tau \in \Omega'_i} \boldsymbol{\mu}_\tau^{\mathbf{B}} y_{i\tau} \quad (15c)$$

$$\boldsymbol{\Xi}_i^{\mathbf{A}} = \left(\hat{\gamma}_i \mathbf{I}_r + \hat{\beta} \sum_{\tau \in \Omega'_i} \boldsymbol{\Sigma}_{\tau,\tau}^{\mathbf{B}} \right)^{-1} \quad (15d)$$

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 $\boldsymbol{\mu}^B$ can be written as

$$\boldsymbol{\mu}^B = \Xi^B \begin{bmatrix} \hat{\beta} \sum_{i \in \Omega_1} y_{i1} \boldsymbol{\mu}_i^A + \Lambda_1^{-1} \boldsymbol{\mu}_1 \\ \hat{\beta} \sum_{i \in \Omega_2} y_{i2} \boldsymbol{\mu}_i^A \\ \vdots \\ \hat{\beta} \sum_{i \in \Omega_t} y_{it} \boldsymbol{\mu}_i^A \end{bmatrix}. \quad (16)$$

Finally, $[\Xi^B]^{-1}$ 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 $(\boldsymbol{\mu}_i^J)^T$, $\Sigma_{(\tau)}^A = \sum_{i \in \Omega_\tau} \Sigma_i^A$, and $\Sigma^J := \sum_{i=1}^r \Sigma_i^J$, the updates take the form:

$$[\Xi^B]^{-1} = \hat{\beta} \text{Diag} \left(\Xi_{(1)}^A, \dots, \Xi_{(t)}^A \right) + \\ + \begin{bmatrix} \Lambda_1^{-1} & -\hat{\mathbf{J}} & \dots & 0 \\ -\hat{\mathbf{J}} & \mathbf{I}_r + \Sigma^J & -\hat{\mathbf{J}} & \dots \\ \vdots & \vdots & & \vdots \\ \dots & 0 & -\hat{\mathbf{J}} & \mathbf{I}_r \end{bmatrix}. \quad (17)$$

It is remarked that although the $rt \times rt$ matrix $[\Xi^B]^{-1}$ is block-tridiagonal, the matrix Ξ^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 [34]. Interestingly, observe that the updates in (14) and (15) depend only on diagonal and super-diagonal blocks of Ξ^B , namely $\Xi_{\tau,\tau}^B$ and $\Xi_{\tau,\tau+1}^B$, respectively. The next subsection details a low-complexity algorithm for carrying out the updates for these blocks as well as for $\boldsymbol{\mu}^B$.

C. Low-complexity updates via LDL-decomposition

Thanks to the block-tridiagonal structure of $[\Xi^B]^{-1}$, it is possible to use the LDL decomposition to carry out the updates in an efficient manner. Decomposing $[\Xi^B]^{-1} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, the key idea is that left multiplication with Ξ^B is equivalent to left multiplication with $\mathbf{L}^{-T}\mathbf{D}^{-1}\mathbf{L}^{-1}$. Towards this end, we utilize the algorithm from [20], 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 $2 \leq \tau \leq t$,

$$\Psi_\tau := \hat{\beta} \sum_{i \in \Omega_\tau} \Sigma_{(i)}^A + \mathbf{I}_r + \mathbf{1}_{\tau \neq t} \sum_{i=1}^r \Sigma_i^J \quad (18)$$

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

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

- 1) Initialize $\hat{\Xi}_{1,1}^B = \Lambda_1$ and $\hat{\boldsymbol{\mu}}_1^B = \boldsymbol{\mu}_1 + \hat{\beta} \sum_{i \in \Omega_1} y_{i1} \boldsymbol{\Lambda}_1 \boldsymbol{\mu}_i^A$
- 2) For $\tau = 1, \dots, t-1$

$$\check{\Xi}_{\tau,\tau+1}^B = -\hat{\Xi}_{\tau,\tau}^B \hat{\mathbf{J}} \quad (20a)$$

$$\check{\Xi}_{\tau+1,\tau+1}^B = (\Psi_{\tau+1} - (\check{\Xi}_{\tau,\tau+1}^B)^T \Psi_{\tau,\tau+1}^B)^{-1} \quad (20b)$$

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

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

$$\Xi_{\tau,\tau+1}^B = -\check{\Xi}_{\tau,\tau+1}^B \Xi_{\tau+1,\tau+1}^B \quad (20d)$$

$$\Xi_{\tau,\tau}^B = \check{\Xi}_{\tau,\tau}^B - \hat{\Xi}_{\tau,\tau+1}^B (\Xi_{\tau,\tau+1}^B)^T \quad (20e)$$

$$\boldsymbol{\mu}_\tau^B = \check{\boldsymbol{\mu}}_\tau^B - \check{\Xi}_{\tau,\tau+1}^B \boldsymbol{\mu}_{\tau+1}^B \quad (20f)$$

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

Note that while $\Xi_{i,j}^B \neq 0$ 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

$$y_{i\tau} = (\boldsymbol{\mu}_\tau^B)^T \boldsymbol{\mu}_i^A. \quad (21)$$

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

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

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 $t = 1, 2, \dots$.

D. EM Bayesian 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 $q_h(\mathcal{H}_h) := q_B(\mathbf{B})q_A(\mathbf{A})q_J(\mathbf{J})$) and uses maximum a posteriori (MAP) estimates for the precision variables $\mathcal{H}_p := \{\boldsymbol{v}, \gamma, \beta\}$. 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

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

- **M-step:** maximize

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

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:

$$\hat{v}_i = \frac{m-2}{\sum_{k=1}^m ([\boldsymbol{\mu}_k^J]_i^2 + [\Xi_k^J]_{ii})} \quad (25a)$$

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

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

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 $\max_x \text{Ga}(x|a, b) = \frac{a-1}{b}$.

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. [35], [36]. 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 [37] for a more general result. However, convergence to the global optimum is not guaranteed.

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.

Algorithm 1: Variational Bayesian Subspace Filtering

```

1 Initialize  $\gamma, \beta, \mathbf{v}$ ,
   sub = 1,  $\Omega_\tau, \Omega'_i, \Xi^{\mathbf{A}}, \mu^{\mathbf{A}}, \Xi^{\mathbf{B}}, \mu^{\mathbf{B}}, \Xi_{diag}^{\mathbf{J}}, \mu^{\mathbf{J}} \Lambda_1, \mu_1,$ 
2  $\hat{\mathbf{Y}} = \mu^{\mathbf{A}}(\mu^{\mathbf{B}})^T$ 
3 while  $Y_{conv} < 10^{-5}$  do
4    $\mathbf{Y}_{old} = \hat{\mathbf{Y}}$ 
5    $\Gamma = diag(\gamma)$ 
6   if sub == 1 then
7     Update using (20)
8     sub = 2
9     Update using (10a), (11), (15a), (15b)  $\forall 1 \leq i \leq r$ 
10  else if sub == 2 then
11    Update using (13), (15c), (15d), (10b)  $\forall 1 \leq i \leq m$ 
12    sub = 1
13  end
14   $\hat{\mathbf{Y}} = \mu^{\mathbf{A}}(\mu^{\mathbf{B}})^T$ 
15  Update using (14)
16   $Y_{conv} = \frac{\|\mathbf{Y} - \mathbf{Y}_{old}\|_F}{\|\mathbf{Y}_{old}\|_F}$ 
17 end
18 return ( $\hat{\mathbf{Y}}, \Xi^{\mathbf{A}}, \mu^{\mathbf{A}}, \Xi^{\mathbf{B}}, \mu^{\mathbf{B}}, \Xi_{diag}^{\mathbf{J}}, \mu^{\mathbf{J}}$ )

```

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. These 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 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:

$$p(y_{i\tau} | \mathbf{a}_i, \mathbf{b}_\tau, e_{i\tau}, \beta) = \mathcal{N}(y_{i\tau} | \mathbf{b}_\tau^T \mathbf{a}_i + e_{i\tau}, \beta^{-1}) \quad i \in \Omega_\tau \quad (26)$$

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.,

$$p(\mathbf{B} | \mathbf{J}) = \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^t \mathcal{N}(\mathbf{b}_\tau | \mathbf{J}\mathbf{b}_{\tau-1}, \mathbf{I}_r) \quad (27a)$$

$$p(\mathbf{A} | \boldsymbol{\gamma}) = \prod_{i=1}^r \mathcal{N}(\mathbf{a}_i | 0, \gamma_i^{-1} \mathbf{I}) \quad (27b)$$

$$p(\mathbf{J} | \boldsymbol{v}) = \prod_{i=1}^r \mathcal{N}(\mathbf{j}_i | 0, v_i^{-1} \mathbf{I}) \quad (27c)$$

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

$$p(e_{i\tau}) = \mathcal{N}(e_{i\tau} | 0, \alpha_{i\tau}^{-1}) \quad i \in \Omega_\tau \quad (28)$$

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 , $\{\gamma_i\}$, $\{v_i\}$, and $\{\alpha_{i\tau}\}$.

$$p(\beta) = \frac{1}{\beta}, \quad p(\gamma_i) = \frac{1}{\gamma_i}, \quad p(v_i) = \frac{1}{v_i}, \quad p(\alpha_{i\tau}) = \frac{1}{\alpha_{i\tau}}. \quad (29)$$

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 $\mathcal{H} := \{\mathbf{A}, \mathbf{B}, \mathbf{J}, \mathbf{e}, \beta, \gamma, \mathbf{v}\}$, the joint distribution of $\{\mathbf{y}_\Omega, \mathcal{H}\}$ can be written as

$$\begin{aligned} p(\mathbf{y}_\Omega, \mathcal{H}) &= p(\mathbf{y}_\Omega | \mathbf{A}, \mathbf{B}, \beta) p(\mathbf{A} | \gamma) p(\mathbf{B} | \mathbf{J}) p(\mathbf{J} | \mathbf{v}) p(\mathbf{e} | \alpha) p(\beta) p(\mathbf{v}) p(\gamma) \\ &= \prod_{\tau=1}^t \prod_{i \in \Omega_\tau} \mathcal{N}(y_{i\tau} | \mathbf{b}_\tau^T \mathbf{a}_{i\cdot}, \beta^{-1}) \mathcal{N}(e_{i\tau} | 0, \alpha_{i\tau}^{-1}) \frac{1}{\alpha_{i\tau}} \\ &\quad \times \prod_{i=1}^r [\mathcal{N}(\mathbf{a}_i | 0, \gamma_i^{-1} \mathbf{I}) \mathcal{N}(\mathbf{j}_i | 0, v_i^{-1} \mathbf{I})] \\ &\quad \times \mathcal{N}(\mathbf{b}_1; \boldsymbol{\mu}_1, \boldsymbol{\Lambda}_1) \prod_{\tau=2}^t \mathcal{N}(\mathbf{b}_\tau | \mathbf{J} \mathbf{b}_{\tau-1}, \mathbf{I}) \frac{1}{\beta} \prod_{i=1}^r \frac{1}{\gamma_i v_i}. \quad (30) \end{aligned}$$

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 $p(\mathcal{H} | \mathbf{y}_\Omega)$ factorizes as

$$\begin{aligned} p(\mathcal{H} | \mathbf{y}_\Omega) &\approx q(\mathcal{H}) \\ &= q_{\mathbf{A}}(\mathbf{A}) q_{\mathbf{B}}(\mathbf{B}) q_{\mathbf{J}}(\mathbf{J}) q_{\mathbf{e}}(\mathbf{e}) q_{\mathbf{v}}(\mathbf{v}) q_\beta(\beta) q_\gamma(\gamma). \quad (31) \end{aligned}$$

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

$$q_{\mathbf{e}}(\mathbf{e}) = \prod_{\tau=1}^t \prod_{i \in \Omega_\tau} \mathcal{N}(e_{i\tau} | \mu_e^{i\tau}, \Xi_e^{i\tau}). \quad (32)$$

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

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

$$\hat{\beta} = \frac{\omega}{\sum_{\tau=1}^t \sum_{i \in \Omega_\tau} \nu_{i\tau}} \quad (33)$$

where,

$$\begin{aligned} \nu_{i\tau} &:= y_{i\tau}^2 - 2(y_{i\tau} - \mu_e^{i\tau})(\boldsymbol{\mu}_i^{\mathbf{A}})^T \boldsymbol{\mu}_\tau^{\mathbf{B}} - 2y_{i\tau} \mu_e^{i\tau} \\ &\quad + (\mu_e^{i\tau})^2 + \Xi_e^{i\tau} + \text{tr}(\Sigma_i^{\mathbf{A}} \Sigma_{\tau,\tau}^{\mathbf{B}}). \quad (34) \end{aligned}$$

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

$$\Xi_e^{i\tau} = \frac{1}{\hat{\beta} + (\mu_e^{i\tau})^2 + \Xi_e^{i\tau}} \quad (35a)$$

$$\mu_e^{i\tau} = \hat{\beta} \Xi_e^{i\tau} (y_{i\tau} - (\boldsymbol{\mu}_i^{\mathbf{A}})^T \boldsymbol{\mu}_\tau^{\mathbf{B}}). \quad (35b)$$

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

$$\boldsymbol{\mu}_i^{\mathbf{A}} = \hat{\beta} \Xi_i^{\mathbf{A}} \sum_{\tau \in \Omega'_i} \boldsymbol{\mu}_\tau^{\mathbf{B}} (y_{i\tau} - \mu_e^{i\tau}). \quad (36)$$

Finally, the updates for $\Xi^{\mathbf{B}}$ 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}_τ in (19) which now looks like

$$\mathbf{v}_\tau = \hat{\beta} \sum_{i \in \Omega_\tau} (y_{i\tau} - \mu_e^{i\tau}). \quad (37)$$

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.

Algorithm 2: Robust Variational Bayesian Subspace Filtering

```

1 Initialize  $\boldsymbol{\alpha}, \gamma, \beta, \mathbf{v}$ ,
     $sub = 1, \Omega_\tau, \Omega'_i, \Xi^{\mathbf{A}}, \boldsymbol{\mu}^{\mathbf{A}}, \Xi^{\mathbf{B}}, \boldsymbol{\mu}^{\mathbf{B}}, \Xi_{diag}^{\mathbf{J}}, \boldsymbol{\mu}^{\mathbf{J}}, \boldsymbol{\Lambda}_1, \mu_1$ ,
2  $\hat{\mathbf{Y}} = \boldsymbol{\mu}^{\mathbf{A}} (\boldsymbol{\mu}^{\mathbf{B}})^T$ 
3 while  $Y_{conv} < 10^{-5}$  do
4    $\mathbf{Y}_{old} = \hat{\mathbf{Y}}$ 
5    $\boldsymbol{\Gamma} = diag(\gamma)$ 
6   if  $sub == 1$  then
7     Update using (20)
8      $sub = 2$ 
9     Update using (10a), (11), (15a), (15b)  $\forall 1 \leq i \leq r$ 
10  else if  $sub == 2$  then
11    Update using (13), (15c), (15d), (10b)  $\forall 1 \leq i \leq m$ 
12     $sub = 3$ 
13  end
14  else
15    Update using (35a), (35b)  $\forall 1 \leq i \leq m, \forall 1 \leq$ 
       $\tau \leq t$ 
16     $sub = 1$ 
17  end
18   $\hat{\mathbf{Y}} = \boldsymbol{\mu}^{\mathbf{A}} (\boldsymbol{\mu}^{\mathbf{B}})^T$ 
19  Update using (33)
20   $Y_{conv} = \frac{\|\mathbf{Y} - \mathbf{Y}_{old}\|_F}{\|\mathbf{Y}_{old}\|_F}$ 
21 return  $(\hat{\mathbf{Y}}, \Xi^{\mathbf{A}}, \boldsymbol{\mu}^{\mathbf{A}}, \Xi^{\mathbf{B}}, \boldsymbol{\mu}^{\mathbf{B}}, \Xi_{diag}^{\mathbf{J}}, \boldsymbol{\mu}^{\mathbf{J}})$ 

```

IV. RESULTS

We now detail the simulation results that evaluate the performance of the proposed VBSF method on variety of datasets to solve the:

- 1) Traffic Estimation and Prediction Problem
- 2) Electricity Load Estimation and Prediction Problem

A. Datasets

- Traffic: for traffic estimation and prediction, 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. 3). The road network can be modeled using a directed graph where each edge represents a road segment and nodes represent intersections. We collect the traffic data in the form of 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 data at the rate of one sample every



Fig. 3: Region where traffic data is collected



Fig. 4: Map with red as missing and blue as known traffic entries

$t_s = 15$ minutes. Note that our algorithm is agnostic of the sampling rate and would work for higher sampling rates as well. Unlike the complete data available from the API, real-world data may have missing entries. For instance, over the smaller area shown in Fig. 4, 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 we the traffic data is corrupted by outliers.

- Electricity: similar to the traffic estimation and prediction task, we evaluate the VBSF algorithm on the electricity dataset [8], also used in [14] to evaluate the online matrix factorisation method. The data contains the hourly power consumption of 370 consumers, sampled every 15 min. The data is recorded from Jan. 1, 2012 to Jan. 1, 2015. Finally, we compare the VBSF method with various methods including the ones proposed and compared in [14].

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 75%, 50%, and 25% of missing data. We select previous $h = 30$ time intervals for traffic and, the previous

$h = 40$ time intervals for electricity 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) [7].
- Grassmannian Rank-One Update Subspace Estimation (GROUSE) [2].
- Historic mean, which is simply the mean of edge speed values at a given time instance calculated using the historic data.

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

- Robust PCA via Outlier Pursuit (OP-RPCA) [13].
- Robust Online Subspace Estimation and Tracking Algorithm (ROSETA) [10].
- Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) [9].

Further, for the electricity load prediction problem, we compare our algorithm with the results of [14] and the Collaborative Kalman Filter (CKF) [14].

B. Traffic Estimation and Prediction Problem

1) *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 data. For any time instance τ , the MRE denoted by MRE_τ is defined as:

$$MRE_\tau = \frac{1}{z} \sum_{k=1}^z \frac{\|\hat{\mathbf{y}}_{\tau,k} - \mathbf{y}_{\tau,k}\|_2}{\|\mathbf{y}_{\tau,k}\|_2}. \quad (38)$$

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.

2) *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. 5a shows the heatmap of the actual traffic 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 5b, 5e and 5h are heatmaps with missing entries of varying degrees. The corresponding completed matrices using VBSF algorithm are shown in Figs. 5c, 5f, and 5i. 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.

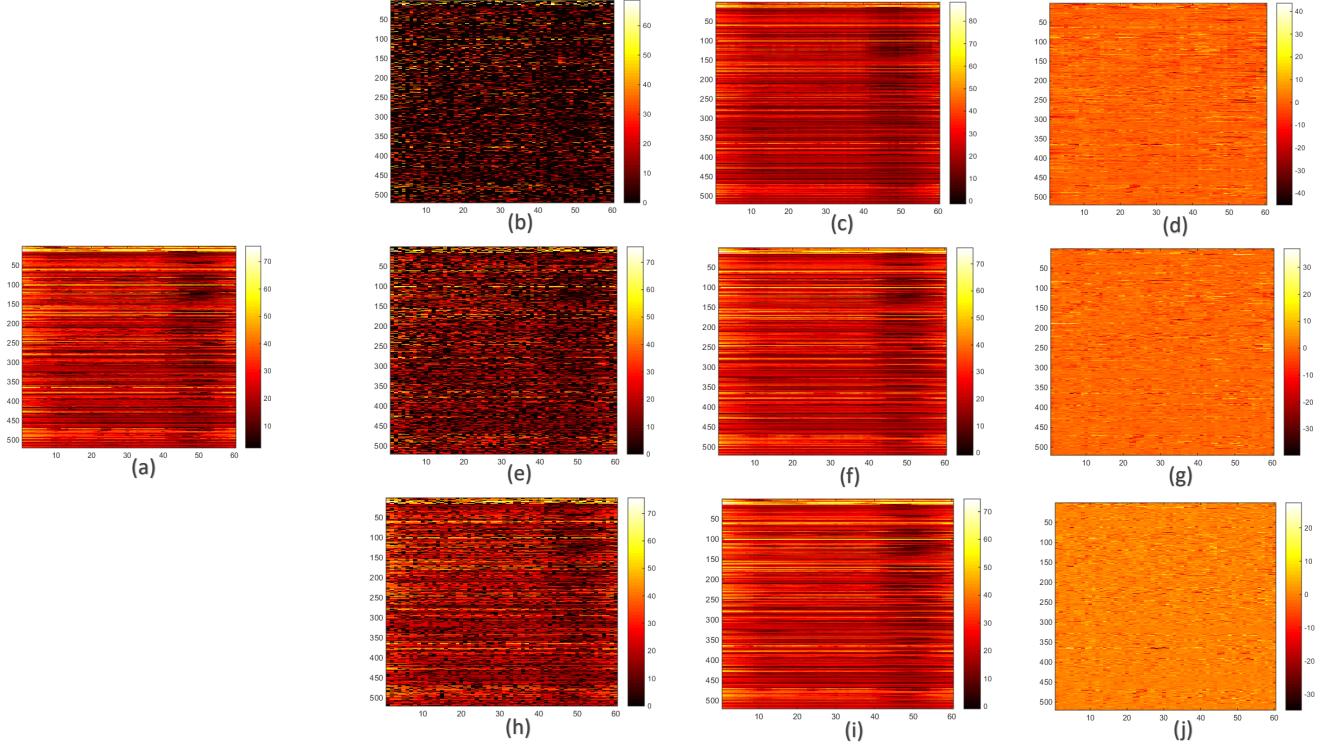


Fig. 5: Estimation of traffic data for different percentage of missing entries

(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

The MRE values for real time traffic estimation using VBSF for weekends is shown in Fig. 6a and for weekdays in Fig. 6b. 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.

The performance of the proposed VBSF algorithm is compared with that of (LRTC) [7], (GROUSE) [2], 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 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.

	$p = 0.25$ 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 algorithms¹

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

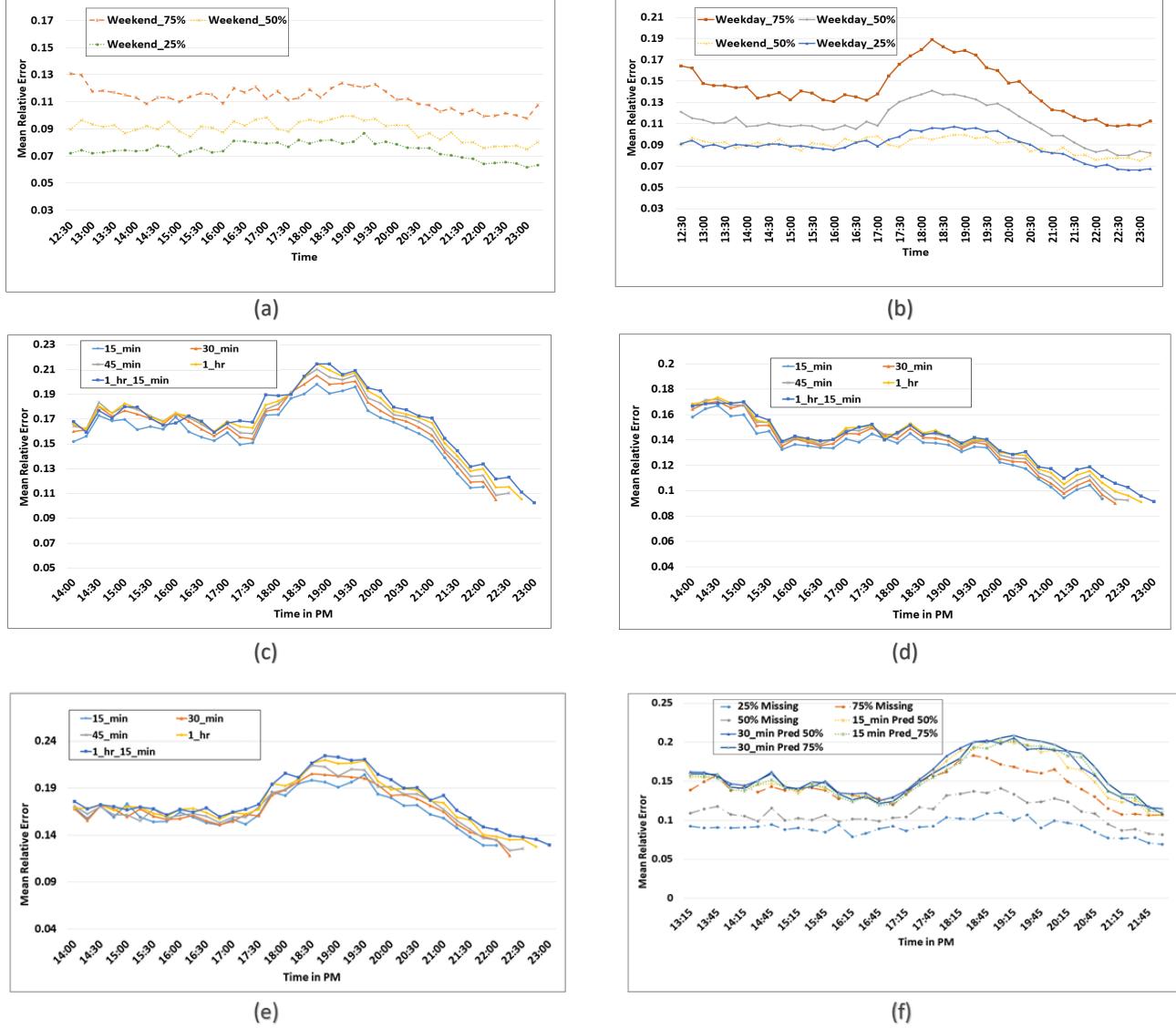


Fig. 6: Real time Traffic Estimation and Prediction for different missing entries

- (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

3) 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 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 are shown in Figs. 6c, 6d, and 6e. 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. 6f.

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. 7c. 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. 7d.

	$p = 0.50$ 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

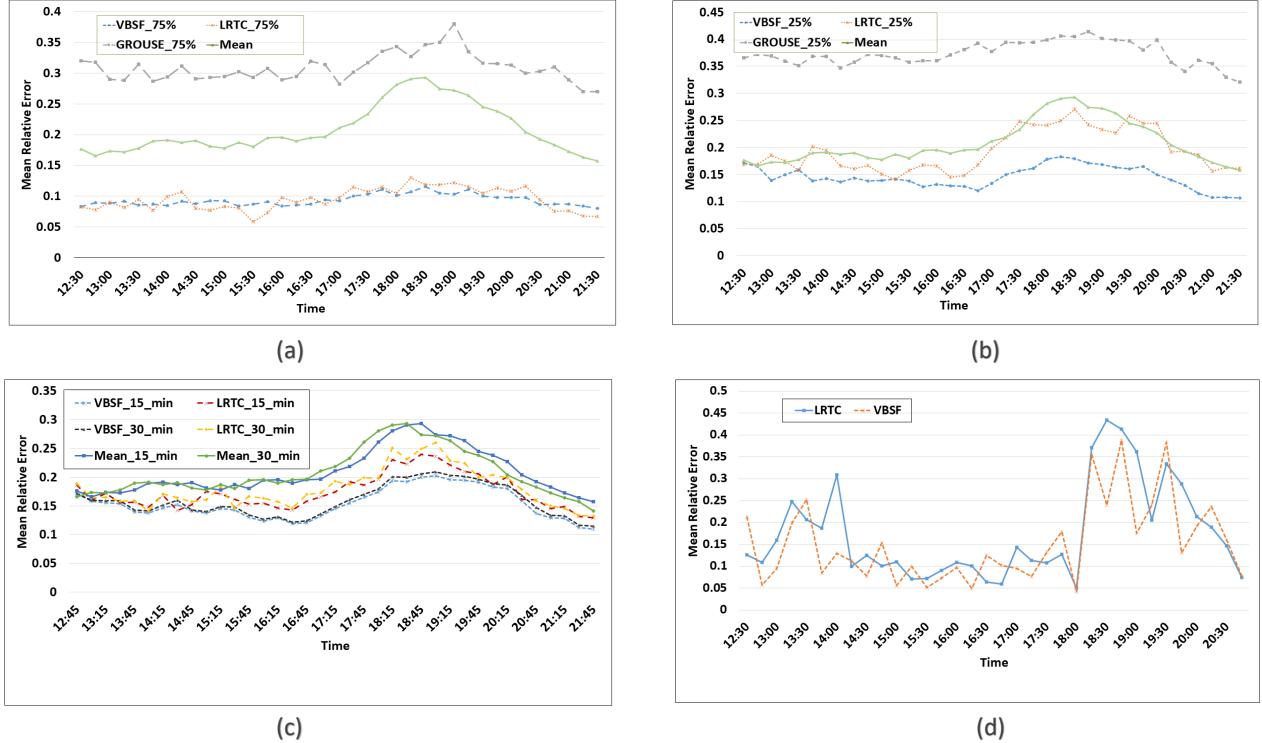


Fig. 7: Comparison between VBSF and Low rank Tensor Completion (LRTC) and Matrix Completion Algorithm (GROUSE)
(a) Real Time Traffic Estimation for 25% percentage of Missing Data, (b) Real time traffic estimation for 75% percentage of missing data, (c) Traffic prediction for 50% of missing data, (d) Traffic prediction for outliers

4) *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 $\mathbf{o}_{i,\tau}$ as follows:

$$\mathbf{o}_{i,\tau} = \max(\mathbf{y}_{i,\tau-1}, \mathbf{y}_{i,\tau+1}) + c \mu_t. \quad (39)$$

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, $p_o = 5\%$ and $c = 0.75$, the plots in Fig. 8a illustrate the performance of the RVBSF algorithm. Similarly for 75% of missing entries, $p_o = 2\%$ the results are shown in Fig. 8b. 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. 8c and Fig. 8d.

	$c = 0.75$ $p_o = 5\%$	$c = 0.75$ $p_o = 2\%$	$c = 1.5$ $p_o = 2\%$
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 [13] GRASTA [9] and ROSETA [10] 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. 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

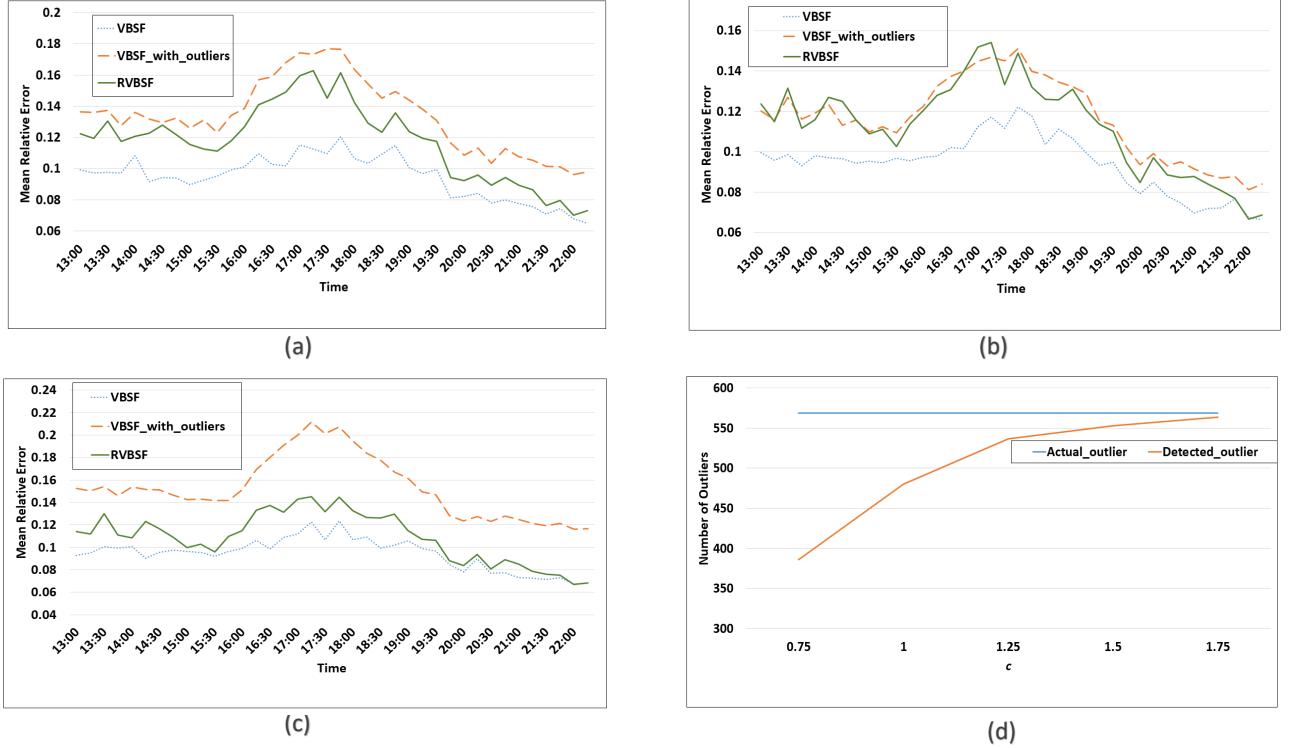


Fig. 8: Robust Bayesian subspace filtering for traffic data

(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

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

TABLE V: Performance Comparison for Robust Traffic Estimation

edges be be ignored by the model.

C. Electricity Load Prediction

We now discuss the performance of the VBSF algorithm on the electricity load data set [8]. Note that the electricity load data is also a time series data with the possibility of missing entries as well as temporal correlation between successive columns.

1) *Performance Index:* The performance of the VBSF method is compared with that of [14] using the metrics mean absolute error (MAE) and MRE, defined as:

$$\text{MAE} = \frac{1}{z} \sum_{k=1}^z \frac{\|\hat{\mathbf{y}}_k - \mathbf{y}_k\|_1}{l(\mathbf{y}_k)} \quad (40)$$

$$\text{MRE} = \frac{1}{z} \sum_{k=1}^z \frac{\|\hat{\mathbf{y}}_k - \mathbf{y}_k\|_2}{\|\mathbf{y}_k\|_2} \quad (41)$$

where \mathbf{y}_k and $\hat{\mathbf{y}}_k$ are the ground truth and estimated data for k^{th} column. We run the algorithm online on dates Jan. 1, 2012 to Jan. 1, 2015 resulting into 26,304 columns. In other words, the value of z is 26,304 for our simulations.

2) *Online Electricity Load Estimation and Prediction:* We run our algorithm for electricity data estimation and prediction. The results for real-time prediction are noted in table VI. It is noted as the percentage of observed data p increases, the real-time prediction accuracy improves.

	$p = 0.25\%$	$p = 0.5\%$	$p = 0.75\%$
MRE	0.1789	0.101	0.0987
MAE(kW)	96.95	66.67	53.95

TABLE VI: Electricity real time load prediction

Further, we predict the one-step ahead electricity load in Fig. 9. To analyze the performance of our algorithm we compare our results with OFMF and CKF [14]. The one-step ahead prediction performance of OFMF and CKF are provided in [14]. OFMF proposes a autoregressive model based optimization to predict the one-step ahead electricity load. We compare our three cases of p with the results shown in OFMF. It can be seen that our algorithm performs better than the OFMF for electricity load dataset.

V. CONCLUSION

This paper considers sequentially arriving multivariate data that resides in a time-varying low-dimensional subspace. The

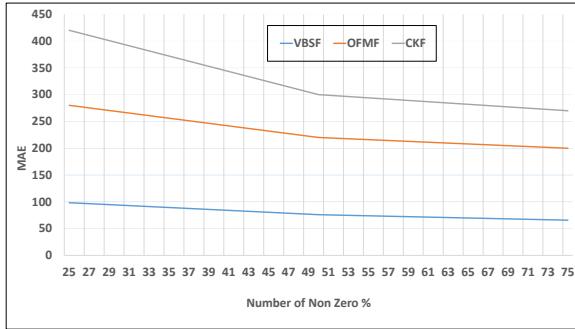


Fig. 9: One-step ahead electricity prediction

temporal evolution of the underlying low-rank subspace is characterized via a state-space model and low-complexity variational Bayesian subspace filtering algorithms are proposed for matrix completion and outlier removal tasks. 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. A similar result is observed on applying the VBSF algorithm on the twin tasks of imputation and prediction on the electricity data-set. Extensive simulations on both the data sets demonstrate that the suggested model and the accompanying algorithms seem to capture the temporal evolution of the data well as compared to the current state-of-the-art matrix completion and the online subspace estimation algorithms.

REFERENCES

- [1] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Found. Comp. Math.*, vol. 9, no. 6, pp. 717–772, 2009.
- [2] 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.
- [3] 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.
- [4] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Kourtoumbas, “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*, vol. 58, no. 3, pp. 11:1–11:37, 2011.
- [6] X. Ding, L. He, and L. Carin, “Bayesian robust principal component analysis,” *IEEE Trans. Image Process.*, 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 Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, 2013.
- [8] UCI, “Electricity load diagrams 2011-2014,” 2014. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014/>
- [9] J. He, L. Balzano, and J. Liu, “Online robust subspace tracking from partial information,” *arXiv preprint arXiv:1109.3827*, 2011.
- [10] H. Mansour and X. Jiang, “A robust online subspace estimation and tracking algorithm,” in *Proc. of the IEEE ICASSP*, Apr. 2015, pp. 4065–4069.
- [11] H.-F. Yu, N. Rao, and I. S. Dhillon, “Temporal regularized matrix factorization for high-dimensional time series prediction,” in *Proc. of NIPS*, Barcelona, Spain., Dec. 2016, pp. 847–855.
- [12] L. Xu and M. Davenport, “Dynamic matrix recovery from incomplete observations under an exact low-rank constraint,” in *Proc. of NIPS*, Barcelona, Spain., Dec. 2016, pp. 3585–3593.
- [13] H. Xu, C. Caramanis, and S. Sanghavi, “Robust PCA via outlier pursuit,” in *Proc. of NIPS*, Vancouver, Canada, Dec. 2010, pp. 2496–2504.
- [14] S. Gultekin and J. Paisley, “Online forecasting matrix factorization,” *IEEE Trans. Signal Process.*, vol. 67, no. 5, pp. 1223–1236, March. 2019.
- [15] 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.
- [16] ———, “Bilinear generalized approximate message passing Part II: Applications,” *IEEE Trans. Signal Process.*, vol. 62, no. 22, pp. 5854–5867, 2014.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] 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.
- [24] D. Mohan, “Moving around in Indian cities,” *Economic and Political Weekly*, vol. 48, no. 48, 2013.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] Z. Zhang, G. Liang, Y.-J. Dai, X.-u. Dong, and P.-x. Wang, “A short-term user load forecasting with missing data,” in *2018 International Conference on Mechanical, Electronic and Information Technology*, Apr. 2018, pp. 395–400.
- [31] 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.
- [32] 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.
- [33] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [34] M. J. Beal, “Variational algorithms for approximate Bayesian inference,” Ph.D. dissertation, University of London London, 2003.
- [35] 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.
- [36] M.-A. Sato, “Online model selection based on the variational Bayes,” *Neural Computation*, vol. 13, no. 7, pp. 1649–1681, 2001.
- [37] 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.