ONLINE VARIATIONAL BAYESIAN SUBSPACE FILTERING

Charul[†], Uttkarsha Bhatt[⋆] , Pravesh Biyani[⋆†] and Ketan Rajawat[⋆]

[†] IIIT-Delhi, India, [⋆] IIT-Kanpur, India

ABSTRACT

Many real world applications that suffer from missing data and outliers can be modeled in a matrix completion framework. In this paper, we consider low-rank matrices whose subspace evolves according to a state-space model and propose an online variational Bayesian formulation to learn the low rank components as well as the state-space model. Unlike the other matrix/tensor completion techniques, in our framework, the key algorithm parameters like rank and various noise power need not be fine-tuned and are learned automatically. We also propose a forward-backward algorithm that allows update to be carried out at low complexity manner. Simulations performed on the real world traffic data illustrates promising imputation as well as temporal prediction performance even in an online setup.

Index Terms— Matrix completion, Variational Bayesian, Traffic Estimation

1. INTRODUCTION

Real-world data collected from sensors are often incomplete as well as noisy with the possibility of outliers. The collected data is often in the form of matrix with missing entries that need to be inferred. Many approaches to impute the missing entries model the data as belonging to an underlying low-dimensional subspace that can subsequently be recovered via matrix completion [1–4], robust principal component analysis (PCA) [5, 6], or their tensor counterparts [7]. Such techniques approach the problem from a static perspective. Specifically, matrix or tensor completion is applied to data collected as a whole to impute the missing entries. In contrast, many real-world problems – for instance the traffic estimation and prediction problem discussed in this paper – are inherently dynamic, consisting of sequentially arriving data that must be dealt in an online manner taking care of an underlying low-rank subspace that also evolves over time.

This work considers the first low-rank robust subspace filtering approach for online data imputation and prediction. We propose a generative data model to do so and subsequently use variational Bayesian formalism to learn the parameters of the model. 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 columns of the 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) [3] 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. We also propose a low-complexity forward-backward algorithm that allows the updates to be carried out efficiently. A robust version of the VBSF algorithm is also developed for outlier removal and data cleansing but omitted due to the lack of space.

The online matrix completion framework discussed in this paper has many applications like traffic monitoring and prediction in the field of urban transportation, air quality monitoring and prediction as well as other traditional applications in machine learning. Traffic prediction has emerged as an important problem, thanks to the proliferation of various on-demand mobility solutions across the world with a variety of modes of transport like buses, cabs and even electric scooters. Moreover, the traffic data generally enjoys spatio-temporal correlation [8] that makes it amenable for an online matrix completion approach. We test the proposed algorithm on the real world traffic data collected over 200 square km. area within the city of New Delhi, India. The resulting matrix with more than 500 measurements per time instant is used for comparing the performance of the proposed algorithm with various state-of-the-art algorithms such as GROUSE [2] and LRTC [7]. The results show that modeling the evolution of the underlying subspace leads to accurate predictions and the low-complexity updates make the algorithm ideal for real-time applications.

1.1. Related Work

Variational Bayesian based approaches for matrix completion are well known [3, 4, 9–13]. 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 [9, 10, 13]. More recently, other approaches towards modeling the measured matrices have also been proposed [11]. The Variational Bayesian approaches mentioned do not use a state space model for the low dimension subspace [3, 4, 13]. In contrast to these, the state-space modeling in our work is inspired by [12], 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 [12] to incorporate low-rank structure.

Notation: 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 $[\mathbf{A}]_{i\cdot}^T$. 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 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 function is generically denoted by $p(\cdot)$. Given data $\mathbf{D}$, we denote $\hat{\mathbf{x}} := \mathbb{E}[\mathbf{x} \mid \mathbf{D}]$.

2. VARIATIONAL BAYESIAN SUBSPACE FILTERING

We consider a scenario where the data with the missing entries is arriving in a sequential manner. For the application of road traffic estimation and prediction considered in the paper, the 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. 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 $1 \leq \tau \leq t$
- *Prediction,* which yields $\{\hat{\mathbf{y}}_{t+\tau}\}_{\tau=1}^{T_p}$ where T_p is the prediction horizon.

We begin with detailing a generative model for the matrix $\mathbf{Y}$. The proposed model will not only capture the rank deficient nature of $\mathbf{Y}$ [14] but also the temporal correlation between successive columns of $\mathbf{Y}$. 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 noise distribution with unknown parameter, while various parameters are assigned different prior distributions. The Bayesian framework allows the use of Automatic Relevance Determination (ARD), wherein associating appropriate priors to the problem parameters leads to pruning of the redundant features [14]. This work uses pdfs from the exponential family that allow for tractable forms of the posterior pdf but are also flexible enough to adequately model the data.

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

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

for all $\tau \geq 1$, 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. The temporal evolution of $\mathbf{Y}$ is modeled such that columns of $\mathbf{B}$ follow first order autoregressive model:

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

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

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

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

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

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

where the precisions $\boldsymbol{\gamma}$ and $\boldsymbol{\upsilon}$ are problem parameters. As Subspace Bayesian Learning, many of the precisions $\{\gamma_i, \upsilon_i\}$ will generally assume large values during the inference, effectively removing the corresponding columns from $\mathbf{A}$ and $\mathbf{J}$ respectively. Finally, the three precision variables are selected to have have non-informative Jeffrey's priors

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

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

$$p(\mathbf{y}_\Omega, \mathcal{H}) = p(\mathbf{y}_\Omega | \mathbf{A}, \mathbf{B}, \beta) p(\mathbf{A}|\boldsymbol{\gamma}) p(\mathbf{B}|\mathbf{J}) p(\mathbf{J}|\boldsymbol{\upsilon}) p(\beta) p(\boldsymbol{\upsilon}) p(\boldsymbol{\gamma}) \tag{7}$$

2.1. Variational Bayesian Inference

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

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

Under the mean-field approximation, the variational lower bound can be maximized via coordinate ascent iterations [15]. Indeed, thanks to the choice of conjugate priors for the parameters, it can be shown that the individual factors in (8) take the following forms:

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

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

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

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

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

$$q_{\upsilon_i}(\upsilon_i) = \text{Ga}(\upsilon_i; a_i^\upsilon, b_i^\upsilon) \tag{9f}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

where $\Omega_i' := \{\tau \mid i \in \Omega_\tau\}$. Observe from the updates that the rows of $\mathbf{J}$ are independent identically distributed under the mean field approximation. The update for $\boldsymbol{\mu}^{\mathbf{B}}$ can be written as

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

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

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

It is remarked that although the $rt \times rt$ matrix $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1}$ is block-tridiagonal, the matrix $\boldsymbol{\Xi}^{\mathbf{B}}$ is dense, and direct inversion would be prohibitively costly. Moreover, the classical Rauch-Tung-Striebel (RTS) smoother cannot be directly applied as the evaluation since evaluating the conditional expectations under $q(\mathbf{B})$ is difficult and not amenable to the Matrix Inversion Lemma [16]. Interestingly, observe that the updates in (14) and (15) depend only on diagonal and super-diagonal blocks of $\boldsymbol{\Xi}^{\mathbf{B}}$, namely $\boldsymbol{\Xi}_{\tau,\tau}^{\mathbf{B}}$ and $\boldsymbol{\Xi}_{\tau,\tau-1}^{\mathbf{B}}$, respectively. The next subsection details a low-complexity algorithm for carrying out the updates for these blocks as well as for $\boldsymbol{\mu}^{\mathbf{B}}$.

2.2. Low-complexity updates via LDL-decomposition

Thanks to the block-tridiagonal structure of $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1}$, it is possible to use the LDL decomposition to carry out the updates in an efficient manner. Decomposing $[\boldsymbol{\Xi}^{\mathbf{B}}]^{-1} = \mathbf{L}\mathbf{D}\mathbf{L}^T$, the key idea is that left multiplication with $\boldsymbol{\Xi}^{\mathbf{B}}$ is equivalent to left multiplication with $\mathbf{L}^{-T}\mathbf{D}^{-1}\mathbf{L}^{-1}$. Towards this end, we utilize the algorithm from [17], that comprises of two phases: the forward pass that carries out the multiplication with $\mathbf{D}^{-1}\mathbf{L}^{-1}$ and the backward pass that implements the multiplication with $\mathbf{L}^{-T}$. Let us define for $2 \leq \tau \leq t$,

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Overall, the different parameters are updated cyclically until convergence for each $t = 1, 2, \ldots$.

EM algorithm is used in the model which 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 $\mathcal{H}_p := \{\boldsymbol{\upsilon}, \boldsymbol{\gamma}, \beta\}$.

3. 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. To evaluate the VBSF algorithm, we use the partial road network of the city of New Delhi with an area of 200 square kms consisting of $m = 519$ edges. To estimate the model parameters and for testing purposes, speed data was collected using the Google Maps API for nearly 3 months across all the 519 edges. Taking advantage of the slow varying nature of the speed in the network edges, we sample the traffic data at the rate of one sample every $t_s = 15$ minutes. Note that our algorithm is agnostic of the sampling rate and would work for higher sampling rates as well.

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

Algorithm 1: Variational Bayesian Subspace Filtering

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

for the same day. Finally, we use the traffic data for the last 3 months to test the VBSF algorithm, wherein we use the first month data is used to estimate the priors, while the subsequent months data is used to evaluate the accuracy of the estimation and prediction tasks.

3.1. 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 any time instance τ, the MRE denoted by MRE_τ is defined as:

$$\text{MRE}_\tau = \frac{1}{z} \sum_{k=1}^{z} \frac{\| \hat{y}_{\tau,k} - y_{\tau,k} \|_2}{\| y_{\tau,k} \|_2}. \tag{23}$$

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

3.2. 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. The MRE values for real time traffic estimation using VBSF is shown in Table 1 . We compare our algorithm with other methods that potentially solve the current traffic estimation problem in the missing data scenario. We use low rank tensor completion (LRTC) [7], Grassmannian Rank-One Update Subspace Estimation (GROUSE) [2] algorithm and finally the historic mean for comparison purposes. The historic mean is simply the mean of edge speed values at a given time instance calculated using the data. Table 1 presents the overall results. 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 2.

	$p = 0.25$ 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 1: 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 2: Comparison of running time for different algorithms

3.3. 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 15 and 30 mins ahead traffic. The performance of the proposed VBSF algorithm is compared with that of LRTC in Table 3. The VBSF performs better than the LRTC.

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

Table 3: Performance comparison for traffic prediction

4. CONCLUSION

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. 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. Moreover, one can also predict the future traffic which in turn can be used to increase the reliability of the public transport even in places with multiple modes of transport.

5. REFERENCES

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

[2] Laura Balzano, Robert Nowak, and Benjamin Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. of the Annual Allerton Conference on Communication,Control, and Computing*. IEEE, 2010, pp. 704–711.

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

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

[5] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, "Robust principal component analysis?," *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 11, 2011.

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

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

[8] Muhammad Tayyab Asif, Nikola Mitrovic, Justin Dauwels, and Patrick Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1816–1825, 2016.

[9] Jason T Parker, Philip Schniter, and Volkan Cevher, "Bilinear generalized approximate message passing part I: Derivation," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5839–5853, 2014.

[10] Jason T Parker, Philip Schniter, and Volkan Cevher, "Bilinear generalized approximate message passing part II: Applications," *IEEE Transactions on Signal Processing*, vol. 62, no. 22, pp. 5854–5867, 2014.

[11] Bo Xin, Yizhou Wang, Wen Gao, and David Wipf, "Exploring algorithmic limits of matrix rank minimization under affine constraints," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 4960–4974, 2016.

[12] Jaakko 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.

[13] Linxiao Yang, Jun Fang, Huiping Duan, Hongbin Li, and Bing Zeng, "Fast low-rank bayesian matrix completion with hierarchical gaussian prior models," *IEEE Transactions on Signal Processing*, vol. 66, no. 11, pp. 2804–2817, 2018.

[14] S Derin Babacan, Martin Luessi, Rafael Molina, and Aggelos K Katsaggelos, "Sparse bayesian methods for low-rank matrix estimation," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3964–3977, 2012.

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

[16] Matthew James Beal et al., *Variational algorithms for approximate Bayesian inference*, Ph.D. thesis, University of London London, 2003.

[17] Jaakko 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.