Bayesian Temporal Factorization for Multidimensional Time Series Prediction

Abstract

Large-scale and multidimensional spatiotemporal data sets are becoming ubiquitous in many real-world applications such as monitoring traffic and air quality. Making predictions on these multidimensional time series has become a critical challenge due to not only the large-scale and high-dimensional nature but also the considerable amount of missing data. In this paper, we propose a Bayesian temporal factorization (BTF) framework for modeling multidimensional time series—and in particular spatiotemporal data—in the presence of missing data. By integrating low-rank matrix/tensor factorization and autoregressive (AR) process into a single probabilistic graphical model, this framework can effectively perform predictions without imputing those missing values. We develop efficient Gibbs sampling algorithms for model inference and test the proposed BTF framework on several real-world spatiotemporal data sets for both missing data imputation and short-term rolling prediction tasks. The numerical experiments demonstrate the superiority of the BTF approach over many state-of-the-art techniques.

1 Introduction

With recent advances in sensing technologies, multidimensional time series data—in particular spatiotemporal data—are collected on a continuous basis from various sensors. Given the complex temporal dynamics in these data sets, making reliable and efficient predictions has been a long-standing research question. The prediction of these time series, such as forecasting urban traffic and air quality, serves as a fundamental input to a variety of real-world applications depending on it. For example, predicting the demand and states of urban traffic is essential to a wide range of intelligent transportation systems (ITS) applications, such trip planning, travel time estimation, route planning, traffic signal control, to name but a few [Li and Shahabi, 2018].

There exist a large body of literature on time series prediction. However, two emerging issues in modern sensing technologies are constantly challenging existing time series modeling frameworks. On the one hand, most existing models essentially require complete time series data as input, while in practice the missing data problem is almost inevitable due to various factors such as hardware/software failure, human error, and network communication problem. Therefore, a critical question is how to perform reliable prediction in the presence of missing data [Anava *et al.*, 2015; Saad and Mansinghka, 2018]. On the other hand, the high-dimensional property makes spatiotemporal data very difficult to model due to the higher-order correlations/dependencies across different dimensions [Jing *et al.*, 2018]. For example, mobility demand for different types of travelers using different modes can be modeled as a 5-d (origin zone$\times$desination zone$\times$travel mode [e.g., car, transit, and bike]$\times$traveler type [e.g., child, adult, and senior]$\times$time interval) time series tensor and all the dimensions interact with each other.

Several notable approaches have been proposed and tested to overcome these issues. In a recent work [Yu *et al.*, 2016], the authors proposed a novel Temporal Regularized Matrix Factorization (TRMF) framework for modeling multivariate time series with missing data by introducing an autoregressive (AR) regularizer on the temporal latent factor in the underlying matrix factorization. An efficient and scalable estimation algorithm based on alternating least square (ALS) was also proposed to process large-scale data sets. This work is further extended in [Takeuchi *et al.*, 2017], which transformed multivariate time series into spatiotemporal tensors and innovatively added a new graph Laplacian regularizer on the spatial factor in tensor factorization. Thus, this model is able to perform prediction on the spatial dimension for unknown locations. These models have shown superior performance from numerical experiments on real-world data sets; however, they have two main drawbacks: (1) they require careful tuning of parameters/hyperparameters to ensure more accuracy and avoid overfitting, and (2) the tuning procedure has been done for each specific study/task/data set since there are no universal solutions. In the meanwhile, [Xiong *et al.*, 2010] proposed a new Bayesian Probabilistic Tensor Factorization (BPTF) framework, which extends Bayesian Matrix Factorization for tensor structure and imposes a Markovian assumption on the temporal latent factor. The fully Bayesian treatment can effectively address the aforementioned estimation problems. However, the simple Markovian assumption has limited capability in capturing the dependencies in complex time series data.

In this paper, we propose a new Bayesian Temporal Factorization (BTF) framework which can effectively handle both the missing data problem and high-dimensional property. Based on existing works of [Yu *et al.*, 2016] and [Xiong *et al.*, 2010], and [Jing *et al.*, 2018], this BTF framework combines low-rank matrix factorization (MF) and tensor factorization (TF) with AR models into a single probabilistic graphical model. We provide a fully Bayesian treatment on this model by placing conjugate prior over all parameters and hyperparameters, and further design efficient Markov chain Monte Carlo (MCMC) algorithms for model inference. Our proposed approach has two major advantages: 1) compared with traditional two-stage approaches (first imputing missing values and then performing prediction), the probabilistic model can solve both the missing data imputation and future value prediction problem simultaneously without introducing potential biases, and 2) the Bayesian approach provides an elegant way to avoid parameter tuning and overfitting issues.

2 Problem Description

We assume a spatiotemporal setting for multidimensional time series throughout this paper. Modern spatiotemporal data sets collected from sensor networks are often organized as time series matrices. For example, we can denote by $Y \in \mathbb{R}^{m\times f}$ a time series matrix (multivariate time series) collected from m locations in f time stamps, with each row

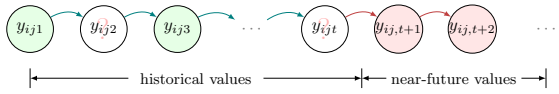
$$\boldsymbol{y}_i = (y_{i1}, y_{i2}, ..., y_{i,t-1}, y_{it}, y_{i,t+1}, ..., y_{if})^\top$$

corresponding to the time series collected at location i. As another example, the time-varying origin-destination travel demand can be organized as a third-order time series tensor $\mathcal{Y} \in \mathbb{R}^{m\times n\times f}$ with m origin and n destination zones ($m = n$ in most cases), with each time series

$$\boldsymbol{y}_{ij} = (y_{ij1}, y_{ij2}, ..., y_{ij,t-1}, y_{ijt}, y_{ij,t+1}, ..., y_{ijf})^\top$$

showing the flow time series from i to j. This formulation essentially can be further extended to even higher-order tensors given the dimension of recorded information.

(a) Multivariate time series as a matrix.



(b) Multivariate time series as a third-order tensor.

Figure 1: Illustration of high-order time series and the prediction problem with missing values (green: observed data; white: missing data; red: prediction).

As mentioned, making accurate predictions on incomplete time series is a very challenging research question, while missing data problem is almost inevitable in any real-world applications. Figure 1 illustrates the prediction problem for incomplete time series data. Here we use $(i,t) \in \Omega$ and $(i,j,t) \in \Omega$ to index the observed entries in matrix Y and tensor $\mathcal{Y}$, respectively. Given that missing data imputation has been studied extensively, a natural solution is to adopt a two-stage approach: first applying imputation algorithms to fill in those missing values and then performing predictions based on the complete time series. This two-stage approach is widely applied in many real-world practices [Che *et al.*, 2018]; however, by applying imputation first, it actually produces a new layer of errors resulted from the imputation algorithm which we should avoid.

3 Bayesian Temporal Matrix Factorization

3.1 Model Specification

Given a partially observed spatiotemporal matrix $Y \in \mathbb{R}^{m\times f}$, one can factorize it into two factor matrices $W \in \mathbb{R}^{m\times r}$ (spatial) and $X \in \mathbb{R}^{f\times r}$ (temporal) following general MF model:

$$Y \approx WX^T, \tag{1}$$

and element-wise, we have

$$y_{it} \approx \boldsymbol{w}_i^T \boldsymbol{x}_t, \quad \forall (i,t), \tag{2}$$

where column vectors $\boldsymbol{w}_i$ and $\boldsymbol{x}_t$ refer to the i-th row of W and the t-th row of X, respectively.

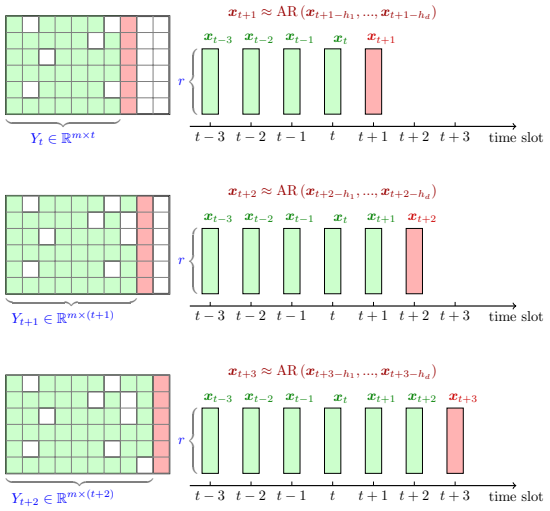


Figure 2: A graphical illustration for the rolling prediction strategy using temporal matrix factorization (green: observed data; white: missing data; red: prediction).

To capture the temporal dynamics in matrix Y, the notable work of [Yu *et al.*, 2016] proposed the TRMF model by introducing an AR regularizer into the MF framework. Given observed Y_t and a trained model, one can first predict $\boldsymbol{x}_{t+1}$ on the latent temporal factor matrix X and then estimated $y_{i,t+1} \approx \boldsymbol{w}_i^T \boldsymbol{x}_{t+1}$ for the original prediction task. Figure 2 illustrates a one-step rolling prediction scheme based on this idea. By performing prediction on X instead of on Y, TRMF model can efficiently and effectively solve the prediction problem for incomplete time series. However, in practice

TRMF requires users to tune parameters/hyperparameters carefully to ensure model accuracy and avoid overfitting issues. Moreover, since there exist no universal/automatic solutions, this tuning procedure has to be done for each particular application (i.e., input data set).

To address these issues, in the following we propose the Bayesian Temporal Matrix Factorization (BTMF) model—as the Bayesian counterpart of TRMF—and extend it to multivariate time series tensors. Although there have been some previous works on modeling temporal dynamics in the Bayesian factorization setting, they essentially impose first-order Markovian/state-space assumptions on the temporal latent factor [Xiong *et al.*, 2010; Charlin *et al.*, 2015]. These model may work well in discovering general temporal trends, but the simple assumption also limits its capability in capturing complex time series dynamics.

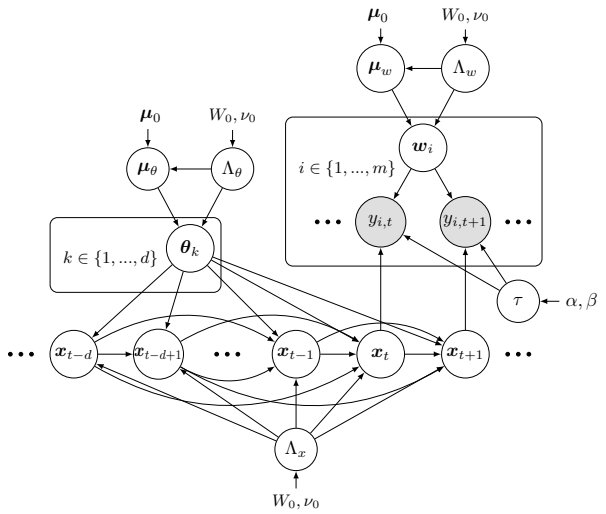


Figure 3: An overview graphical model of BTMF (time lag set: $\{1, 2, ..., d\}$). The shaded nodes ($y_{i,t}$) are the observed data in Ω.

To better model temporal dynamics, we introduce AR process in modeling X. Figure 3 shows the whole graphical representation of BTMF. Note that this model is solely built on observed data in Ω and thus it can learn from partially observed data. We next introduce each component in this graphical model in detail.

Following the general Bayesian probabilistic MF/TF models [Salakhutdinov and Mnih, 2008; Xiong *et al.*, 2010], we assume that each observed entry in Y follows a Gaussian distribution with precision τ:

$$y_{it} \sim \mathcal{N}\left(\boldsymbol{w}_i^T \boldsymbol{x}_t, \tau^{-1}\right), \quad (i,t) \in \Omega. \tag{3}$$

This assumption is equivalent to the $\ell 2$-loss in general MF. On the spatial dimension, we only use a simple Gaussian factor matrix without imposing any dependencies explicitly:

$$\boldsymbol{w}_i \sim \mathcal{N}\left(\boldsymbol{\mu}_w, \Lambda_w^{-1}\right), \tag{4}$$

and we place a conjugate Gaussian-Wishart prior on the mean vector and the precision matrix:

$$\boldsymbol{\mu}_w | \Lambda_w \sim \mathcal{N}\left(\boldsymbol{\mu}_0, (\beta_0 \Lambda_w)^{-1}\right), \Lambda_w \sim \mathcal{W}\left(W_0, \nu_0\right).$$

In modeling the time-evolving dynamics in X, we assume that the vectors $\boldsymbol{x}_t$ ($\forall t \in \{1, 2, ..., f\}$) follow an AR model:

$$\boldsymbol{x}_t \sim \mathcal{N}\left(\tilde{\boldsymbol{x}}_t, \Lambda_x^{-1}\right), \tag{5}$$

with the mean vector defined as:

$$\tilde{\boldsymbol{x}}_t = \begin{cases} \boldsymbol{0} & \text{if } t \in \{1, 2, ..., h_d\} \\ \sum_{k=1}^{d} \boldsymbol{\theta}_k \circledast \boldsymbol{x}_{t-h_k} & \text{otherwise} \end{cases}, \tag{6}$$

where $\{h_1, \ldots, h_k, \ldots, h_d\}$ is a set of time lags, $\boldsymbol{\theta}_k \in \mathbb{R}^r$ is a vector of regression coefficients for lag k, and the symbol $\circledast$ denotes Hadamard product. For simplicity, we define $A_k = \text{diag}\left(\boldsymbol{\theta}_k\right)$. Since the mean vector has been defined, we only need to place a Wishart prior on the precision matrix $\Lambda_x \sim \mathcal{W}\left(W_0, \nu_0\right)$. We assume $\boldsymbol{\theta}_k$ also follows a Gaussian distribution $\boldsymbol{\theta}_k \sim \mathcal{N}(\boldsymbol{\mu}_\theta, \Lambda_\theta^{-1})$ and place conjugate Gaussian-Wishart prior on the hyper-parameters:

$$\boldsymbol{\mu}_\theta | \Lambda_\theta \sim \mathcal{N}\left(\boldsymbol{\mu}_0, (\beta_0 \Lambda_\theta)^{-1}\right), \quad \Lambda_\theta \sim \mathcal{W}\left(W_0, \nu_0\right).$$

For the only remaining parameter τ, we place a Gamma prior $\tau \sim \text{Gamma}\left(\alpha, \beta\right)$ where α and β are the shape and rate parameters, respectively. The above specifies the full generative process of the proposed BTMF.

3.2 Model Inference

Given the complex structure of BTMF, it is intractable to write down the posterior distribution. Here we rely on the MCMC technique for model inference. In detail, we introduce a Gibbs sampling algorithm by deriving the full conditional distributions for each parameter and hyperparameter. The use of conjugate priors in Figure 3 allows us to derive all conditional distributions analytically.

Sampling $(\boldsymbol{\mu}_w, \Lambda_w)$: The conditional distribution is given by a Gaussian-Wishart:

$$p\left(\boldsymbol{\mu}_w, \Lambda_w | -\right) = \mathcal{N}(\boldsymbol{\mu}_w^*, \left(\left(\beta_0 + m\right) \Lambda_w\right)^{-1}) \times \mathcal{W}\left(W_w^*, \nu_w^*\right),$$

where

$$\boldsymbol{\mu}_w^* = \frac{1}{\beta_0 + m}\left(\beta_0 \boldsymbol{\mu}_0 + d\bar{\boldsymbol{w}}\right), \quad \nu_w^* = \nu_0 + m,$$

$$\left(W_w^*\right)^{-1} = W_0^{-1} + m S_w + \frac{\beta_0 m}{\beta_0 + m}\left(\bar{\boldsymbol{w}} - \boldsymbol{\mu}_0\right)\left(\bar{\boldsymbol{w}} - \boldsymbol{\mu}_0\right)^T,$$

$$\bar{\boldsymbol{w}} = \frac{1}{m}\sum\nolimits_{i=1}^{m} \boldsymbol{w}_i, \; S_w = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left(\boldsymbol{w}_i - \bar{\boldsymbol{w}}\right)\left(\boldsymbol{w}_i - \bar{\boldsymbol{w}}\right)^T.$$

Sampling Λ_x: Given the Wishart prior, the corresponding conditional distribution is $p\left(\Lambda_x | -\right) = \mathcal{W}\left(W_x^*, \nu_x^*\right)$ and its parameters are given by:

$$\left(W_x^*\right)^{-1} = W_0^{-1} + \sum\nolimits_{t=1}^{f}\left(\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\right)\left(\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t\right)^T, \; \nu_x^* = \nu_0 + f.$$

Sampling $(\boldsymbol{\mu}_\theta, \Lambda_\theta)$: The conditional posterior distribution $p\left(\boldsymbol{\mu}_\theta, \Lambda_\theta | -\right)$ is a Gaussian-Wishart and it is of exactly the same form as $p\left(\boldsymbol{\mu}_w, \Lambda_w | -\right)$.

Sampling spatial factor $\boldsymbol{w}_i$: The conditional posterior distribution $p\left(\boldsymbol{w}_i \mid \boldsymbol{y}_i, X, \tau, \boldsymbol{\mu}_w, \Lambda_w\right)$ is a Gaussian distribution. Thus, we can sample $\boldsymbol{w}_i | - \sim \mathcal{N}(\boldsymbol{\mu}_w^*, \left(\Lambda_w^*\right)^{-1})$ with

$$\Lambda_w^* = \tau \sum\nolimits_{t:(i,t)\in\Omega} \boldsymbol{x}_t \boldsymbol{x}_t^T + \Lambda_w,$$

$$\boldsymbol{\mu}_w^* = \left(\Lambda_w^*\right)^{-1}\left(\tau \sum\nolimits_{t:(i,t)\in\Omega} \boldsymbol{x}_t y_{it} + \Lambda_w \boldsymbol{\mu}_w\right).$$

Sampling temporal factor $\boldsymbol{x}_t$: Given the AR process, the conditional distribution of $\boldsymbol{x}_t$ is also a Gaussian. However, for a particular time lag set, we need to define different updating rules for $1 \leq t \leq f - h_1$ and $f - h_1 < t \leq f$. Overall, the conditional distribution can be written as $p\left(\boldsymbol{x}_t|-\right) = \mathcal{N}(\boldsymbol{\mu}_t^*, (\Lambda_t^*)^{-1})$ with

$$\begin{aligned}\Lambda_t^* =& \tau \sum\nolimits_{i:(i,t)\in\Omega} \boldsymbol{w}_i \boldsymbol{w}_i^T + M_t + \Lambda_x,\\ \boldsymbol{\mu}_t^* =& (\Lambda_t^*)^{-1}\left(\tau \sum\nolimits_{i:(i,t)\in\Omega} \boldsymbol{w}_i y_{it} + N_t + Q_t\right),\end{aligned} \tag{7}$$

where M_t and N_t are two auxiliary variables. In general cases where $1 \leq t \leq f - h_1$, we define M_t and N_t as follows:

$$\begin{aligned}M_t &= \sum\nolimits_{k=1, h_d < t+h_k \leq f}^{d} A_k \Lambda_x A_k,\\ N_t &= \sum\nolimits_{k=1, h_d < t+h_k \leq f}^{d} A_k \Lambda_x \boldsymbol{\psi}_{t+h_k},\\ \boldsymbol{\psi}_{t+h_k} &= \boldsymbol{x}_{t+h_k} - \sum\nolimits_{l=1, l\neq k}^{d} A_l \boldsymbol{x}_{t+h_k-h_l}.\end{aligned}$$

Otherwise, we define $M_t = 0$ and $N_t = 0$.

The variable Q_t in Equ (7) is defined as

$$Q_t = \begin{cases}\boldsymbol{0} & \text{if } t \in \{1, 2, ..., h_d\}\\ \Lambda_x \sum_{l=1}^{d} A_l \boldsymbol{x}_{t-h_l} & \text{otherwise}\end{cases}.$$

Sampling coefficient $\boldsymbol{\theta}_k$: Parameter $\boldsymbol{\theta}_k$ is a vector of AR coefficients. The full conditional distribution is also a Gaussian distribution. Similar to other parameters, we rewrite $p\left(\boldsymbol{\theta}_k|-\right) = \mathcal{N}(\boldsymbol{\mu}_k^*, (\Lambda_k^*)^{-1})$ and define the mean vector and precision matrix correspondingly as:

$$\begin{aligned}\Lambda_k^* &= \sum\nolimits_{t=h_d+1}^{f} B_{t-h_k} \Lambda_x B_{t-h_k} + \Lambda_\theta,\\ \boldsymbol{\mu}_k^* &= (\Lambda_k^*)^{-1}\left(\sum\nolimits_{t=h_d+1}^{f} B_{t-h_k} \Lambda_x \boldsymbol{\pi}_t^{(k)} + \Lambda_\theta \boldsymbol{\mu}_\theta\right),\end{aligned}$$

where $\boldsymbol{\pi}_t^{(k)} = \boldsymbol{x}_t - \sum_{l=1, l\neq k}^{d} \text{diag}\left(\boldsymbol{x}_{t-h_l}\right) \boldsymbol{\theta}_l$ and $B_{t-h_k} = \text{diag}\left(\boldsymbol{x}_{t-h_k}\right)$.

Sampling precision τ: Based on the Gamma prior, the conditional distribution of τ is also a Gamma distribution, i.e., $\tau|- \sim \text{Gamma}\left(\alpha^*, \beta^*\right)$ with $\alpha^* = \frac{1}{2}|\Omega| + \alpha$ and $\beta^* = \frac{1}{2}\sum_{(i,t)\in\Omega}\left(y_{it} - \boldsymbol{w}_i^T \boldsymbol{x}_t\right)^2 + \beta$.

4 Bayesian Temporal Tensor Factorization

In this section, we extend BTMF for multidimensional (order>2) time series tensors and use a third-order tensor $\mathcal{Y} \in \mathbb{R}^{m\times n\times f}$ as an example throughout the following. For factorization techniques, we employ CANDECOMP/PARAFAC (CP) decomposition [Kolda and Bader, 2009], which approximates $\mathcal{Y}$ by the sum of r rank-one tensors:

$$\mathcal{Y} \approx \sum\nolimits_{s=1}^{r} \boldsymbol{u}_s \circ \boldsymbol{v}_s \circ \boldsymbol{x}_s, \tag{8}$$

where $\boldsymbol{u}_s \in \mathbb{R}^m$, $\boldsymbol{v}_s \in \mathbb{R}^n$, and $\boldsymbol{x}_s \in \mathbb{R}^f$ are the s-th column of factor matrices $U \in \mathbb{R}^{m\times r}$, $V \in \mathbb{R}^{n\times r}$, and $X \in \mathbb{R}^{f\times r}$, respectively (see Figure 4). The symbol $\circ$ denotes vector outer product. Essentially, this model can be considered a high-order extension of Equ (1). Element-wise, we have

$$y_{ijt} \approx \sum\nolimits_{s=1}^{r} u_{is} v_{js} x_{ts}, \quad \forall\left(i, j, t\right). \tag{9}$$

The CP decomposition provides us a natural way to extend BTMF to tensors by assuming:

$$y_{ijt} \sim \mathcal{N}(\sum\nolimits_{s=1}^{r} u_{is} v_{js} x_{ts}, \tau^{-1}), \quad (i, j, t) \in \Omega. \tag{10}$$

Following the same idea as BTMF, we define the generative process of Bayesian Temporal Tensor Factorization (BTTF) as follows:

$$\begin{aligned}&\boldsymbol{u}_i \sim \mathcal{N}\left(\boldsymbol{\mu}_u, \Lambda_u^{-1}\right), \quad \boldsymbol{v}_j \sim \mathcal{N}\left(\boldsymbol{\mu}_v, \Lambda_v^{-1}\right),\\ &\boldsymbol{x}_t \sim \mathcal{N}\left(\tilde{\boldsymbol{x}}_t, \Lambda_x^{-1}\right), \quad \tau \sim \text{Gamma}\left(\alpha, \beta\right).\end{aligned}$$

In the above model, we may consider both U and V as spatial factor matrices, while in practice they may refer to any features in which dependencies are not explicitly encoded (e.g., type of travelers in [Sun and Axhausen, 2016] and type of sensors in [Takeuchi *et al.*, 2017]).

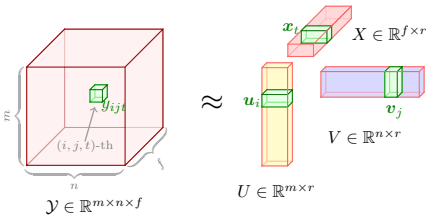


Figure 4: A graphical illustration of CP factorization.

5 Experiments

In this section we apply BTMF and BTTF on several real-world spatiotemporal data sets for both imputation and prediction tasks, and evaluate the effectiveness of these two models against recent state-of-the-art approaches. We use the mean absolute percentage error (MAPE) and root mean square error (RMSE) as evaluation metrics.

5.1 Testing BTMF

Data set (G): Guangzhou traffic speed[1]. This data set registered traffic speed data from 214 road segments over two months (61 days from August 1 to September 30 in 2016) with a 10-minute resolution (144 time intervals per day) in Guangzhou, China. We organize the raw data set into a time series matrix of 214×8784.

Data set (B): Birmingham parking [2]. This data set registered occupancy (i.e., number of parked vehicles) of 30 car parks in Birmingham City for every half an hour between 8:00 and 17:00 over more than two months (77 days from October 4, 2016 to December 19, 2016). The size of this time series matrix is 30×1386 with 18 time intervals per day and there are 14.89% missing values after data processing. In particular, the data is completely missing on four days (October 20/21 and December 6/7).

[1] https://doi.org/10.5281/zenodo.1205229

[2] https://archive.ics.uci.edu/ml/datasets/Parking+Birmingham

Baselines. We consider the following baseline models for missing data imputation: 1) TRMF [Yu *et al.*, 2016]; 2) Bayesian CP factorization using variational inference (BCPF) [Zhao *et al.*, 2015]; 3) Bayesian Gaussian CP decomposition (BGCP) [Chen *et al.*, 2019], which is a high-order extension of Bayesian Probabilistic Matrix Factorization (BPMF) by [Salakhutdinov and Mnih, 2008]; 4) HaLRTC: High-accuracy Low-Rank Tensor Completion [Liu *et al.*, 2013]; 5) MissForest: A non-parametric imputation method based on random forests [Stekhoven and Bühlmann, 2012]; and 6) VAE: Variational Autoencoder [McCoy *et al.*, 2018].

For the prediction tasks, we compare BTMF with TRMF and a family of k-nearest neighbors (kNN)-based models: kNN+ARIMA (Autoregressive Integrated Moving Average), kNN+GRU (Gated Recurrent Units), and kNN+LSTM (Long Short-Term Memory) [Che *et al.*, 2018]. The kNN-based models also adopt a two-stage approach: first impute missing values in the matrix using kNN and then predict future values based on the complete time series by using ARIMA, GRU and LSTM. In addition, we also implement a two-stage BPMF-AR model, which first learns latent factor X using BPMF and then predicts future values of X using AR.

Experiment setup. We consider two common missing data scenarios—random missing (**RM**) and non-random missing (**NM**). For RM, we simply remove a certain amount of observed entries in the matrix randomly and use these entries as ground truth to evaluate MAPE and RMSE. The percentages of missing are set to 20% and 40% for data set (G) and 10% and 30% for (B), respectively. For NM, we apply a fiber/block missing experiment by randomly choosing (e.g., 20%) location$\times$day combinations and removing the all observations in each combination. Again, the removed entries are used for evaluation. The NM scenario corresponds to cases where sensors have a certain probability to fail on each day. For tensor-based baseline models, we use a third-order (location$\times$day$\times$time slot) structure as input. Other models are tested with the time series matrix representation (location$\times$time series). Since MissForest is very time-consuming for large-scale data sets, we create a specific imputation model for each week in data set (G). Due to the high missing rate, MissForest cannot work on data set (B). For BTMF and TRMF, the time lags are set as $\{1, 2, 144\}$ and $\{1, 2, 18\}$, respectively, for both data sets (G) and (B) .

For the prediction tasks, we apply a short-term rolling-based prediction experiment (see Figure 2 for an illustration). The evaluation is done by making rolling predictions over the last five days (i.e., 5×144 time slots) for data set (G) and the last seven days (i.e., 7×18 time slots) for data set (B). The same time lag sets as the imputation experiment are applied. Note that BTMF, BPMF-AR and TRMF can perform prediction without imputing missing values, while the kNN family perform imputation first before making prediction.

Results and analysis. We evaluate the proposed BTMF model on both imputation and prediction tasks. Our first experiment is for missing data imputation. Although BTMF is design for making prediction with missing data, it can be also used for imputing missing values as an additional feature. The TRMF and BGCP baseline models are of particular interest to us. In general, TRMF can be considered the non-probabilistic version of BTMF and it requires us to define regularization parameters carefully. The BGCP is tensor-based imputation method. Although this model can capture high-order correlations by taking advantage from the tensor structure, it cannot model temporal dynamics explicitly using a simple multivariate Gaussian factors. Table 1 shows the imputation performance of BTMF and other baselines for data sets (G) and (B). The results in all experiments are given by 'MAPE/RMSE'. As can be seen, the proposed BTMF clearly outperforms the TRMF and BGCP in most experiments. The results suggest that BTMF inherits the advantages of both TRMF and BGCP: it not only provides a flexible and automatic Bayesian inference technique for model estimation, but also offers superior imputation performance by integrating temporal dynamics into matrix factorization. In addition, BTMF shows superior performance even compared with state-of-the-art missing data imputation models.

We conduct the experiment for making rolling predictions (see Figure 2) on the two data sets and Table 2 shows the performance of BTMF and corresponding baselines for prediction tasks. The proposed BTMF shows clear superiority over other baseline models. The comparison with TRMF further clarifies the value of the fully Bayesian treatment and temporal modeling. In the meanwhile, BTMF also outperforms all two-stage models, including BPMF-AR, kNN+ARIMA, kNN+GRU, and kNN+LSTM. Figure 5 gives the visualization on the prediction results achieved by BTMF on Birmingham parking data with missing values.

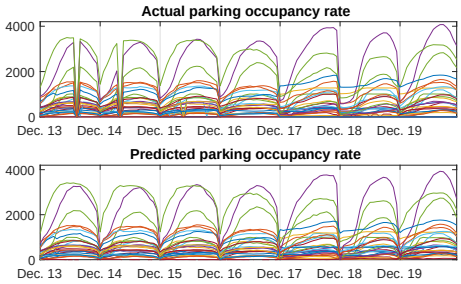


Figure 5: Predicted occupancy of BTMF at 10% NM missing scenario vs. actual observations. Each curse corresponds to a car park.

5.2 Testing BTTF

Data set (N): NYC taxi[3]. This data set registers trip information (pick-up/drop-off locations and start time) for different types of taxi trips. For the experiment, we choose the trips collected during May and June 2018 (61 days) and organize the raw data into a third-order (pick-up zone$\times$drop-off zone$\times$time slot) tensor. We define in total 30 pick-up/drop-off zones and the temporal resolution for aggregating trips is selected as 1h. The size of this spatiotemporal tensor is $30 \times 30 \times 1464$.

Baselines. We select the Temporal Collaborative Filtering (TCF) technique as a benchmark model [Xiong *et al.*, 2010].

[3]http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

	BTMF	TRMF	BCPF	BGCP	HaLRTC	MissForest	VAE
20%, RM-G	7.33/**3.12**	10.24/4.33	8.42/3.64	8.42/3.61	8.15/3.33	**7.23**/3.40	16.55/7.30
40%, RM-G	**7.50/3.21**	10.41/4.44	8.47/3.65	8.33/3.60	8.87/3.61	8.22/3.80	18.39/8.31
20%, NM-G	10.21/4.28	10.34/4.35	10.62/4.42	10.21/4.27	10.46/4.21	12.53/5.13	**3.24/3.17**
40%, NM-G	10.43/4.51	10.89/4.52	11.38/4.70	10.25/**4.33**	10.88/4.38	18.61/7.46	**7.43**/5.10
10%, RM-B	**1.84/6.49**	13.13/69.77	10.65/36.08	6.33/19.60	4.85/17.35	-/-	43.08/255.56
30%, RM-B	**2.51/14.02**	15.90/106.98	10.67/40.78	6.99/22.15	6.64/26.79	-/-	56.03/302.84
10%, NM-B	8.04/**16.53**	20.29/47.15	24.63/86.38	10.90/29.46	9.47/34.72	-/-	**3.25**/38.19
30%, NM-B	15.52/59.42	20.25/69.20	-/-	15.57/**58.95**	**14.83**/92.59	-/-	16.63/170.86

Table 1: Performance comparison for RM and NM for imputation tasks on data sets (G) and (B).

	BTMF	BPMF-AR	TRMF	kNN+ARIMA	kNN+GRU	kNN+LSTM
Original G	7.40/**2.92**	7.58/2.96	**7.30**/2.99	7.35/3.29	7.72/3.36	7.55/3.28
20%, RM-G	**7.87/3.12**	8.37/3.26	11.11/4.34	9.13/3.76	10.13/4.06	10.12/4.01
40%, RM-G	**8.27/3.32**	9.02/3.53	11.22/4.40	10.90/4.24	12.51/4.72	12.55/4.69
20%, NM-G	10.38/4.15	10.39/4.03	11.14/4.35	**9.37**/4.01	9.68/4.06	9.50/**3.99**
40%, NM-G	**11.39/4.59**	11.74/4.64	12.46/4.78	11.44/4.72	11.66/4.74	11.51/4.68
Original B	**8.74/58.57**	14.38/101.84	16.53/113.90	-/-	-/-	-/-
10%, RM-B	**8.04/54.95**	14.17/107.26	24.16/120.60	-/-	-/-	-/-
30%, RM-B	**10.15/60.21**	14.60/109.74	21.74/124.74	-/-	-/-	-/-
10%, NM-B	**8.94/53.93**	13.21/88.29	24.66/150.53	-/-	-/-	-/-
30%, NM-B	**18.70**/157.89	23.08/**118.29**	-/-	-/-	-/-	-/-

Table 2: Performance comparison for RM and NM for prediction tasks on data sets (G) and (B).

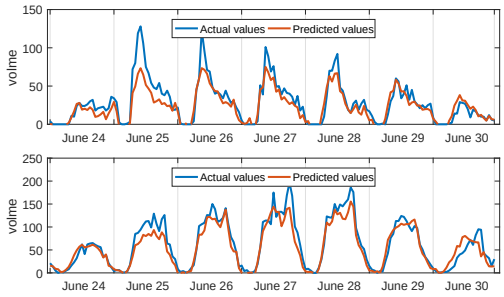


Figure 6: Examples of two pick-up/drop-off pairs (from zone 17 to zone 13 and from zone 27 to zone 27). We show the predicted time series using BTTF#1 (time lags are $\{1, 2, 24\}$) with 30% NM and the actual observations.

Experiment setup. Similar to the analyses on BTMF, we also design two missing data scenarios: random missing (RM) by randomly removing entries in the tensor and non-random missing (NM) by randomly selecting pick-up$\times$drop-off$\times$day combinations and for each of them removing the corresponding 24h block entirely. We examine two missing rates (10% and 30%) and use the last seven days (i.e., 168 time slots) as the prediction period. The Baseline TCF model in general makes a first-order Markovian assumption. Therefore, for better comparison we define two time lag sets: $\{1, 2, 24\}$ (BTTF#1) and $\{1\}$ (BTTF#2).

Results and analysis. Table 3 shows the performance of the two BTTF models and TCF on both imputation and prediction tasks. Essentially, BTTF achieves better performance on both imputation and prediction task. As an example, we depict the actual and predicted values for two randomly selected time series in Figure 6. We can see that the temporal trend is well characterized by the BTTF model.

		BTTF#1	BTTF#2	TCF
imputation	10%, RM	52.07/**4.60**	**51.24**/4.61	51.89/4.66
	30%, RM	52.39/**4.67**	52.39/4.71	**52.11**/4.73
	10%, NM	**52.24/4.74**	52.46/4.78	53.14/4.88
	30%, NM	**51.97/4.78**	52.59/4.82	53.11/5.07
prediction	Original	52.86/5.34	**46.20**/5.66	52.62/**4.14**
	10%, RM	52.56/5.36	**46.76**/6.05	52.78/**4.20**
	30%, RM	53.07/5.46	**47.46**/5.95	53.28/**4.32**
	10%, NM	52.52/5.42	**47.17**/5.64	51.86/**4.23**
	30%, NM	52.66/5.44	**47.04**/5.68	52.25/**4.65**

Table 3: Performance comparison on data set NYC taxi data (N).

6 Conclusion and Future Work

This paper presents the BTF framework by integrating an AR layer into traditional Bayesian probabilistic MF/TF algorithms. This integration allows us to model the complex temporal evolution of multidimensional time series data on the latent factor, thus providing a powerful tool to handle incomplete/corrupted time series data sets for both imputation and prediction tasks. For model inference, we derive an efficient and scalable Gibbs sampling procedure. The full Bayesian treatment also provides additional flexibility in terms of parameter tuning and avoids overfitting issues. We examine the framework on several real-world time series matrices/tensors, and BTF framework demonstrates superior performance over other baselines. Although we introduce BTF in a spatiotemporal setting, the model can work for any multidimensional time series. Since we only model temporal dynamics explicitly, BTF is limited in capturing spatial correlations or other dependencies. In the future, we will extend this framework to account for spatial dependencies by incorporating tools such as spatial AR and Laplacian kernels and also explore the application of deep learning models [Zhang *et al.*, 2017; Yu *et al.*, 2018; Liang *et al.*, 2018].

References

[Anava *et al.*, 2015] Oren Anava, Elad Hazan, and Assaf Zeevi. Online time series prediction with missing data. In *International Conference on Machine Learning*, volume 37, pages 2191–2199, 2015.

[Charlin *et al.*, 2015] Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 155–162. ACM, 2015.

[Che *et al.*, 2018] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

[Chen *et al.*, 2019] Xinyu Chen, Zhaocheng He, and Lijun Sun. A bayesian tensor decomposition approach for spatiotemporal traffic data imputation. *Transportation Research Part C: Emerging Technologies*, 98:73 – 84, 2019.

[Jing *et al.*, 2018] P. Jing, Y. Su, X. Jin, and C. Zhang. High-order temporal correlation model learning for time-series prediction. *IEEE Transactions on Cybernetics*, pages 1–13, 2018.

[Kolda and Bader, 2009] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[Li and Shahabi, 2018] Yaguang Li and Cyrus Shahabi. A brief overview of machine learning methods for short-term traffic forecasting and future directions. *SIGSPATIAL Special*, 10(1):3–9, 2018.

[Liang *et al.*, 2018] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. Geoman: Multi-level attention networks for geo-sensory time series prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3428–3434, 2018.

[Liu *et al.*, 2013] 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*, 35(1):208–220, 2013.

[McCoy *et al.*, 2018] John T McCoy, Steve Kroon, and Lidia Auret. Variational autoencoders for missing data imputation with application to a simulated milling circuit. *IFAC-PapersOnLine*, 51(21):141–146, 2018.

[Saad and Mansinghka, 2018] Feras Saad and Vikash Mansinghka. Temporally-reweighted chinese restaurant process mixtures for clustering, imputing, and forecasting multivariate time series. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 755–764, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[Salakhutdinov and Mnih, 2008] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *International Conference on Machine Learning*, pages 880–887, 2008.

[Stekhoven and Bühlmann, 2012] Daniel J. Stekhoven and Peter Bühlmann. Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[Sun and Axhausen, 2016] Lijun Sun and Kay W Axhausen. Understanding urban mobility patterns with a probabilistic tensor factorization framework. *Transportation Research Part B: Methodological*, 91:511–524, 2016.

[Takeuchi *et al.*, 2017] Koh Takeuchi, Hisashi Kashima, and Naonori Ueda. Autoregressive tensor factorization for spatio-temporal predictions. In *IEEE International Conference on Data Mining*, pages 1105–1110, 2017.

[Xiong *et al.*, 2010] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *SIAM International Conference on Data Mining*, pages 211–222, 2010.

[Yu *et al.*, 2016] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *Advances in Neural Information Processing Systems*, pages 847–855, 2016.

[Yu *et al.*, 2018] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 3634–3640, 2018.

[Zhang *et al.*, 2017] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI Conference on Artificial Intelligence*, pages 1655–1661, 2017.

[Zhao *et al.*, 2015] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1751–1763, 2015.