Dynamic Nonlinear Matrix Completion for Time-Varying Data Imputation

Jicong Fan[1,2]

[1]The Chinese University of Hong Kong (Shenzhen)
[2]Shenzhen Research Institute of Big Data
Shenzhen, China
fanjicong@cuhk.edu.cn

Abstract

Classical matrix completion methods focus on data with stationary latent structure and hence are not effective in missing value imputation when the latent structure changes with time. This paper proposes a dynamic nonlinear matrix completion (D-NLMC) method, which is able to recover the missing values of streaming data when the low-dimensional nonlinear latent structure of the data changes with time. The paper provides an efficient approach to updating the nonlinear model dynamically. D-NLMC incorporates the information of new data and remove the information of earlier data recursively. The paper shows that the missing data can be estimated if the change of latent structure is slow enough. Different from existing online or adaptive low-rank matrix completion methods, D-NLMC does not require the local low-rank assumption and is able to adaptively recover high-rank matrices with low-dimensional latent structures. Note that existing high-rank matrix completion methods have high-computational costs and are not applicable to streaming data with varying latent structures, which fortunately can be handled by D-NLMC efficiently and accurately. Numerical results show that D-NLMC outperforms the baselines in real applications.

Introduction

Low-rank matrix completion (LRMC) aims to recover the missing entries of a partially observed matrix of low rank (Candès and Recht 2009). It has numerous real applications such as image inpainting (Guillemot and Meur 2014), collaborative filtering (Su and Khoshgoftaar 2009), and classification (Goldberg et al. 2010). There are many LRMC algorithms proposed in the past decade (Wen, Yin, and Zhang 2012; Hu et al. 2013; Nie, Huang, and Ding 2012; Gu et al. 2014; Lu et al. 2014; Wang et al. 2014; Xie et al. 2016; Fan and Chow 2017; Fan et al. 2019). These algorithms are usually based on low-rank matrix factorization, nuclear norm minimization (Candès and Recht 2009), Schatten-p quasi-norm minimization (Nie, Huang, and Ding 2012), or their extensions (Hu et al. 2013; Gu et al. 2014).

LRMC assumes that the matrix to be recovered is low-rank or can be well approximated by a low-rank matrix. This assumption does not hold in the cases that the data are drawn from a nonlinear low-dimensional latent variable model or a union of low-dimensional subspaces (Fan 2021). To solve the problem, recently, a few researchers proposed high-rank matrix completion methods (Eriksson, Balzano, and Nowak 2011; Li and Vidal 2016; Alameda-Pineda et al. 2016; Elhamifar 2016; Ongie et al. 2017; Fan and Chow 2018; Fan and Cheng 2018; Fan, Zhang, and Udell 2020; Le Morvan et al. 2020). For instance, Ongie et al. (2017) and Fan and Chow (2018) proposed to minimize the rank of the matrix in a feature space induced by kernel to recover the missing values in the data space. Fan and Udell (2019) proposed a kernel factorization method for high-rank matrix completion, which is more efficient than the methods of (Ongie et al. 2017) and (Fan and Chow 2018).

The aforementioned matrix completion methods focus on the case that the latent structure of data is stationary. In many real applications especially time series analysis (Afrifa-Yamoah et al. 2020), the latent structure of data often changes with time. The problem is also closely related to dynamic subspace tracking (Balzano, Nowak, and Recht 2010; Narayanamurthy and Vaswani 2018; Vaswani et al. 2018) and time series imputation (Yozgatligil et al. 2013; Yu, Rao, and Dhillon 2015), which have been systematically discussed in the review paper of Vaswani and Narayanamurthy (2018). One naive method for the case of changing latent structure is performing static matrix completion on the data segments selected by a sliding window with small width. However, such a method is time-consuming because it has to perform singular value decomposition (SVD), eigenvalue decomposition (EVD), or matrix factorization for lt times, where t denotes the total number of data samples currently and l denotes the number of optimization iterations for the matrix in the sliding window. For example, the nuclear minimization method has a time complexity of $O(dw^2lt)$ on a streaming dataset of size $d \times t$, where w is the width of the sliding window and we have assumed $w > d$. When using the NLMC method of (Fan and Chow 2018), the time complexity is $O(w^3lt)$. Therefore, such a naive method is not applicable to large datasets.

Related work A few researchers have studied the missing data imputation problem in the case of time-varying latent structures (Balzano, Nowak, and Recht 2010; Devooght, Kourtellis, and Mantrach 2015; Xu and Davenport 2016; Chouvardas et al. 2017; Balzano, Chi, and Lu 2018; Afrifa-Yamoah et al. 2020). For instance, Brand (2003) proposed an online SVD method for recommendation system. Dhanjal, Gaudel, and Clémençon (2014) proposed to perform randomized SVD (Halko, Martinsson, and Tropp 2011) successively for nuclear norm minimization based matrix completion. Guo (2015) proposed an online LRMC method based on matrix factorization. In the method, an online data sample $\boldsymbol{x}_t$ is reconstructed by $\boldsymbol{U}^{(t-1)}\boldsymbol{v}_t$ and $\boldsymbol{U}$ is updated dynamically and hence is able to fit the changes of the subspace. One limitation of the method is that it is hard to ensure that the subspace change at t is accurately encoded by $\boldsymbol{U}^{(t)}$. In other word, the performance of the imputation heavily relies on how much $\boldsymbol{U}$ is updated. Motivated from the least-mean square algorithm, Tripathi, Mohan, and Rajawat (2017) provided an adaptive LRMC method for time-varying scenarios. The method showed promising performance in mobile network localization and streaming video denoising. It should be pointed out that these methods are not applicable to streaming data with time-varying nonlinear latent structures, which widely exist in real problems.

Contributions In this work, we focus on the missing value imputation for time-varying streaming data.

- The paper presents a dynamic nonlinear matrix completion method for streaming data with time-varying nonlinear latent structures.
- The paper provides a fast rank-two modification method of eigenvalue decomposition to improve the efficiency of the proposed online imputation method.
- The paper provides theoretical analysis for the time-varying nonlinear latent variable model and verifies the effectiveness of the proposed method.

Numerical results on synthetic data and real data show that the proposed method has state-of-the-art performance.

Proposed Method

Property of Time-Varying Nonlinear Model

First, we make the following assumption.

Assumption 1. *Suppose the d-dimensional sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T\}$ is generated from*

$$\boldsymbol{x}_t = g_t(\boldsymbol{z}_t), \tag{1}$$

where $\boldsymbol{z}_t = \boldsymbol{A}\boldsymbol{z}_{t-1} + \boldsymbol{\varepsilon}_t$, $\boldsymbol{A} \in \mathbb{R}^{r\times r}$, $\boldsymbol{\varepsilon}_t \in \mathbb{R}^r$ is independently drawn from a distribution $\mathcal{D}_\varepsilon$, $g_t : \mathbb{R}^r \rightarrow \mathbb{R}^d$ is a polynomial function with order at most θ and with parameters relying on t, and $\|g_t(\boldsymbol{z}) - g_{t-1}(\boldsymbol{z})\| \leq \gamma\|\boldsymbol{z}\|$, $\forall \boldsymbol{z} \in \mathbb{R}^r$.

More intuitively, we give an example of g_t.

Example 1. *Let $r = 1$, $d = 3$, and $\theta = 3$. $x_{t1} = z_t$, $x_{t2} = (1 + \sin(0.01t))z_t^2$, $x_{t3} = z_t^3$.*

Assumption 1 defined a time-varying nonlinear latent variable model in the form of a multivariate polynomial function g_t, in which $\boldsymbol{z}$ is generated from an auto regressive model. $\mathcal{D}_\varepsilon$ has many choices such as Gaussian distribution or uniform distribution. When $\boldsymbol{A} = \boldsymbol{0}$, $\boldsymbol{z}_t$ reduces to $\boldsymbol{\varepsilon}_t$ and $\{\boldsymbol{x}_t\}$ are independent samples. The parameter γ quantifies the rate of the change of the latent structure. When γ is large, the latent structure of the data changes quickly, which will make the missing value imputation problem more difficult.

We use polynomial assumption because it is universal to approximate smooth functions (prevalent in real problems) according to the Taylor series, provided that θ is sufficiently large. For example, let h be an arbitrary smooth function, then for any ϵ, there exists a polynomial function g such that $\|h(\boldsymbol{z}) - g(\boldsymbol{z})\| \leq \epsilon$. Therefore, we can extend Assumption 1 to smooth functions via letting $\boldsymbol{x}_t = h_t(\boldsymbol{z}) = g_t(\boldsymbol{z}) + \boldsymbol{\epsilon}_t$, where $h_t : \mathbb{R}^r \rightarrow \mathbb{R}^d$ is a smooth function associated with t and $\boldsymbol{\epsilon}_t$ denotes the residuals.

With Assumption 1, we have

Theorem 1. *Suppose $\boldsymbol{X}_t = [\boldsymbol{x}_{t-w+1}, \boldsymbol{x}_{t-w+2}, \ldots, \boldsymbol{x}_t]$ is given by Assumption 1. Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^{\binom{d+q}{q}}$ be a q-order polynomial feature map*[1]. *Let $c_t = \max(\|\boldsymbol{z}_{t-w+1}\|, \ldots, \|\boldsymbol{z}_t\|)$. Then with probability 1, there exists a matrix $\hat{\boldsymbol{X}}_t$ with rank at most $\min\left\{\binom{r+\theta}{\theta}, d, w\right\}$ such that $\|\boldsymbol{X}_t - \hat{\boldsymbol{X}}_t\|_F \leq \dfrac{\gamma c_t w^{1.5}}{3}$ and $\operatorname{rank}(\phi(\hat{\boldsymbol{X}}_t)) \leq \min\left\{\binom{r+\theta q}{\theta q}, \binom{d+q}{q}, w\right\}$.*

The theorem[2] indicates that, although $\boldsymbol{X}_t$ can be full-rank (when θ is large enough), it can be approximated by a matrix $\hat{\boldsymbol{X}}_t$ with error at most $\gamma c_t w^{1.5}/3$, where the polynomial feature matrix $\phi(\hat{\boldsymbol{X}}_t)$ is low-rank relatively provided that r is much smaller than d and w is sufficiently large. When $\gamma = 0$, $\phi(\boldsymbol{X}_t)$ can be exactly low-rank. When γ is small enough, namely, the change of the latent structure is slow enough, we can well approximate $\boldsymbol{X}_t$ with $\hat{\boldsymbol{X}}_t$. More intuitively, let δ be a sufficiently large constant, we have

$$\|\phi(\boldsymbol{X}_t) - \phi(\hat{\boldsymbol{X}}_t)\|_F \leq \delta\|\boldsymbol{X}_t - \hat{\boldsymbol{X}}_t\|_F \leq \frac{\delta\gamma c_t w^{1.5}}{3}. \tag{2}$$

It means that $\phi(\boldsymbol{X}_t)$ is approximately low-rank provided that γ is small enough. Therefore, we may recover the missing values of $\boldsymbol{X}_t$ by minimizing the rank, nuclear norm, or Schatten-p quasi-norm of $\phi(\boldsymbol{X}_t)$.

Note that in Assumption 1, if the coefficients of the polynomial function g are randomly generated, the equalities for the rank of $\hat{\boldsymbol{X}}_t$ and $\phi(\hat{\boldsymbol{X}}_t)$ in Theorem 1 hold almost surely, namely, $\operatorname{rank}(\hat{\boldsymbol{X}}_t) = \min\left\{\binom{r+\theta}{\theta}, d, w\right\}$ and $\operatorname{rank}(\phi(\hat{\boldsymbol{X}}_t)) = \min\left\{\binom{r+\theta q}{\theta q}, \binom{d+q}{q}, w\right\}$. Therefore, Theorem 1 is a general case of the rank property of $\hat{\boldsymbol{X}}_t$ and $\phi(\hat{\boldsymbol{X}}_t)$. The worst or most difficult case happens when the equalities hold.

Dynamic Nonlinear Matrix Completion

In this study, we aims to recover the missing values of the sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots\}$ generated from Assumption 1 without knowing g_t, $\boldsymbol{\varepsilon}_t$, $\boldsymbol{A}$, and $\boldsymbol{z}_t$. Inspired by (Ongie et al. 2017) and (Fan and Chow 2018), we here propose to solve

$$\underset{[\boldsymbol{X}_t]_{\bar{\Omega}_t}}{\text{minimize}} \ \|\phi(\boldsymbol{X}_t)\|_{S_p}^p, \quad t = 1, 2, \ldots \tag{3}$$

[1] An example for ϕ: let $d = 2$ and $q = 2$. $\phi(\boldsymbol{x}_t) = [1, x_{t1}, x_{t2}, x_{t1}^2, x_{t2}^2, x_{t1}x_{t2}]^\top$.

[2] The proof is in the appendix.

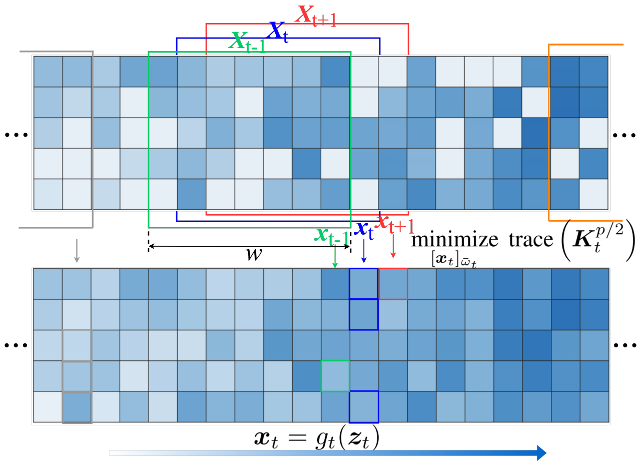

Figure 1: Dynamic nonlinear matrix completion (the white squares denote missing values)

where $\boldsymbol{X}_t = [\boldsymbol{x}_{t-w+1}, \boldsymbol{x}_{t-w+2}, \ldots, \boldsymbol{x}_t]$. Ω_t ($\bar{\Omega}_t$) denotes the set of locations of the observed (missing) entries of $\boldsymbol{X}_t$. $\|\boldsymbol{Y}\|_{Sp}$ denotes the Schatten-p quasi-norm of $\boldsymbol{Y}$, i.e., $\|\boldsymbol{Y}\|_{Sp} = (\sum_i \sigma_i^p(\boldsymbol{Y}))^{1/p}$, where $\sigma_i(\boldsymbol{Y})$ denotes the i-th singular value of $\boldsymbol{Y}$ and $0 < p < 1$. Suppose ϕ is the feature map given by a kernel function $k(\cdot, \cdot)$, then problem (3) can be reformulated as

$$\underset{[\boldsymbol{X}_t]_{\bar{\Omega}_t}}{\text{minimize}} \ \text{trace}\left(\boldsymbol{K}_t^{p/2}\right), \quad t = 1, 2, \ldots \tag{4}$$

where $\boldsymbol{K}_t = \phi(\boldsymbol{X}_t)^\top \phi(\boldsymbol{X}_t) \in \mathbb{R}^{w \times w}$. The feature map ϕ of a polynomial kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^\top \boldsymbol{x}_j + a)^q$ is exactly a q-order polynomial feature map, which matches the condition in Theorem 1. The feature map of a Gaussian kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$ is an infinite-order polynomial feature map but the weight of high-order terms decrease quickly especially when σ is large (Fan, Zhang, and Udell 2020). Hence a Gaussian kernel with a large enough σ can be well approximated by low-order polynomial kernels. It indicates that when we use a Gaussian kernel, $\phi(\boldsymbol{X}_t)^\top \phi(\boldsymbol{X}_t)$ is still approximately low-rank, and (4) is useful to recover the missing entries of $\boldsymbol{X}_t$, which will be further justified later.

Suppose we have recovered the missing values of $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_w$ using (4), for $\boldsymbol{X}_{w+1}$, there is no need to estimate the missing values of $\boldsymbol{x}_2, \ldots, \boldsymbol{x}_w$ again. Then in stead of (4), we proposed to solve the following problem

$$\underset{[\boldsymbol{x}_t]_{\bar{\omega}_t}}{\text{minimize}} \ \text{trace}\left(\boldsymbol{K}_t^{p/2}\right), \quad t = 1, 2, \ldots \tag{5}$$

where $\bar{\omega}_t$ denotes the locations of the missing entries of $\boldsymbol{x}_t$. The scheme is shown in Figure 1. The method is called dynamic nonlinear matrix completion (D-NLMC).

Since problem (5) is nonconvex and the number of decision variables is not large, we propose to use L-BFGS (Liu and Nocedal 1989) to solve the optimization. Denote by $\mathcal{L}_t$ the objective function in (5). We can get the gradient by

$$\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{K}_t} = \frac{p}{2} \boldsymbol{K}_t^{\frac{p}{2}-1} = \frac{p}{2} \boldsymbol{V}_t \boldsymbol{\Lambda}_t^{\frac{p}{2}-1} \boldsymbol{V}_t^\top, \tag{6}$$

where $\boldsymbol{V}_t$ and $\text{diag}(\boldsymbol{\Lambda}_t)$ are the eigenvectors and eigenvalues of $\boldsymbol{K}_t$ respectively. For convenience, suppose we are using the Gaussian kernel. We have

$$\begin{aligned} \frac{\partial \mathcal{L}_t}{\partial [\boldsymbol{x}_t]_{\bar{\omega}}} &= \sum_{i=1}^{w} \sum_{j=1}^{w} \frac{\partial \mathcal{L}_t}{\partial [\boldsymbol{K}_t]_{ij}} \frac{\partial [\boldsymbol{K}_t]_{ij}}{\partial [\boldsymbol{x}_t]_{\bar{\omega}}} \\ &= \left[\frac{2}{\sigma^2} \left(\boldsymbol{X}_t \boldsymbol{\alpha} - \sum_{j=1}^{w} \alpha_i \boldsymbol{x}_t \right) \right]_{\bar{\omega}}, \end{aligned} \tag{7}$$

where $\boldsymbol{\alpha} = \left[\frac{\partial \mathcal{L}_t}{\partial \boldsymbol{K}_t}\right]_{:w} \odot [\boldsymbol{K}_t]_{:w}$. The gradient involving polynomial kernels is in the appendix.

When using the Gaussian kernel, we have the following theoretical result.

Theorem 2. *Let $\boldsymbol{K}_t$ be the Gaussian kernel matrix with parameter σ. There exists a matrix $\tilde{\boldsymbol{K}}_t$ with rank at most min $\left\{\binom{r+\theta q}{\theta q}, \binom{d+q}{q}, w\right\}$ such that*

$$\|\boldsymbol{K}_t - \tilde{\boldsymbol{K}}_t\|_F \leq \frac{C_t \gamma w^2}{2\sigma^2} + \frac{C_t' w}{\sigma^{2(q+1)}(q+1)!}, \tag{8}$$

where C_t and C_t' are positive values relying on θ, q, and $\max(\|\boldsymbol{z}_{t-w+1}\|, \ldots, \|\boldsymbol{z}_t\|)$.

The specific values of C_t and C_t' are in the appendix. The theorem indicates that the Gaussian kernel matrix $\boldsymbol{K}_t$ is approximately low-rank provided that w is much larger than r, which explained the effectiveness of (5) when we use the Gaussian kernel. In (8), larger σ leads to tighter upper bound. When q increases, the rank of $\tilde{\boldsymbol{K}}_t$ becomes higher but the upper bound becomes tighter. Since the diagonal elements of $\boldsymbol{K}_t$ are all ones, the nuclear norm of $\boldsymbol{K}_t$ is a constant w. Hence, using $p/2$ instead of 1 in (5) is reasonable.

Note that in (6), we need to compute the EVD of $\boldsymbol{K}_t$, which is time-consuming. Let $\boldsymbol{K}_{t-1}' := \phi(\boldsymbol{X}_{t-1}')^\top \phi(\boldsymbol{X}_{t-1}')$, where $\boldsymbol{X}_{t-1}' = [\boldsymbol{x}_{t-w+1}, \ldots, \boldsymbol{x}_{t-1}]$. Suppose we have already got the eigenvalues and eigenvectors of $\boldsymbol{K}_{t-1}'$, we propose to compute the EVD of $\boldsymbol{K}_t$ by exploiting the EVD of $\boldsymbol{K}_{t-1}'$. In this study, we take advantage of the fast low-rank modification method proposed by (Brand 2006). Specifically, denoting $\boldsymbol{k}_t' = [k(\boldsymbol{x}_{t-w+1}, \boldsymbol{x}_t), \ldots, k(\boldsymbol{x}_{t-1}, \boldsymbol{x}_t)]^\top$, we have

$$\boldsymbol{K}_t = \begin{bmatrix} \boldsymbol{K}_{t-1}' & \boldsymbol{k}_t' \\ \boldsymbol{k}_t'^\top & k(\boldsymbol{x}_t, \boldsymbol{x}_t) \end{bmatrix}. \tag{9}$$

Suppose the truncated EVD of $\boldsymbol{K}_{t-1}'$ is $\boldsymbol{K}_{t-1}' \approx \boldsymbol{V}_{t-1}' \boldsymbol{\Lambda}_{t-1}' \boldsymbol{V}_{t-1}'^\top$, where $\boldsymbol{V}_{t-1}' \in \mathbb{R}^{(w-1) \times R}$. According to Theorem 1, we have $R \leq \binom{r+\theta q}{\theta q}$ provided that w/r is sufficiently large. We rewrite (9) as

$$\boldsymbol{K}_t = \bar{\boldsymbol{V}}_{t-1} \bar{\boldsymbol{\Lambda}}_{t-1} \bar{\boldsymbol{V}}_{t-1}^\top + \boldsymbol{G}_t \boldsymbol{H}_t^\top \tag{10}$$

where $\bar{\boldsymbol{V}}_{t-1} = [\boldsymbol{V}_{t-1}'^\top \ \boldsymbol{0}]^\top$, $\bar{\boldsymbol{\Lambda}}_{t-1} = \boldsymbol{\Lambda}_{t-1}'$, $\boldsymbol{G}_t = [\boldsymbol{e}_w \ \bar{\boldsymbol{k}}']$, and $\boldsymbol{H}_t = [\tilde{\boldsymbol{k}}_t' \ \boldsymbol{e}_w]$. $\boldsymbol{e}_w = [0, 0, \ldots, 0, 1]^\top$, $\bar{\boldsymbol{k}}' = [\boldsymbol{k}'^\top \ 0]^\top$, and $\tilde{\boldsymbol{k}}' = [\boldsymbol{k}'^\top \ k(\boldsymbol{x}_t, \boldsymbol{x}_t)]^\top$. Then according to (Brand 2006), the EVD of $\boldsymbol{K}_t$ can be efficiently computed from

```
Algorithm 1: Fast EVD for $\boldsymbol{K}_t$
Input: $\bar{\boldsymbol{V}}_{t-1}, \bar{\boldsymbol{\Lambda}}_{t-1}, \boldsymbol{G}_t, \boldsymbol{H}_t$
  1: $\begin{bmatrix}\bar{\boldsymbol{V}}_{t-1} & \boldsymbol{P}\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & \bar{\boldsymbol{V}}_{t-1}^\top\boldsymbol{G}_t \\ \boldsymbol{0} & \boldsymbol{R_G}\end{bmatrix} \xleftarrow{\text{QR decomp.}} \begin{bmatrix}\bar{\boldsymbol{V}}_{t-1} & \boldsymbol{G}_t\end{bmatrix}$
  2: $\begin{bmatrix}\bar{\boldsymbol{V}}_{t-1} & \boldsymbol{Q}\end{bmatrix}\begin{bmatrix}\boldsymbol{I} & \bar{\boldsymbol{V}}_{t-1}^\top\boldsymbol{H}_t \\ \boldsymbol{0} & \boldsymbol{R_H}\end{bmatrix} \xleftarrow{\text{QR decomp.}} \begin{bmatrix}\bar{\boldsymbol{V}}_{t-1} & \boldsymbol{H}_t\end{bmatrix}$
  3: $\boldsymbol{C} \longleftarrow \begin{bmatrix}\bar{\boldsymbol{\Lambda}}_{t-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} + \begin{bmatrix}\bar{\boldsymbol{V}}_{t-1}^\top\boldsymbol{G}_t \\ \boldsymbol{R_G}\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{V}}_{t-1}^\top\boldsymbol{H}_t \\ \boldsymbol{R_H}\end{bmatrix}$
  4: $\boldsymbol{C} = \boldsymbol{V_C}\boldsymbol{S_C}\boldsymbol{V_C}^\top$
  5: $\boldsymbol{V}_t \longleftarrow \begin{bmatrix}\bar{\boldsymbol{V}}_{t-1} & \boldsymbol{Q}\end{bmatrix}\boldsymbol{V_C},\ \ \boldsymbol{\Lambda}_t \longleftarrow \boldsymbol{S_C}$
Output: $\boldsymbol{K}_t = \boldsymbol{V}_t\boldsymbol{\Lambda}_t\boldsymbol{V}_t^\top$.
```

```
Algorithm 2: Fast EVD for $\boldsymbol{K}_t'$
Input: $\boldsymbol{V}_t, \boldsymbol{\Lambda}_t, \boldsymbol{G}_{t-w+1}, \boldsymbol{H}_{t-w+1}$
  1: Follow the procedures 1–5 of Algorithm 1
  2: Remove the smallest two eigenvalues and the corresponding eigenvectors
Output: $\boldsymbol{K}_t' = \boldsymbol{V}_t'\boldsymbol{\Lambda}_t'\boldsymbol{V}_t'^\top$.
```

$\{\bar{\boldsymbol{V}}_{t-1}, \bar{\boldsymbol{\Lambda}}_{t-1}, \boldsymbol{G}_t, \boldsymbol{H}_t\}$, which is detailed in Algorithm 1. Note that we can reformulate Algorithm 1 as two rank-one modifications to obtain further acceleration, which is detailed in the appendix.

After we complete $\boldsymbol{x}_t$ using (5), we need to compute the EVD of $\boldsymbol{K}_t'$ so as to compute the EVD of $\boldsymbol{K}_{t+1}$ efficiently, where $\boldsymbol{K}_t' = \phi(\boldsymbol{X}_t')^\top\phi(\boldsymbol{X}_t')$ and $\boldsymbol{X}_t' = [\boldsymbol{x}_{t-w+2}, \ldots, \boldsymbol{x}_t]$. We can get the EVD of $\boldsymbol{K}_t'$ efficiently from $\boldsymbol{K}_t$ which we have obtained after solving (5). Specifically, let $\boldsymbol{G}_{t-w+1} = [\boldsymbol{e}_1 \ \ -\bar{\boldsymbol{k}}_{t-w+1}]$ and $\boldsymbol{H}_{t-w+1} = [-\tilde{\boldsymbol{k}}_{t-w+1} \ \boldsymbol{e}_1]$, where $\boldsymbol{e}_1 = [1, 0, \ldots, 0, 0]^\top$, $\bar{\boldsymbol{k}}_{t-w+1} = [0 \ \ \boldsymbol{k}_{t-w+1}^\top]^\top$, $\tilde{\boldsymbol{k}}_{t-w+1} = [k(\boldsymbol{x}_{t-w+1}, \boldsymbol{x}_{t-w+1}) \ \boldsymbol{k}_{t-w+1}^\top]^\top$, and $\boldsymbol{k}_{t-w+1} = [k(\boldsymbol{x}_{t-w+1}, \boldsymbol{x}_{t-w+2}), \ldots, k(\boldsymbol{x}_{t-w+1}, \boldsymbol{x}_t)]^\top$. We have

$$\begin{bmatrix}0 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_t'\end{bmatrix} = \boldsymbol{K}_t + \boldsymbol{G}_{t-w+1}\boldsymbol{H}_{t-w+1}^\top. \tag{11}$$

Then the EVD of $\boldsymbol{K}_t'$ is computed by Algorithm 2. It is nearly the same as Algorithm 1 except for the step of the reduction of eigenvalues and eigenvectors, because Algorithm 1 raised the number of eigenvalues and eigenvectors by two.

The entire algorithm is shown in Algorithm 3. At time point t, the algorithm uses $\{\boldsymbol{x}_{t-w+1}, \ldots, \boldsymbol{x}_{t-1}\}$ to recover the missing values of $\boldsymbol{x}_t$ and hence can adapt to the changes of the latent structure. In addition, since we have used Algorithm 1 and Algorithm 2, D-NLMC is efficient and applicable to large datasets. Specifically, the time complexity (per iteration) of solving (5) without using Algorithm 1 and Algorithm 2 is $O(w^3)$. By using Algorithm 1 and Algorithm 2, the time complexity is reduced to $O(w^2R + R^3)$, where $w > R$. Note that the improvement is actually much larger than w/R times because the omitted constant in $O(w^3)$ (from EVD) is much larger than the omitted constant in $O(w^2R)$ (from matrix multiplication), which will be verified by the right plot of Figure 2 in Section .

```
Algorithm 3: D-NLMC
Input: $\boldsymbol{X}_0$, $R$.
  1: Recover the missing entries of $\boldsymbol{X}_0$ via solving (4)
  2: Compute partial ($R$) EVD of $\boldsymbol{K}_0'$, i.e., $\boldsymbol{K}_0' \approx \boldsymbol{V}_0'\boldsymbol{\Lambda}_0'\boldsymbol{V}_0'^\top$
  3: for $t = 1, 2, 3, \ldots$ do
  4:     Complete $\boldsymbol{x}_t$ via solving (5) (L-BFGS incorporating (7) and Algorithm 1)
  5:     Compute the EVD of $\boldsymbol{K}_t'$ by Algorithm 2
  6: end for
Output: Completed $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots$
```

Experiments

Synthetic Data

We consider the following data generating model

$$\begin{aligned} x_1(t) &= z(t), \\ x_2(t) &= z(t)^2 \times \alpha(t), \\ x_3(t) &= z(t)^3, \end{aligned} \tag{12}$$

where $t = 1, 2, \ldots, T$, $z(t)$ is drawn from $\mathcal{U}(0, 1)$ independently, and $\alpha(t) = 1 + \sin(\beta t)$. The parameter β controls the change rate of the nonlinear latent structure. When $\beta = 0$, the model is static. Suppose we get a data matrix $\boldsymbol{X}$ of size $3 \times \Delta_t$ from (12) and $\Delta_t = t_2 - t_1 \geq 3$, the rank of $\boldsymbol{X}$ is 3, though the latent dimension is only 1. In this study, we let $T = 2020$ and get a matrix $\boldsymbol{X} \in \mathbb{R}^{3\times 2020}$. We randomly remove a fraction of entries in the last 2000 columns of $\boldsymbol{X}$ to test the performance of the proposed method. The first 20 columns of $\boldsymbol{X}$ do not have missing values. We compare our D-NLMC with the following baselines:

- **NNM** nuclear norm minimization based low-rank matrix completion method (Candès and Recht 2009)
- **NNM+** perform NNM in a sliding window of data continuously
- **OL-LRMC** the online low-rank matrix completion method proposed by (Guo 2015)
- **VMC** the algebraic variety method proposed by (Ongie et al. 2017)
- **PMC-T** the polynomial matrix completion method proposed by (Fan, Zhang, and Udell 2020)
- **NLMC+** perform NLMC (Fan and Chow 2018) in a sliding window of data continuously

We tune the hyper-parameters of all algorithms carefully to provide their best performance as much as possible. In D-NLMC, we set $w = 20$, $R = 15$, and use Gaussian kernel with $\sigma = \mu w^{-2}\sum_{i=1}^{w}\sum_{j=1}^{w}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ (similar to (Fan, Zhang, and Udell 2020)), where μ is a constant such as 1 or 3. In this case, we use $\mu = 1$. Note that through out this paper, we let the w in OL-LRMC be the same as the w in D-NLMC. We evaluate the performance of missing data imputation using the following relative error:

$$\text{RE} = \sqrt{\sum_{(i,j)\in\bar{\Omega}}(x_{ij} - \hat{x}_{ij})^2 \Big/ \sum_{(i,j)\in\bar{\Omega}} x_{ij}^2}, \tag{13}$$

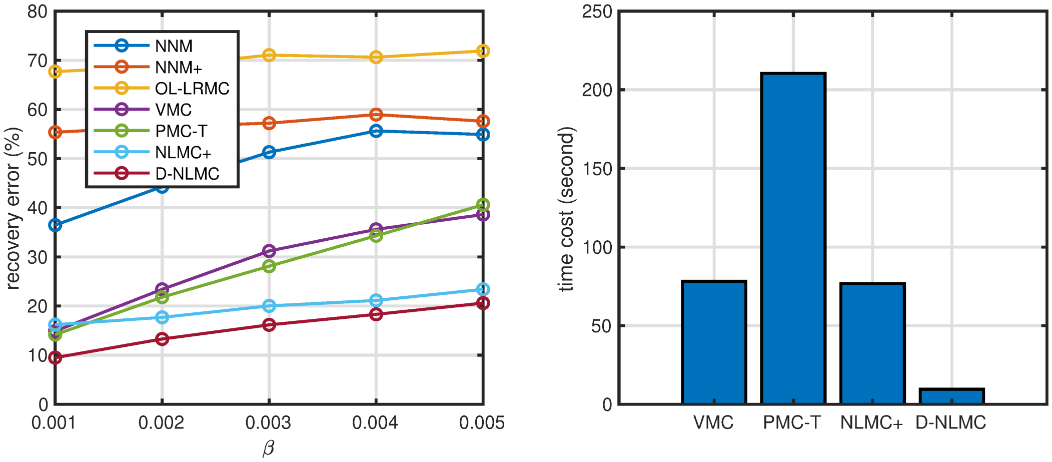

Figure 2: Recovery error and time cost on the synthetic data with different β ($\rho = 0.5$)

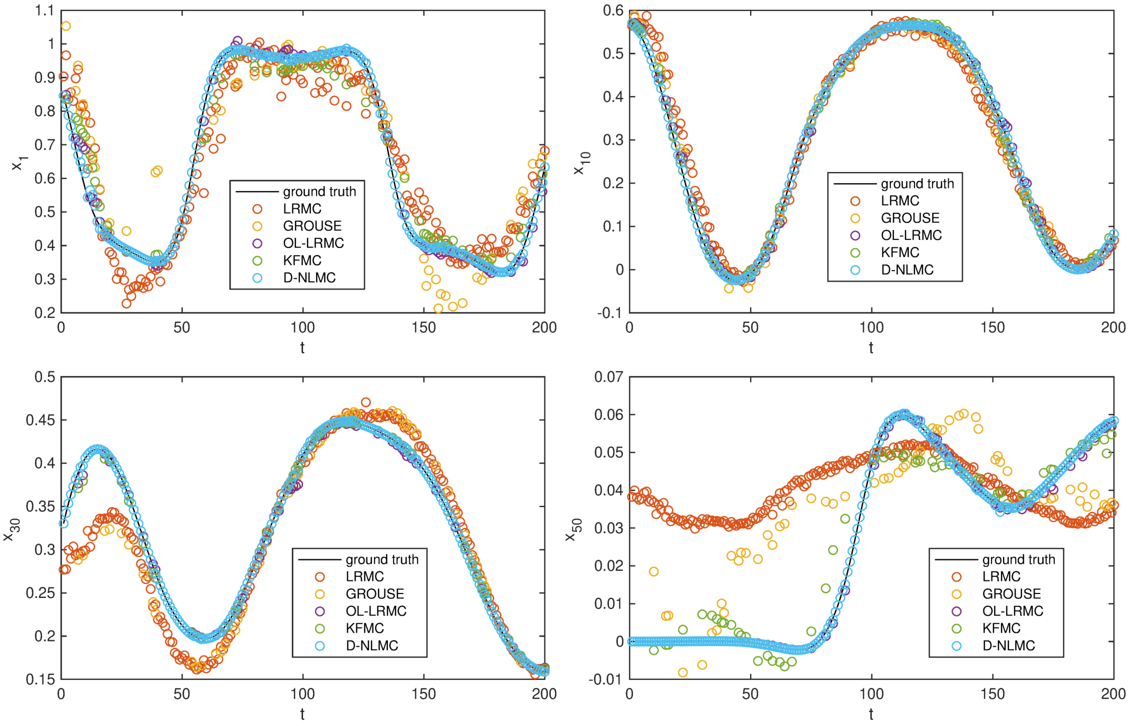

Figure 3: Examples of Chlorine level data imputation (variables 1, 10, 30, and 50) in the case of random missing pattern

where $\bar{\Omega}$ denotes the locations of the missing values. We report the average recovery error of 20 repeated trials in the left of Figure 2. The low-rank matrix completion methods NNM, NNM+, and OL-LRMC have very high recovery error because the data were drawn from a nonlinear model, namely, (12). The recovery errors of VMC and PMC-T increase quickly when β is large because they are static methods and are not able to adapt to the changes of latent structure. Our method D-NLMC is more accurate than other methods in all cases. The right of Figure 2 compares the time costs of the nonlinear methods. D-NLMC is more efficient than VMC, PMC, and NLMC+.

Chlorine Level Dataset

We test the proposed method on the Chlorine level dataset used in (Papadimitriou, Sun, and Faloutsos 2005; Balzano, Nowak, and Recht 2010). The dataset has 166 variables and 4610 samples. Since the number of the samples is too large for NNM+, VMC, PMC-T and NLMC+, we only consider NNM and OL-LRMC. We also compare the GROUSE method of (Balzano, Nowak, and Recht 2010) and the KFMC method of (Fan and Udell 2019), which are scalable

		NNM	GROUSE	OL-LRMC	KFMC	D-NLMC
missing randomly	ρ=0.1	8.18±0.05	7.57±0.12	2.21±0.01	3.61±0.04	**0.17**±0.01
	ρ=0.3	9.14±0.04	7.86±0.12	3.06±0.02	4.04±0.05	**0.78**±0.04
	ρ=0.5	10.51±0.46	9.12±0.71	4.55±0.01	4.43±0.04	**2.78**±0.06
missing non-randomly	κ=100	8.92±0.27	8.63±0.45	6.73±0.46	5.23±0.38	**2.29**±0.41
	κ=500	9.25±0.18	8.77±0.25	6.91±0.29	5.41±0.28	**2.96**±0.33
	κ=1000	9.84±0.34	8.95±0.44	8.27±0.52	5.69±0.22	**3.63**±0.29

Table 1: Recovery error (%) on the Chlorine level dataset (20 repeated trials; ρ denotes the missing rate in the random missing pattern and κ denotes the number of missing blocks in the non-random missing pattern)

		NNM	GROUSE	OL-LRMC	KFMC	D-NLMC
missing randomly	ρ=0.1	10..06±0.35	15.18±0.26	6.68±0.29	8.04±0.51	**5.10**±0.38
	ρ=0.3	13.54±0.36	17.59±0.38	8.02±0.35	10.51±0.52	**7.19**±0.63
	ρ=0.5	18.34±0.29	21.22±0.94	17.93±1.38	15.69±1.07	**12.07**±1.14
missing non-randomly	κ=100	20.51±2.05	24.66±3.46	12.93±1.92	13.51±2.67	**11.25**±3.06
	κ=500	22.43±1.44	27.09±2.36	18.78±3.97	17.62±2.68	**14.07**±1.64
	κ=1000	27.91±1.32	29.56±2.01	33.68±4.34	19.82±1.98	**17.64**±1.56

Table 2: Recovery error (%) on the SML2010 indoor temperature data (20 repeated trials)

		NNM	GROUSE	OL-LRMC	KFMC	D-NLMC
missing randomly	ρ=0.1	27.09±0.43	35.49±0.57	17.83±0.31	20.44±0.96	**14.65**±0.72
	ρ=0.3	31.11±0.21	38.22±0.85	22.05±0.20	21.96±0.67	**17.63**±0.59
	ρ=0.5	39.69±0.51	44.61±1.38	33.46±0.35	**27.18**±0.54	28.53±0.37
missing non-randomly	κ=100	27.49±1.87	36.93±1.63	20.64±1.94	19.75±2.63	**14.85**±2.57
	κ=500	26.58±1.48	39.20±0.92	21.52±0.76	19.92±2.32	**18.93**±2.29
	κ=1000	28.34±0.86	40.17±1.05	23.26±1.37	21.02±1.27	**19.89**±2.36

Table 3: Recovery error (%) on the Air Quality data (20 repeated trials)

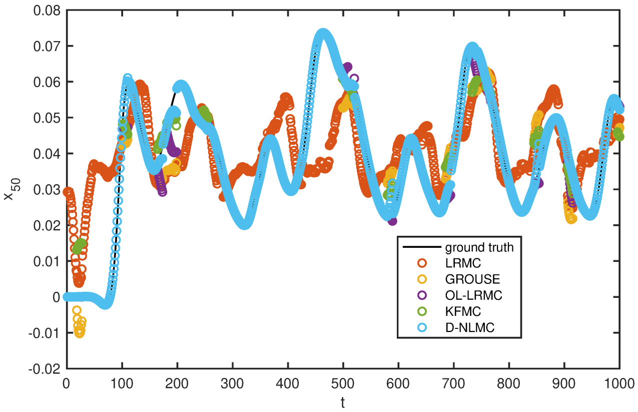

Figure 4: Examples of Chlorine level data imputation (variable 50) in the case of non-random missing pattern

to large datasets. In D-NLMC, we set $w = 100$, $R = 50$, and $\mu = 1$.

We consider two different patterns of missing data. The first one is randomly missing. We randomly remove 10%, 30%, or 50% entries of the data matrix. In the second pattern, we randomly remove 100, 500, or 1000 squares of size 10×10 from the data matrix. Figure 3 shows an example of completion performance in the case of randomly missing, where the missing rate ρ is 0.3. We see that the data recovered by our D-NLMC are nearly the same as the ground truth while other methods especially LRMC and GROUSE have much higher recovery errors. Figure 4 presents an example when the missing values emerge in random blocks. Since LRMC is a linear and static method, it cannot handle the nonlinearity and nonstationarity. The proposed method D-NLMC can adapt to the changes of the latent structure and explore the nonlinearity, which makes it have better recovery performance visually. We report the average results of 20 repeated trials in Table 1. It can be seen that our method D-NLMC outperformed other methods significantly.

SML2010 Indoor Temperature Data

We test the proposed method on the SML2010 indoor temperature dataset[3] from the UCI machine learning repository. The dataset consists of 2764 samples of 24 variables such as

[3] https://archive.ics.uci.edu/ml/datasets/SML2010

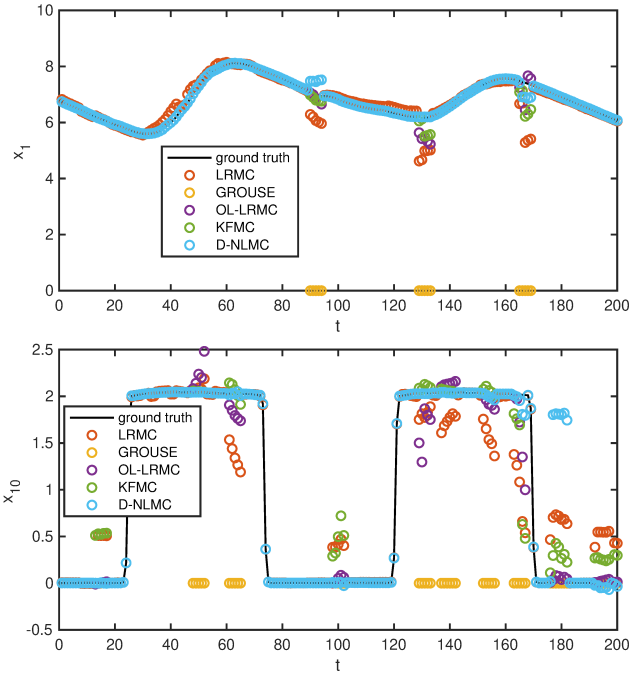

Figure 5: Examples of SML2010 imputation (variables 1 and 10) in the case of non-random missing pattern ($\kappa = 500$)

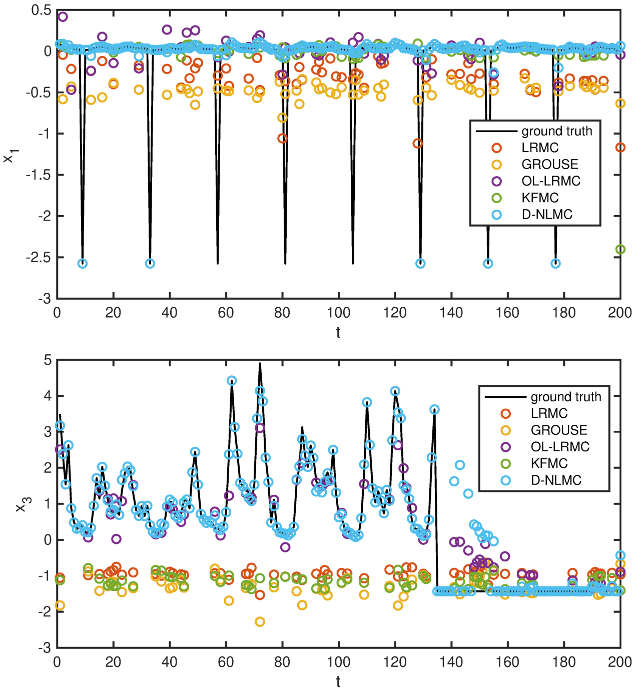

Figure 6: Examples of Air Quality data imputation (variables 1 and 3) in the case of random missing pattern ($\rho = 0.3$)

indoor temperature, relative humidity, and lightning. Since the variables have very different scales, we standardize them to have unit variances. Similar to Section , we also consider the random missing pattern and non-random missing pattern. One difference is that in the non-random case, the size of the missing blocks is reduced to 5×5 because the dimension of the data is much lower than the previous case. In D-NLMC, we set $w = 50$, $R = 25$, and $\mu = 1$.

Table 2 shows the recovery errors in the cases of different missing rate ρ and different block numbers κ. We see that the proposed method D-NLMC has much lower recovery error than other methods in all cases. The superiority of D-NLMC over OL-LRMC becomes more significant when the problem is harder. In addition, Figure 5 shows an example of the recovery performance in the case of non-randomly missing ($\kappa = 500$). The performance of D-NLMC is better than other methods visually. Additionally, Figure 5 indicates that part of the latent structure (related to x_{10}) of the data changes abruptly instead of smoothly, which is beyond the assumption made in this paper.

Air Quality Data

We consider the air quality dataset (De Vito et al. 2008) from the UCI machine learning repository[4]. The dataset contains 9358 samples of hourly averaged responses from an array of 5 metal oxide chemical sensors embedded in an Air Quality Chemical Multisensor Device. There are 13 variables, further standardized to have unit variances. In D-NLMC, we set $w = 50$, $R = 25$, and $\mu = 3$.

Similar to Section and Section , we also consider the random missing pattern and non-random missing pattern (with block size 3×5). Figure 6 shows an example of the imputation performance in the case of random missing ($\rho = 0.3$). We see that, many data points recovered by LRMC, GROUSE, OL-LRMC, and KFMC are far from the ground truth. Compared to these methods, our D-NLMC has better recovery performance intuitively. The average recovery errors over 20 repeated trials are reported in Table 3. The proposed method D-NLMC outperformed other methods in almost all cases. These results are consistent with the results in Table 1 and Table 2.

Conclusion

This paper has proposed a new method called D-NLMC for the missing value imputation of data with time-varying nonlinear latent structures. Compared to online low-rank matrix completion methods such as (Balzano, Nowak, and Recht 2010; Guo 2015), D-NLMC has much higher recovery accuracy in recovering the missing values of data with nonlinear structures. Compared to existing nonlinear matrix completion methods such as (Ongie et al. 2017; Fan, Zhang, and Udell 2020), D-NLMC can adapt to the changes of the latent structure of online data and has much higher recovery accuracy and much lower time cost.

In this study, although we focused only on missing value imputation, it is possible to extend D-NLMC to robust and dynamic subspace tracking (Vaswani et al. 2018), which can be a future work.

[4] https://archive.ics.uci.edu/ml/datasets/Air+Quality

Proof for Theorem 1

Proof. Without loss of generality, we assume that w is an odd number. We obtain

$$\begin{aligned}&\|g_t(\boldsymbol{z}_t) - g_{t-\frac{w-1}{2}}(\boldsymbol{z}_t)\| \\ \leq&\|g_t(\boldsymbol{z}_t) - g_{t-1}(\boldsymbol{z}_t)\| + \|g_{t-1}(\boldsymbol{z}_t) - g_{t-2}(\boldsymbol{z}_t)\| + \cdots \\ &+ \|g_{t-\frac{w-1}{2}+1}(\boldsymbol{z}_t) - g_{t-\frac{w-1}{2}}(\boldsymbol{z}_t)\| \\ \leq&\tfrac{w-1}{2}\gamma\|\boldsymbol{z}_t\|.\end{aligned} \tag{14}$$

Similarly, for $s = t - \frac{w-1}{2}, \ldots, t$, we have

$$\|g_s(\boldsymbol{z}_s) - g_{t-\frac{w-1}{2}}(\boldsymbol{z}_s)\| \leq (\tfrac{w-1}{2} + s - t)\gamma\|\boldsymbol{z}_s\|, \tag{15}$$

and for $s = t - w + 1, \ldots, t - \frac{w-1}{2} - 1$, we have

$$\|g_s(\boldsymbol{z}_s) - g_{t-\frac{w-1}{2}}(\boldsymbol{z}_s)\| \leq (t - s - \tfrac{w-1}{2})\gamma\|\boldsymbol{z}_s\|. \tag{16}$$

Putting (15) and (16) together, we get

$$\begin{aligned}&\sum_{s=t-w+1}^{t} \left\|g_s(\boldsymbol{z}_s) - g_{t-\frac{w-1}{2}}(\boldsymbol{z}_s)\right\|^2 \\ \leq&\sum_{s=t-w+1}^{t-\frac{w-1}{2}-1} (t - s - \tfrac{w-1}{2})^2\gamma^2\|\boldsymbol{z}_s\|^2 \\ &+ \sum_{s=t-\frac{w-1}{2}+1}^{t} (\tfrac{w-1}{2} + s - t)^2\gamma^2\|\boldsymbol{z}_s\|^2 \\ \leq&2\sum_{v=1}^{(w-1)/2} v^2\gamma^2c_t^2 \\ =&\gamma^2c_t^2(w-1)w(w+1)/12 \\ \leq&\gamma^2c_t^2w^3/12,\end{aligned} \tag{17}$$

where $c_t = \max(\|\boldsymbol{z}_{t-w+1}\|, \ldots, \|\boldsymbol{z}_t\|)$. Let

$$\hat{\boldsymbol{X}}_t = (\hat{\boldsymbol{x}}_{t-w+1}, \hat{\boldsymbol{x}}_{t-w+2}, \ldots, \hat{\boldsymbol{x}}_t),$$

where $\hat{\boldsymbol{x}}_s = g_{t-\frac{w-1}{2}}(\boldsymbol{z}_s)$, $s = t-w+1, \ldots, t$. According to Lemma 1 of (Fan, Zhang, and Udell 2020), with probability 1, we have

$$\text{rank}(\hat{\boldsymbol{X}}_t) \leq \min\left\{\tbinom{r+\theta}{\theta}, d, w\right\}. \tag{18}$$

On the other hand, according to (17) and the definition of $\hat{\boldsymbol{X}}_t$, we have

$$\|\boldsymbol{X}_t - \hat{\boldsymbol{X}}_t\|_F \leq \frac{\gamma c_t w^{1.5}}{3}. \tag{19}$$

Now combining (18) and (19), we conclude that $\boldsymbol{X}_t$ can be approximated by a matrix $\hat{\boldsymbol{X}}_t$ with rank at most $\min\left\{\tbinom{r+\theta}{\theta}, d, w\right\}$ and the approximation error is at most $\gamma c_t w^{1.5}/3$. This finished the proof for the first part of the theorem.

Let ϕ be a q-order polynomial feature map. According to Lemma 1 of (Fan, Zhang, and Udell 2020), we have

$$\text{rank}(\phi(\hat{\boldsymbol{X}}_t)) \leq \min\left\{\binom{r+\theta q}{\theta q}, \binom{d+q}{q}, w\right\}. \tag{20}$$

Then we conclude that $\boldsymbol{X}_t$ can be approximated by a matrix $\hat{\boldsymbol{X}}_t$ satisfying $\text{rank}(\phi(\hat{\boldsymbol{X}}_t)) \leq \min\left\{\tbinom{r+\theta q}{\theta q}, \tbinom{d+q}{q}, w\right\}$. This finished the proof. □

Gradient Related to Polynomial Kernels

Denote by $\mathcal{L}_t$ the objective function in (5) of the main paper. We have

$$\frac{\partial\mathcal{L}_t}{\partial\boldsymbol{K}_t} = \frac{p}{2}\boldsymbol{K}_t^{\frac{p}{2}-1} = \frac{p}{2}\boldsymbol{V}_t\boldsymbol{\Lambda}_t^{\frac{p}{2}-1}\boldsymbol{V}_t^{\top}, \tag{21}$$

where $\boldsymbol{V}_t$ and $\text{diag}(\boldsymbol{\Lambda}_t)$ are the eigenvectors and eigenvalues of $\boldsymbol{K}_t$ respectively. When $\boldsymbol{K}_t$ is computed by a polynomial kernel $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i^{\top}\boldsymbol{x}_j + a)^q$, we have

$$\begin{aligned}\frac{\partial\mathcal{L}_t}{\partial[\boldsymbol{x}_t]_{\bar{\omega}}} =&\sum_{i=1}^{w}\sum_{j=1}^{w}\frac{\partial\mathcal{L}_t}{\partial[\boldsymbol{K}_t]_{ij}}\frac{\partial[\boldsymbol{K}_t]_{ij}}{\partial[\boldsymbol{x}_t]_{\bar{\omega}}} \\ =&\left[2q\boldsymbol{X}_t\left(\boldsymbol{\alpha}\odot\left(\boldsymbol{X}_t^{\top}\boldsymbol{x}_t + a\right)^{\odot(q-1)}\right)\right]_{\bar{\omega}}\end{aligned} \tag{22}$$

where $\boldsymbol{\alpha} = \left[\frac{\partial\mathcal{L}_t}{\partial\boldsymbol{K}_t}\right]_{:w}$. Invoking (21) into (22), we arrive at

$$\frac{\partial\mathcal{L}_t}{\partial[\boldsymbol{x}_t]_{\bar{\omega}}}\left[2q\boldsymbol{X}_t\Big((\frac{p}{2}\boldsymbol{V}_t\boldsymbol{\Lambda}_t^{\frac{p}{2}-1}\boldsymbol{v}_t)\odot\left(\boldsymbol{X}_t^{\top}\boldsymbol{x}_t + a\right)^{\odot(q-1)}\Big)\right]_{\bar{\omega}} \tag{23}$$

where $\boldsymbol{v}_t$ denotes the last column of $\boldsymbol{V}_t^{\top}$.

Proof for Theorem 2

Proof. According to Corollary 1 of (Fan, Zhang, and Udell 2020), there exists a matrix $\tilde{\boldsymbol{K}} \in \mathbb{R}^{w\times w}$ with rank at most $\binom{r+\theta q}{\theta q}$ such that

$$\left\|\hat{\boldsymbol{K}}_\sigma - \tilde{\boldsymbol{K}}\right\|_F \leq C_1, \tag{24}$$

where $C_1 = w\exp\big(-\frac{\min_i\|\hat{\boldsymbol{x}}_i\|^2}{\sigma^2}\big)\frac{\max_i\|\hat{\boldsymbol{x}}_i\|^{q+1}}{\sigma^{2(q+1)}(q+1)!}$, provided that w/r is large enough. On the other hand, using the local Lipschitz continuity of exponential function, we have

$$\begin{aligned}&\|\boldsymbol{K}_\sigma - \hat{\boldsymbol{K}}_\sigma\|_F^2 \\ \leq&\frac{1}{4\sigma^4}\sum_{ij}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 - \|\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j\|^2\right)^2 \\ \leq&\frac{1}{4\sigma^4}\sum_{ij}C_{ij}\left(\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\| + \|\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j\|\right)^2 \\ \leq&\frac{\max_{ij}C_{ij}}{4\sigma^4}\sum_{ij}\left(\frac{w-1}{2}\gamma\|\boldsymbol{z}_i\| + \frac{w-1}{2}\gamma\|\boldsymbol{z}_j\|\right)^2 \\ \leq&\frac{\gamma^2w^2(w-1)^2\max_{ij}C_{ij}\max_i\|\boldsymbol{z}_i\|^2}{4\sigma^4},\end{aligned} \tag{25}$$

where $C_{ij} = 2\max(\|\boldsymbol{x}_i\|, \|\boldsymbol{x}_j\|, \|\hat{\boldsymbol{x}}_i\|, \|\hat{\boldsymbol{x}}_j\|)$. Combining (24) with (25), we obtain

$$\begin{aligned}&\|\boldsymbol{K}_\sigma - \tilde{\boldsymbol{K}}\|_F \\ \leq&\|\boldsymbol{K}_\sigma - \hat{\boldsymbol{K}}_\sigma\|_F + \left\|\hat{\boldsymbol{K}}_\sigma - \tilde{\boldsymbol{K}}\right\|_F \\ \leq&\frac{\gamma w^2 C_x C_z}{2\sigma^2} + \frac{wC_x'(C_x^2/2)^{q+1}}{\sigma^{2(q+1)}(q+1)!},\end{aligned} \tag{26}$$

where $C_x' = \exp\big(-\frac{\min_i\|\hat{\boldsymbol{x}}_i\|^2}{\sigma^2}\big)$, $C_x = \sqrt{2\max(\|\boldsymbol{x}_i\|, \|\boldsymbol{x}_j\|, \|\hat{\boldsymbol{x}}_i\|, \|\hat{\boldsymbol{x}}_j\|)}$, and $C_z = \max_i\|\boldsymbol{z}_i\|$.

Since g_t is polynomial, there exists a constant C_θ large enough such that $\max_i \|\hat{\boldsymbol{x}}_i\| \leq C_\theta \max_i \|\boldsymbol{z}_i\|$, where $i = t-w+1, \ldots, t$. Letting $C_t = \sqrt{2C_\theta}\,(\max_i \|\boldsymbol{z}_i\|)^{3/2}$ and $C_t' = \exp(-\frac{C_\theta^2(\max_i \|\boldsymbol{z}_i\|)^2}{\sigma^2})\,(2C_\theta \max_i \|\boldsymbol{z}_i\|)^{q+1}$. It follows from (26) that

$$\|\boldsymbol{K}_\sigma - \tilde{\boldsymbol{K}}\|_F \leq \frac{C_t \gamma w^2}{2\sigma^2} + \frac{C_t' w}{\sigma^{2(q+1)}(q+1)!}. \tag{27}$$

This finished the proof. □

Rank-One Modification for Fast EVD

Here we show how to perform rank-one modification (Brand 2006) twice to compute the eigenvalue decomposition of $\boldsymbol{K}_t$. Let $\boldsymbol{e}_w = [0, 0, \ldots, 0, 1]^\top$ and $\tilde{\boldsymbol{k}}' = [\boldsymbol{k}'^\top \; k(\boldsymbol{x}_t, \boldsymbol{x}_t)]^\top$. The method is detailed in Algorithm 4.

Algorithm 4: Rank-one modification for fast EVD of $\boldsymbol{K}_t$

```
Input: $\boldsymbol{V}'_{t-1}, \boldsymbol{\Lambda}'_{t-1}, \boldsymbol{e}_w, \boldsymbol{k}', \tilde{\boldsymbol{k}}'$
  $\boldsymbol{U} \leftarrow \boldsymbol{V}'_{t-1}, \boldsymbol{V} \leftarrow [\boldsymbol{V}'^\top_{t-1}\ \boldsymbol{0}]^\top, \boldsymbol{a} \leftarrow \bar{\boldsymbol{k}}', \boldsymbol{b} \leftarrow \boldsymbol{e}_w$
  $\boldsymbol{m} = \boldsymbol{U}^\top \boldsymbol{a}, \boldsymbol{p} = \boldsymbol{a} - \boldsymbol{U}\boldsymbol{m}, \bar{\boldsymbol{p}} = \boldsymbol{p}/\|\boldsymbol{p}\|.$
  $\boldsymbol{n} = \boldsymbol{V}^\top \boldsymbol{b}, \boldsymbol{q} = \boldsymbol{b} - \boldsymbol{V}\boldsymbol{n}, \bar{\boldsymbol{q}} = \boldsymbol{q}/\|\boldsymbol{q}\|.$
  $\boldsymbol{W} := \begin{bmatrix} \boldsymbol{\Lambda}'_{t-1} & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{m} \\ \|\boldsymbol{p}\| \end{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ \|\boldsymbol{q}\| \end{bmatrix}^\top.$
  $\boldsymbol{W} = \boldsymbol{U}'\boldsymbol{\Sigma}'\boldsymbol{V}'^\top.$
  $\bar{\boldsymbol{U}} \leftarrow \boldsymbol{U}\ \bar{\boldsymbol{p}}]\boldsymbol{U}', \bar{\boldsymbol{V}} \leftarrow \boldsymbol{V}\ \bar{\boldsymbol{q}}]\boldsymbol{V}'.$
  $\boldsymbol{U} \leftarrow [\bar{\boldsymbol{U}} \top\ \boldsymbol{0}]^\top, \boldsymbol{V} \leftarrow \bar{\boldsymbol{V}}, \boldsymbol{a} \leftarrow \boldsymbol{e}_w, \boldsymbol{b} \leftarrow \tilde{\boldsymbol{k}}'$
  $\boldsymbol{m} = \boldsymbol{U}^\top \boldsymbol{a}, \boldsymbol{p} = \boldsymbol{a} - \boldsymbol{U}\boldsymbol{m}, \bar{\boldsymbol{p}} = \boldsymbol{p}/\|\boldsymbol{p}\|.$
  $\boldsymbol{n} = \boldsymbol{V}^\top \boldsymbol{b}, \boldsymbol{q} = \boldsymbol{b} - \boldsymbol{V}\boldsymbol{n}, \bar{\boldsymbol{q}} = \boldsymbol{q}/\|\boldsymbol{q}\|.$
  $\boldsymbol{W} := \begin{bmatrix} \boldsymbol{\Sigma}' & \boldsymbol{0} \\ \boldsymbol{0} & 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{m} \\ \|\boldsymbol{p}\| \end{bmatrix} \begin{bmatrix} \boldsymbol{n} \\ \|\boldsymbol{q}\| \end{bmatrix}^\top.$
  $\boldsymbol{W} = \boldsymbol{U}'\boldsymbol{\Sigma}'\boldsymbol{V}'^\top.$
  $\boldsymbol{U}_t \leftarrow [\boldsymbol{U}\ \bar{\boldsymbol{p}}]\boldsymbol{U}', \boldsymbol{\Lambda}_t \leftarrow \boldsymbol{\Sigma}', \boldsymbol{V}_t \leftarrow [\boldsymbol{V}\ \bar{\boldsymbol{q}}]\boldsymbol{V}'.$
Output: $\boldsymbol{K}_t \approx \boldsymbol{V}_t \boldsymbol{\Lambda}_t \boldsymbol{V}_t^\top.$
```

Acknowledgements

The author gratefully acknowledges the support from the funding NO.T00120210002 of Shenzhen Research Institute of Big Data and the funding NO.AC01202101102 of Shenzhen Institute of Artificial Intelligence and Robotics for Society.

References

Afrifa-Yamoah, E.; Mueller, U. A.; Taylor, S.; and Fisher, A. 2020. Missing data imputation of high-resolution temporal climate time series data. *Meteorological Applications*, 27(1): e1873.

Alameda-Pineda, X.; Ricci, E.; Yan, Y.; and Sebe, N. 2016. Recognizing Emotions From Abstract Paintings Using Non-Linear Matrix Completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5240–5248.

Balzano, L.; Chi, Y.; and Lu, Y. M. 2018. Streaming pca and subspace tracking: The missing data case. *Proceedings of the IEEE*, 106(8): 1293–1310.

Balzano, L.; Nowak, R.; and Recht, B. 2010. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, 704–711.

Brand, M. 2003. Fast online svd revisions for lightweight recommender systems. In *Proceedings of the 2003 SIAM international conference on data mining*, 37–46. SIAM.

Brand, M. 2006. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and Its Applications*, 415(1): 20–30.

Candès, E. J.; and Recht, B. 2009. Exact Matrix Completion via Convex Optimization. *Foundations of Computational Mathematics*, 9(6): 717–772.

Chouvardas, S.; Abdullah, M. A.; Claude, L.; and Draief, M. 2017. Robust online matrix completion on graphs. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4019–4023. IEEE.

De Vito, S.; Massera, E.; Piga, M.; Martinotto, L.; and Di Francia, G. 2008. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2): 750–757.

Devooght, R.; Kourtellis, N.; and Mantrach, A. 2015. Dynamic matrix factorization with priors on unknown values. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 189–198.

Dhanjal, C.; Gaudel, R.; and Clémençon, S. 2014. Online matrix completion through nuclear norm regularisation. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, 623–631. SIAM.

Elhamifar, E. 2016. High-Rank Matrix Completion and Clustering under Self-Expressive Models. In *Advances in Neural Information Processing Systems 29*, 73–81.

Eriksson, B.; Balzano, L.; and Nowak, R. D. 2011. High-Rank Matrix Completion and Subspace Clustering with Missing Data. *CoRR*, abs/1112.5629.

Fan, J. 2021. Large-Scale Subspace Clustering via k-Factorization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 342–352.

Fan, J.; and Cheng, J. 2018. Matrix completion by deep matrix factorization. *Neural Networks*, 98: 34–41.

Fan, J.; and Chow, T. W. 2017. Matrix completion by least-square, low-rank, and sparse self-representations. *Pattern Recognition*, 71: 290 – 305.

Fan, J.; and Chow, T. W. 2018. Non-linear matrix completion. *Pattern Recognition*, 77: 378 – 394.

Fan, J.; Ding, L.; Chen, Y.; and Udell, M. 2019. Factor group-sparse regularization for efficient low-rank matrix recovery. In *Advances in Neural Information Processing Systems*, 5104–5114.

Fan, J.; and Udell, M. 2019. Online high rank matrix completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8690–8698.

Fan, J.; Zhang, Y.; and Udell, M. 2020. Polynomial matrix completion for missing data imputation and transductive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3842–3849.

Goldberg, A.; Recht, B.; Xu, J.; Nowak, R.; and Zhu, X. 2010. Transduction with Matrix Completion: Three Birds with One Stone. In *Advances in Neural Information Processing Systems 23*, 757–765. Curran Associates, Inc.

Gu, S.; Zhang, L.; Zuo, W.; and Feng, X. 2014. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2862–2869.

Guillemot, C.; and Meur, O. L. 2014. Image Inpainting : Overview and Recent Advances. *IEEE Signal Processing Magazine*, 31(1): 127–144.

Guo, X. 2015. Online robust low rank matrix recovery. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Halko, N.; Martinsson, P. G.; and Tropp, J. A. 2011. Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions. *SIAM Review*, 53(2): 217–288.

Hu, Y.; Zhang, D.; Ye, J.; Li, X.; and He, X. 2013. Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(9): 2117–2130.

Le Morvan, M.; Josse, J.; Moreau, T.; Scornet, E.; and Varoquaux, G. 2020. NeuMiss networks: differentiable programming for supervised learning with missing values. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 5980–5990. Curran Associates, Inc.

Li, C.-G.; and Vidal, R. 2016. A Structured Sparse Plus Structured Low-Rank Framework for Subspace Clustering and Completion. *IEEE Transactions on Signal Processing*, 64(24): 6557–6570.

Liu, D. C.; and Nocedal, J. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1): 503–528.

Lu, C.; Tang, J.; Yan, S.; and Lin, Z. 2014. Generalized Nonconvex Nonsmooth Low-Rank Minimization. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 4130–4137.

Narayanamurthy, P.; and Vaswani, N. 2018. Nearly optimal robust subspace tracking. In *International Conference on Machine Learning*, 3701–3709. PMLR.

Nie, F.; Huang, H.; and Ding, C. 2012. Low-rank Matrix Recovery via Efficient Schatten P-norm Minimization. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, 655–661. AAAI Press.

Ongie, G.; Willett, R.; Nowak, R. D.; and Balzano, L. 2017. Algebraic Variety Models for High-Rank Matrix Completion. In *Proceedings of the 34th International Conference on Machine Learning*, 2691–2700. PMLR.

Papadimitriou, S.; Sun, J.; and Faloutsos, C. 2005. Streaming Pattern Discovery in Multiple Time-Series. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, 697–708. VLDB Endowment. ISBN 1595931546.

Su, X.; and Khoshgoftaar, T. M. 2009. A Survey of Collaborative Filtering Techniques. *Adv. in Artif. Intell.*, 2009: 4:2–4:2.

Tripathi, R.; Mohan, B.; and Rajawat, K. 2017. Adaptive low-rank matrix completion. *IEEE Transactions on Signal Processing*, 65(14): 3603–3616.

Vaswani, N.; Bouwmans, T.; Javed, S.; and Narayanamurthy, P. 2018. Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery. *IEEE signal processing magazine*, 35(4): 32–55.

Vaswani, N.; and Narayanamurthy, P. 2018. Static and dynamic robust PCA and matrix completion: A review. *Proceedings of the IEEE*, 106(8): 1359–1379.

Wang, Z.; jun Lai, M.; Lu, Z.; Fan, W.; Davulcu, H.; and Ye, J. 2014. Rank-One Matrix Pursuit for Matrix Completion. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 91–99.

Wen, Z.; Yin, W.; and Zhang, Y. 2012. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 4(4): 333–361.

Xie, Y.; Gu, S.; Liu, Y.; Zuo, W.; Zhang, W.; and Zhang, L. 2016. Weighted Schatten p-norm minimization for image denoising and background subtraction. *IEEE transactions on image processing*, 25(10): 4842–4857.

Xu, L.; and Davenport, M. 2016. Dynamic matrix recovery from incomplete observations under an exact low-rank constraint. *Advances in Neural Information Processing Systems*, 29: 3585–3593.

Yozgatligil, C.; Aslan, S.; Iyigun, C.; and Batmaz, I. 2013. Comparison of missing value imputation methods in time series: the case of Turkish meteorological data. *Theoretical and applied climatology*, 112(1): 143–167.

Yu, H.-F.; Rao, N.; and Dhillon, I. S. 2015. High-dimensional time series prediction with missing values. *arXiv preprint arXiv:1509.08333*.