Dynamic Matrix Recovery

Ziyuan Chen
School of Mathematical Sciences, Center for Statistical Science,
Peking University, Beijing, China
Ying Yang
Academy of Mathematics and Systems Science,
Chinese Academy of Sciences, Beijing, China
Fang Yao *
School of Mathematical Sciences, Center for Statistical Science,
Peking University, Beijing, China

Abstract

Matrix recovery from sparse observations is an extensively studied topic emerging in various applications, such as recommendation system and signal processing, which includes the matrix completion and compressed sensing models as special cases. In this work, we propose a general framework for dynamic matrix recovery of low-rank matrices that evolve smoothly over time. We start from the setting that the observations are independent across time, then extend to the setting that both the design matrix and noise possess certain temporal correlation via modified concentration inequalities. By pooling neighboring observations, we obtain sharp estimation error bounds of both settings, showing the influence of the underlying smoothness, the dependence and effective samples. We propose a dynamic fast iterative shrinkage-thresholding algorithm that is computationally efficient, and characterize the interplay between algorithmic and statistical convergence. Simulated and real data examples are provided to support such findings.

Keywords: compressed sensing, local smoothing, low rank, matrix completion

*Ziyuan Chen and Ying Yang are the co-first authors, and contribute equally to this work. Fang Yao is the corresponding author, E-mail: fyaomath.pku.edu.cn. Fang Yao's research is partially supported by the National Key R&D Program of China (No. 2022YFA1003801, 2020YFE0204200), the National Natural Science Foundation of China (No. 12292981, 11931001), the LMAM and the Fundamental Research Funds for the Central Universities and the LMEQF. Ying Yang's research is partially supported by China Postdoctoral Science Foundation (No. 2022TQ0360, 2022M723334), the National Natural Science Foundation of China (No. 71988101) and the Guozhi Xu Posdoctoral Research Foundation.

1 Introduction

Matrix recovery from sparse observations has been extensively studied in a wide range of problems, such as signal processing (Li et al. 2019), recommendation system (Koren et al. 2009) and the quantum state tomography (Gross et al. 2010). The classical model is to link the response Y with the random matrix $X \in \mathbb{R}^{m_1 \times m_2}$ via

$$Y = \mathrm{Tr}(X^\top M^0) + \xi \tag{1}$$

to recover the low-rank matrix $M^0 \in \mathbb{R}^{m_1 \times m_2}$ from the independent observations $\{(X_i, Y_i)\}_{i=1}^n$, where $\mathrm{Tr}(\cdot)$ is the trace of a matrix, ξ is a mean zero random error and n denotes the sample size, which is typically much smaller than $m_1 m_2$ (i.e., $n \ll m_1 m_2$). By imposing different assumptions on X, the model includes a variety of matrix recovery models, in which matrix completion and compressed sensing are two important cases. Most existing works assume that the target matrix M^0 is static (Klopp 2011, Koltchinskii & Xia 2015, Fan et al. 2021, Xia & Yuan 2021). However, there are various scenarios where M^0 is dynamic and observed at multiple time points, such as in time-varying user interests in recommendation systems (Xu et al. 2020), temporally-changed graph models (Qiu et al. 2017) and dynamic quantum dot matrices (Csurgay et al. 2000). To formulate a general framework accounting the dynamic changes in numerous applications, we propose investigating the recovery of the dynamic matrix as follows,

$$Y_t = \mathrm{Tr}(X_t^\top M_t^0) + \xi_t,\ t = 1, \ldots, T, \tag{2}$$

where t is the timestamp, $\{Y_t, X_t, \xi_t\}$ is the tuple of the response, the design matrix and the noise at time t, respectively, and M_t^0 is a dynamic matrix which is assumed smooth across time t with a low-rank structure.

A prevalent line of research for solving the classical static model (1) is to impose strict

low-rank constraints based on matrix factorization. For instance, Keshavan et al. (2009) proposed singular value decomposition to make the recovery, Zhao et al. (2015) used QR decomposition instead, and Zheng & Lafferty (2016) adopted Burer-Monteiro factorization for rectangle matrix completion. Another stream is the convex approaches, thanks to Candès & Recht (2009) that first relaxed the non-convex rank restriction to nuclear norm for efficient optimization. A variety of convex algorithms and theoretical results are further established (Recht et al. 2010, Candes & Plan 2011, Agarwal et al. 2012). Though M_t^0 in (2) can be recovered by adopting the static matrix recovery at each time t, these *single-stage* estimates are less desirable, as they only use data at time t to reconstruct M_t^0 and fail to take advantage of the neighboring observations and smoothness of the underlying matrix. Related work also includes tensor regression to rebuild M_t^0, which views the dynamic model (2) from a static perspective by letting $N \in \mathbb{R}^{T\times m_1\times m_2}$ be a three order tensor whose (t,i,j)th element is the (i,j)th element of M_t^0. Gandy et al. (2011) and Liu et al. (2012) proposed algorithms for tensor completion and Zhang et al. (2020) considered the low-rank tensor trace regression via the importance sketching algorithm. We mention that their low-rank assumption of N is different from model (2) that M_t^0 is dynamically and smoothly evolving along time t. Consequently, one needs to process the whole data to recover M_t in tensor regression setting, which is time- and space-consuming.

A more efficient approach is to adopt the dynamic modeling/algorithm that avoids computing the whole data for update and has been adopted in some relevant problems. Lois & Vaswani (2015) designed a memory-saving algorithm for batch robust principle component analysis; Gao & Wang (2018) studied dynamic robust principal component analysis and proposed a fast iterative algorithm; Hu & Yao (2022) proposed a unified framework to directly estimate dynamic principal subspace in high dimension. The dynamic

algorithm for matrix recovery has been explored in Xu & Davenport (2016), which assumed that ξ_t are independently and identically distributed (i.i.d.) zero-mean Gaussian error and imposed a parametric assumption on structure of the underlying matrix, $M_t^0 = M_{t-1}^0 + \varepsilon_t$, where ε_t are i.i.d. zero-mean Gaussian noises. Another line of work, namely the dynamic tensors, also finds extensive applications in various domains, and several works have been noted in this direction. For instance, Zhang et al. (2021) proposed a time-varying coefficient model for temporal tensor factorization, employing a polynomial spline approximation, and Zhou et al. (2013) introduced a family of rank-R generalized linear tensor regression models. Most of the current research on dynamic tensors relies on tensor decomposition techniques and assumes that only certain decomposition factors or coefficients are changing (Zhang et al. 2021, Bi et al. 2021, Koren 2009, Wang et al. 2016).

Different from the existing work, we consider the general scenario where the dynamics are not limited to decomposition components or a parametric structure *priori*. There exist two major challenges. First, one needs to maintain the low-rank structure in the whole procedure of dynamic algorithm. Second, the combination of neighboring information often inflates the estimation error when the observations are correlated along time. To resolve these, we utilize local smoothing which has the advantages that the manifold structure of low-dimensional matrices can be approximately retained and temporal dependence has less influence when T increases. Based on these considerations, we propose a dynamic matrix recovery algorithm without imposing structural assumption besides smoothness and allowing the measurement error ξ_t and/or design matrix X_t to be possibly dependent across time. The idea is to pool together the observations that lie in the local window to estimate M_t^0. Noting that M_t^0 is low-rank, we set the target function composed of a locally weighted loss function and the nuclear norm penalty of M_t^0 and propose a Dynamic Fast

Iterative Shrinkage-Thresholding Algorithm (DFISTA) for optimization.

The main contributions of this work are summarized as follows. First, we establish a general framework for dynamic matrix recovery with theoretical guarantees. We allow the underlying matrix varying smoothly under the low-rank constraint and allow the observations to be dependent across time, which has more flexibility and applicability. While existing works only study the estimation error bounds for specific problems such as matrix completion and compressed sensing (Koltchinskii et al. 2011a, Negahban & Wainwright 2011, Xu & Davenport 2016, Fan et al. 2021), we derive the error bounds under this general framework, which can be readily adopted to such matrix recovery problems. Second, the proposed method attains a faster convergence rate than classical static methods in the following sense. Take the matrix completion for instance. The ratio of error bounds of the proposed estimator compared to the classical one (i.e., single-stage estimates at each t) is $O_p\big(\{m \log m/n_t\}^{-1/10} T^{-2/5}\big)$, where n_t is the number of observations at time t and $m = m_1 \vee m_2$ ("$\vee$" indicates the greater of two quantities). When the observations are sparse in the sense that $n_t \ll m \log m$, the ratio becomes $o_p\big(T^{-2/5}\big)$. Specifically, the dynamic method uses only $n' = \big(m \log m/n\big)^{-1/4} n/T = o(n/T)$ observations to attain the same convergence rate as the static method using n observations. Last but not least, a computationally efficient algorithm is devised for dynamic matrix recovery. This algorithm iterates only a subset of the data to update the estimate at each step. By using the estimate of the last step as the initial of the current optimization procedure, the algorithm converges faster than the single-stage recovery. Specifically, the ratio of computational complexities of our algorithm compared to the single-stage recovery is $O(T^{-3/5} \vee n^{-1/2})$.

The rest of the paper is organized as follows. In Section 2, we present the proposed method and algorithm for dynamic matrix recovery, and establish theoretical guarantees

in Section 3 for both independent and dependent observations with algorithmic analysis. Finally, we conduct simulation and real data experiments in Section 4 and 5 to show the favorable performance of the proposed dynamic method/algorithm compared with other existing methods. Additional theoretical results and technical proofs are deferred to an online Supplementary Material due for space economy.

2 Methodology and Algorithm

We consider the general framework in the form of dynamic trace regression whose coefficient matrix is changing smoothly along time. Specifically, at time $t = 1, \ldots, T$, we observe (Y_{ti}, X_{ti}) following model (2),

$$Y_{ti} = \mathrm{Tr}(X_{ti}^\top M_t^0) + \xi_{ti}, \quad i = 1, \ldots, n_t,$$

where M_t^0 is the time-varying coefficient matrix, X_{ti} and ξ_{ti} are design matrices and zero-mean noises, respectively, $\{X_{ti}, \xi_{ti}\}_i$ (given t) are mutually independent across subject and $\{X_{ti}, \xi_{ti}\}_t$ (given i) may be correlated across time. Parallel to the static model (1), the dynamic trace regression includes several problems such as dynamic matrix completion and dynamic compressed sensing that can be applied to dynamic recommendation systems and video signal processing as illustrated in numerical experiments of Section 5.

Example 1 *(Dynamic matrix completion). For each t, suppose that the design matrices X_{ti} are i.i.d. uniformly distributed on the set*

$$\mathcal{E} = \{e_j(m_1)e_k^\top(m_2) : 1 \le j \le m_1, 1 \le k \le m_2\}, \tag{3}$$

where $e_j(m)$ are the canonical basis vectors in $\mathbb{R}^m$ and then $\mathcal{E}$ forms an orthonormal basis in the space $\mathbb{R}^{m_1 \times m_2}$. Then estimating M_t^0 is equivalent to the problem of matrix completion under uniform sampling at random (USR).

Example 2 *(Dynamic compressed sensing). For each t, suppose that the design matrices X_{ti} are i.i.d. replicates of a random matrix X_t such that $\langle M, X_t\rangle$ is a sub-gaussian random variable for any $M \in \mathbb{R}^{m_1\times m_2}$. In this work, we focus on the concrete example that each design matrix X_{ti} is a random matrix whose elements are independent mean-zero sub-gaussian random variables with variance $\sigma^2_{X_t}$. Moreover, if the design X_t is homogeneous across t, we denote the variance as σ^2_X.*

We first briefly review the classical static method in the context of dynamic modeling, i.e., adopting the static method at each time t, which is referred to as the single-stage estimates in the sequel. Define the matrix inner product by $\langle M, X\rangle = \mathrm{Tr}(M^\top X)$. Koltchinskii et al. (2011a) proposed to incorporate the knowledge of the distribution of X_{ti} and estimate M_t^0 by

$$\widehat{M}_t^\lambda = \underset{M\in\mathbb{M}}{\mathrm{argmin}} \left[\frac{1}{n_t}\sum_{i=1}^{n_t} \mathbb{E}(\langle M, X_{ti}\rangle)^2 - \left\langle \frac{2}{n_t}\sum_{i=1}^{n_t} Y_{ti}X_{ti}, M \right\rangle + \lambda\|M\|_1 \right], \tag{4}$$

where $\mathbb{M} \subset \mathbb{R}^{m_1\times m_2}$ is a convex set of matrices, $\lambda > 0$ is a regularization parameter and $\|M\|_1$ is the nuclear norm of M. They proved that in the setting of matrix completion in Example 1, if λ is appropriately set, with probability at least $1 - 3/(m_1 + m_2)$,

$$(m_1m_2)^{-1/2}\|\widehat{M}_t^\lambda - M_t^0\|_2 \le C\left\{\frac{\log(m_1+m_2)\max(m_1,m_2)r_t}{n_t}\right\}^{1/2},$$

where $r_t = \mathrm{rank}(M_t^0)$ and $\|M\|_2$ is the Frobenius norm of matrix M. Though they have proved that the rate is optimal up to logarithmic factors comparing to the minimax lower bounds in classical trace regression, this approach fails to incorporate temporal information and hence is not efficient in multi-stage case.

A direct modification is to utilize the *two-step* procedure that first obtains the single-stage estimates $\widehat{M}_t^\lambda$ and then applies local smoothing to $\{\widehat{M}_t^\lambda\}_t$ to achieve the final estimate

as follows,

$$\widehat{M}_t^{\text{t-s}} = \sum_{j=1}^{T} \omega_h(j-t)\widehat{M}_j^{\lambda}, \tag{5}$$

where the local smoothing weights $\omega_h(j-t) = K_{Th}(j-t) / \sum_{k=1}^{T} K_{Th}(k-t)$ with the kernel function $K(x)$ and bandwidth h. The two-step estimation approach computes a weighted average of low-rank matrices that have been estimated at distinct time points. However, this method encounters a significant challenge when dealing with sparse observations. In such scenarios, the estimated matrix for each time point could potentially disrupt the true low-rank structure, displaying significant deviation from the actual value. Unfortunately, this disparity cannot be corrected through smoothing techniques.

Motivated by the observation that the manifold structure of low-rank matrices can be approximately preserved locally without matrix factorization, we design the dynamic target function with nuclear norm penalty, i.e.,

$$\widetilde{M}_t^{\lambda} = \underset{M\in\mathbb{M}}{\operatorname{argmin}} \sum_{j=1}^{T} \omega_h(j-t) \left[\frac{1}{n_j} \sum_{i=1}^{n_j} \mathbb{E}(\langle M, X_{ji}\rangle)^2 - 2 \left\langle \frac{1}{n_j} \sum_{i=1}^{n_j} Y_{ji} X_{ji}, M \right\rangle \right] + \lambda \|M\|_1, \tag{6}$$

which combines the observations across time and employs local smoothing for incorporating the temporal information. It is worth noting that in many applications including our Examples 1 and 2, the distribution of X_t is known, and yet this information has not been exploited in empirical risk. Thus we may utilize the distribution knowledge by substituting the empirical form with true values $\mathbb{E}\langle M, X_{ji}\rangle$. The temporal smoothness of M_t^0 guarantees that information in adjacent time points can be utilized to reduce the estimation error. We mention that the classical estimate $\widehat{M}_t^{\lambda}$ in (4) can be regarded as a special case of the proposed local estimate $\widetilde{M}_t^{\lambda}$ in (6) as the bandwidth $h \to 0$. By choosing h appropriately, our estimate pools more observations to reduce the estimation variance while controlling the bias caused by the varying of M_t^0, as discussed in detail in Section 3.

For implementation, we adopt the Fast Iterative Shrinkage-Thresholding Algorithm

(FISTA) which is an accelerated proximal method proposed by Beck & Teboulle (2009). It improves traditional proximal methods and terminates in $\mathcal{O}(\varepsilon^{-1/2})$ iterations with a ε-optimal solution. To handle with the trace norm penalty optimization problem (6), we use the matrix form FISTA established by Ji & Ye (2009) and Toh & Yun (2010). Define the objective function in (6) as $F_t(M) = f_t(M) + g(M)$, where

$$f_t(M) = \sum_{j=1}^{T} \omega_h(j-t) \left[\frac{1}{n_j} \sum_{i=1}^{n_j} \mathbb{E}(\langle M, X_{ji} \rangle)^2 - 2 \left\langle \frac{1}{n_j} \sum_{i=1}^{n_j} Y_{ji} X_{ji}, M \right\rangle \right].$$

We mention that the gradient function of f_t is equipped with Lipschitz constant L_f presented in Section 3.4, and that $g(M) = \lambda \|M\|_1$ is a continuous convex function. Then the matrix form FISTA can be applied to solve the dynamic trace regression problem

$$\widetilde{M}_t^{\lambda} = \arg\min_{M \in \mathbb{M}} f_t(M) + g(M).$$

Note that $\widetilde{M}_t^{\lambda}$ need to be calculated for a sequence $t = 1, 2, \cdots, T$. The algorithm is accelerated at time t by utilizing the estimate from the timestamp $(t-1)$. Specifically, suppose that at time t we apply the matrix form FISTA for k_t iteration times to numerically approximate the estimate $\widetilde{M}_t^{\lambda} = \arg\min_{M \in \mathbb{M}} f_t(M) + g(M)$ denoted by $M_t^{(k_t)}$. Then input $M_t^{(k_t)}$ as the initial matrix in the procedure of estimating $\widetilde{M}_{t+1}^{\lambda} = \arg\min_{M \in \mathbb{M}} f_{t+1}(M) + g(M)$ at time $(t+1)$. Such initialization method accelerates the algorithm, as demonstrated in Section 3.4. We summarize the implementations of the proposed DFISTA in Algorithm 1. In the algorithm, "svd" means calculating the single value decomposition components U, D, V for a matrix $G \in \mathbb{R}^{m_1 \times m_2}$ and "$(\cdot)_+$" means the threshold operator such as $(D - \gamma)_+ = \mathrm{diag}((\lambda_1 - \gamma) \vee 0, \cdots, (\lambda_{m_1} - \gamma) \vee 0)$ where $D = \mathrm{diag}(\lambda_1, \cdots, \lambda_{m_1})$. The learning rate s_k is updated according to Nesterov (1983).

Algorithm 1 Dynamic Low-Rank Matrix Recovery

Input: $\sum_{t=1}^{T} n_t$ matrices $X_{ti} \in \mathbb{R}^{m_1 \times m_2}$ and $\sum_{t=1}^{T} n_t$ scalars Y_{ti}, $t = 1, 2, \ldots, T$, $i = 1, 2, \cdots, n_j$, max iteration steps K and torrent `tor`.

Output: T matrices $M_t \in \mathbb{R}^{m_1 \times m_2},\ t = 1, 2, \ldots, T$.

Choose a initial matrix $M_0 \in \mathbb{R}^{m_1 \times m_2}$.

while $t \leq T$ **do**

  $M_t^{(-1)} \leftarrow M_{t-1},\ M_t^{(0)} \leftarrow M_{t-1}$;

  $s_0 \leftarrow 1,\ s_{-1} \leftarrow 1$;

  for $k = 0, 1, \ldots, K$ **do**

    $N_t^{(k)} \leftarrow M_t^{(k)} + \frac{s_{k-1}-1}{s_k}\left(M_t^{(k)} - M_t^{(k-1)}\right)$;

    $G_t^{(k)} \leftarrow N_t^{(k)} - L_f^{-1} \sum_{j=(t-Th/2)\vee 0}^{(t+Th/2)\wedge T} \frac{w_h(j-t)}{n_j} \sum_{i=1}^{n_j} X_{ji}\left(\langle X_{ji}, N_t^{(k)}\rangle - Y_j\right)$;

    $U, D, V \leftarrow \mathtt{svd}(G_t^{(k)})$;

    $M_t^{(k+1)} \leftarrow U(D - 2\lambda/L_f)_+ V^T$;

    $s_{k+1} \leftarrow \frac{1+\sqrt{1+4s_k^2}}{2}$;

    if $|F_t(M_t^{(k+1)}) - F_t(M_t^{(k)})| \leq \mathtt{tor}$ **then**

      break

    end if

  end for

  $M_t = M_t^{(k+1)}$;

end while

return $M_t, t = 1, 2, \cdots, T$

Unlike most factorization based algorithms that are challenging to analyze for their convergence properties, we are able to analyze the error bounds of the algorithm's results with respect to the true value. This analysis bridges the gap between the statistical estimator and the estimator obtained from the algorithm, as discussed in detail in Section 3.4.

3 Theoretical Guarantees

In this section, we study both the statistical properties of the estimation error $\|\widetilde{M}_t^{\lambda} - M_t^0\|_2$ and the algorithmic convergence of the proposed method. We first present the error bound of the estimate $\widetilde{M}_t^{\lambda}$ for general dynamic trace regressions. Then we derive the explicit expressions of the upper bound of $\|\widetilde{M}_t^{\lambda} - M_t^0\|_2$ when $\{X_t\}_t$ and $\{\xi_t\}_t$ are independent across t and when $\{X_t\}_t$ and $\{\xi_t\}_t$ are ϕ-mixing processes, respectively. Noting that X_t and/or ξ_t may be unbounded, the assumptions for classical concentration inequalities are violated (Merlevède et al. 2009, 2011, Hang & Steinwart 2017). We thus modify the Talagrand's inequalities and adopt truncation technique appropriately to derive the convergence rate for the dependent setting. In both scenarios, we shall explain how the number of time points T, the smoothness of M_t^0 and the dependence of X_t and ξ_t across time influence the estimation error explicitly.

In the sequel, we denote $\|M\|_0$, $\|M\|_1$, $\|M\|_2$, $\|M\|_\infty$ and $\langle M, N\rangle$ as the rank, nuclear norm, Frobenius norm, maximum singular value of M and the trace of $M^\top N$, respectively.

3.1 General error bounds

Let $\mathrm{vec}(X) \in \mathbb{R}^{m_1 m_2}$ be the vectorization of $X \in \mathbb{R}^{m_1 \times m_2}$ and define the second moment matrix of X_t by $\Sigma_t \in \mathbb{S}^{m_1 m_2 \times m_1 m_2}$ such that

$$\Sigma_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \mathbb{E}[\mathrm{vec}(X_{ti})\, \mathrm{vec}(X_{ti})^\top].$$

It is necesary to guarantee the identifiability of matrix recovery that the positive semi-definite matrix Σ_t is invertible. To simplify, we consider the homogeneous scenario such that at each time point t, the design $\{X_t\}_t$ have the same second moment matrix as stated below.

Assumption 1 *There exists a positive definite matrix* $\Sigma \in \mathbb{S}_+^{m_1 m_2 \times m_1 m_2}$ *with smallest eigenvalue* $\mu > 0$ *such that* $\Sigma_j = \Sigma,\ j = 1, 2, \ldots, T$.

We remark that in the heterogeneous situation, the convergence rate can be derived similarly, which is at the same order as the homogeneous setting with a more tedious presentation. An important indication of Assumption 1 is that, for all $M \in \mathbb{R}^{m_1 \times m_2}$ and $j = 1, 2, \cdots, T$,

$$\frac{1}{n_j} \sum_{i=1}^{n_j} \mathbb{E}(\langle M, X_{ji} \rangle)^2 = \operatorname{vec}(M)^\top \Sigma \operatorname{vec}(M) \geq \mu \|M\|_2^2,$$

which is a standard assumption in matrix recovery. The interpretation is that the smallest eigenvalue $\mu > 0$ determines the information contained in the design matrix X_t. Intuitively, the larger μ is, the more information is revealed by data, thus the more precise the estimator $\widetilde{M}_t^\lambda$ is. In most cases, μ can be calculated directly. For instances, in matrix completion of Example 1, the smallest eigenvalue of Σ is $\mu = 1/m_1 m_2$, and in compressed sensing of Example 2, μ is the smallest variance of elements in X_{ji}, which is equal to σ_X^2 under Assumption 1.

For brevity, we further denote $\Delta_j = 1/n_j \sum_{i=1}^{n_j} (Y_{ji} X_{ji} - \mathbb{E}(Y_{ji} X_{ji}))$ and

$$\delta_h M_t = \left\| M_t^0 - \sum_{j=1}^{T} \omega_h(j-t) M_j^0 \right\|_2, \quad \mathcal{W}_h \Delta_t = \left\| \sum_{j=1}^{T} \omega_h(j-t) \Delta_j \right\|_\infty.$$

The following theorem gives the general upper bound of estimation error $\|\widetilde{M}_t^\lambda - M_t^0\|_2$.

Theorem 1 *Under Assumption 1, if* $\lambda \geq 2\mathcal{W}_h \Delta_t$, *then*

$$\left\| \widetilde{M}_t^\lambda - M_t^0 \right\|_2 \leq \delta_h M_t + \min \left\{ \left((\delta_h M_t)^2 + \frac{2\lambda}{\mu} \left\| M_t^0 \right\|_1 \right)^{1/2}, \frac{1+\sqrt{2}}{2} \frac{\lambda}{\mu} \sqrt{r_t} \right\}. \tag{7}$$

When selecting $\lambda = 2\mathcal{W}_h \Delta_t$, *we have*

$$\left\| \widetilde{M}_t^\lambda - M_t^0 \right\|_2 \leq \delta_h M_t + \min \left\{ \left((\delta_h M_t)^2 + 2\mu^{-1} \mathcal{W}_h \Delta_t \left\| M_t^0 \right\|_1 \right)^{1/2}, (1+\sqrt{2}) \mu^{-1} \mathcal{W}_h \Delta_t \sqrt{r_t} \right\}. \tag{8}$$

The term $\delta_h M_t$ corresponds to the bias caused by kernel smoothing which is affected by kernel $K(x)$ and the bandwidth h, and $\mathcal{W}_h\Delta_t$ corresponds to the variance term caused by the effective samples in the local window and the measurement error ξ. When $h = 0$, the result in Theorem 1 coincides with the classical result of Corollary 1 in Koltchinskii et al. (2011a) such that when $\lambda = 2\|\Delta_t\|_\infty$,

$$\left\|\widehat{M}_t^\lambda - M_t^0\right\|_2 \leq \min\left\{\left(2\mu^{-1}\left\|\Delta_t\right\|_\infty \left\|M_t^0\right\|_1\right)^{1/2}, (1+\sqrt{2})\mu^{-1}\|\Delta_t\|_\infty\sqrt{r_t}\right\}. \tag{9}$$

Compared to (9), the bound (8) replaces the error term Δ_t by the smoothed version $\mathcal{W}_h\Delta_t$ at the cost of the bias $\delta_h M_t$.

To obtain an explicit expression of the error bound that may be used for bandwidth selection, we further investigate the properties of $\delta_h M_t$ and $\mathcal{W}_h\Delta_t$. Assumption 2 and 3 are standard conditions for temporal smoothness and the kernel function.

Assumption 2 *There exists a matrix function $M(t)$ supported on $[0,1]$, satisfying that*

1. *for any $j = 1, 2, \ldots, T$, $M_j^0 = M(j/T)$ and $\|M_j^0\|_1 \leq C_*$, $\max_{j,s,l}|[M_j^0]_{s,l}| \leq C_M$, for some constants C_* and C_M;*

2. *the first and second derivatives $\nabla M(t), \nabla^2 M(t)$ are continuous and bounded by some constants D_1 and D_2, respectively.*

Assumption 3 *The kernel function $K(x)$ is a symmetric probability density function on [-1, 1] and satisfies that*

$$\alpha(K) = \int x^2 K(x)\, dx < \infty, \quad R(K) = \int K(x)^2\, dx < \infty. \tag{10}$$

Define the $\psi-$norm $\|Z\|_{\psi(\eta)}$ for random variable Z with $0 < \eta < \infty$ as

$$\|Z\|_{\psi(\eta)} = \inf\left\{t > 0, \mathbb{E}\left[\exp\left(\frac{\|Z\|_\infty^\eta}{t^\eta}\right)\right] \leq 2\right\}$$

and $\|Z\|_{\psi(\infty)} = \inf\left\{t > 0, \mathbb{E}\mathbf{1}_{\{\|Z\|_\infty > t\}} = 0\right\}$. Then we impose the following assumption on the distributions of ξ_{ji} and X_{ji}.

Assumption 4 *The designs and noises at the same time point* $\{X_{ji}, \xi_{ji}\},\ i = 1, 2, \ldots, n_j$ *are mutually independent for each fixed* j *and there exists constants* $K_1, K_2 > 0,\ \alpha \geq 1,\ \beta \geq 2$ *that for any* $j = 1, 2 \ldots, T,\ i = 1, 2, \ldots, n_j$

1. ξ_{ji} *is sub-exponential distribution with* $\|\xi_{ji}\|_{\psi(\alpha)} \leq K_1$*;*

2. X_{ji} *is sub-gaussian distribution with* $\|X_{ji}\|_{\psi(\beta)} \leq K_2$*,*

3. *There exists* $\gamma \geq 1$ *such that* $1/\alpha + 1/\beta = 1/\gamma$*.*

The tail distributions of $\xi_{ji}X_{ji}$ and $\langle M, X_{ji}\rangle X_{ji}$ are both sub-exponential when X_{ji}, ξ_{ji} satisfy Assumption 4. Denote $\mathcal{D}_{\alpha,\beta,K_1,K_2}$ as the distribution set of $\{X_{ji}, \xi_{ji}\}_{ji}$ satisfying Assumption 4. We remark that $\mathcal{D}_{\alpha,\beta,K_1,K_2}$ is indeed general enough for applications. In the matrix completion problem of Example 1, X_{ji} are bounded with $\|X_{ji}\|_{\psi(\infty)} = 1$. Then the distributions of X_{ji}, ξ_{ji} lie in $\mathcal{D}_{1,\infty,K_1,1}$ as long as the noises ξ_{ji} follow sub-exponential distributions with $\alpha \geq 1$. In the compressed sensing problem of Example 2, from Theorem 4.4.5 in Vershynin (2018), we can set X_{ji} to be sub-guassian distributed with $\beta = 2$ and $K_2 \asymp (m_1 \vee m_2)^{1/2}\sigma_X$. Then the distributions of X_{ji}, ξ_{ji} lie in $\mathcal{D}_{2,2,K_1,\sigma_X}$ if noises ξ_{ji} follow a sub-guassian distributions with $\alpha = 2$.

Next we consider the temporal dependence between $\{X_{ji}, \xi_{ji}\}, i = 1, 2, \ldots, n_j$ and $\{X_{si}, \xi_{si}\}, i = 1, 2, \ldots, n_s$ that decays when the distance between time points j and s increases. To characterize the structure of such dependence, we impose the strongly dependent assumption that X_{ji} and ξ_{ji} are ϕ-mixing processes across j. Our definition of ϕ-mixing is the same as in Doukhan (2012), and the detail definition is given in Supplementary Material S.4. In the following subsections, we first derive the estimation error

bound in the ideal case that ξ_{ji} and X_{ji} are independent across time j, and then extend to the dependent case. These two settings are described in the following assumptions, respectively.

Assumption 5 *(Independent case) The designs and noises* $\{X_{ji}, \xi_{ji}\}$ *are mutually independent across* j *for each* i, $j = 1, 2, \ldots, T,\ i = 1, 2, \ldots, N_j$.

Assumption 6 *(ϕ-mixing processes) There exist two independent sequences of σ-fields $\mathcal{X}_j$ and $\mathcal{Y}_j, j \in \mathbb{N}^*$ such that*

$$\sigma((X_{j1}, X_{j2}, \ldots, X_{jn_j})^\top) \subseteq \mathcal{X}_j, \quad \sigma((\xi_{j1}, \xi_{j2}, \ldots, \xi_{jn_j})^\top) \subseteq \mathcal{Y}_j.$$

The σ-fields $\mathcal{X}_j$ and $\mathcal{Y}_j$, $j \in \mathbb{N}^*$ are both ϕ-mixing with coefficients $\phi_{\mathcal{X}}(k)$ and $\phi_{\mathcal{Y}}(k)$, $k \in \mathbb{N}$ satisfying that

$$\Phi_{\mathcal{X}} \triangleq \sum_{k=0}^{\infty} \sqrt{\phi_{\mathcal{X}}(k)} < \infty, \quad \Phi_{\mathcal{Y}} \triangleq \sum_{k=0}^{\infty} \sqrt{\phi_{\mathcal{Y}}(k)} < \infty.$$

3.2 Independent case

The theorem below states the upper bound of $\|\widetilde{M}_t^{\lambda} - M_t^0\|_2$ under the independent case when the distributions of X_{ji}, ξ_{ji} belong to $\mathcal{D}_{\alpha,\beta,K_1,K_2}$. For conciseness, we assume that the numbers of observations $n_j, j = 1, 2, \cdots, T$ have the same order denoted by n, i.e., $n_j \asymp n$ for $j = 1, 2, \cdots, T$. The conclusions for n_j of different orders can be derived similarly. In the following, the notations $a_n \ll b_n$, $a_n \gg b_n$ and $a_n \asymp b_n$ mean that the ratio a_n/b_n approaches to 0, infinity and a constant, respectively, and "$\lceil \cdot \rceil$" denotes the ceiling function.

Theorem 2 *Under Assumption 1–5, let* $n\lceil Th \rceil \gg (K_*/\sigma_*)^2 \log(m_1 + m_2)$, $h \to 0$ *and* $(m_1 m_2)^{1/2} \lceil T^2 h^2 \rceil T^{-1} \to \infty$ *as* $n, m_1, m_2, T \to \infty$, *when*

$$\lambda = 2C_1 \sigma_* \sqrt{\frac{\log(m_1 + m_2)}{n\lceil Th \rceil}},$$

where C_1 and σ_* are defined in Lemma S.4 of Supplementary Material, then with probability at least $1 - 3/(m_1 + m_2)$,

$$(m_1 m_2)^{-1/2} \left\| \widetilde{M}_t^{\lambda} - M_t^0 \right\|_2 \leq \frac{1}{2}\alpha(K) D_2 h^2 + \left(1 + \sqrt{2}\right) C_1 \sigma_* \left(\frac{r_t \log(m_1 + m_2)}{\mu^2 m_1 m_2 n \lceil Th \rceil} \right)^{1/2} + o(h^2), \tag{11}$$

where $\alpha(K)$ is defined in (10) *and D_2 in Assumption 2.*

We remark that $n\lceil Th \rceil$ is the number of samples used in a local window and the condition $n\lceil Th \rceil \gg (K_*/\sigma_*)^2 \log(m_1 + m_2)$ is the requirement for the effective sample size to guarantee the upper bound (11) tending to zero. The error bound (11) is a trade-off between the bias and variance which is controlled by the bandwidth h. When $n\mu^2 m_1 m_2/(T^4 \sigma_*^2 r_t \log(m_1 + m_2)) = o(1)$, we can choose the optimal bandwidth

$$h = C_h \left(\frac{\sigma_*^2 r_t \log(m_1 + m_2)}{\mu^2 m_1 m_2 n T} \right)^{1/5}, \tag{12}$$

where $C_h = \left[(2 + 2\sqrt{2}) C_1/(\alpha(K) D_2)\right]^{2/5}$ is a constant. When $n\mu^2 m_1 m_2/(T^4 \sigma_*^2 r_t \log(m_1 + m_2)) = \mathcal{O}(1)$, we choose $h = 0$ which is degenerate to the classical situation with the convergence rate $n^{-1/2}$.

We focus on a large enough value for T in the subsequent discussion, which allows us to omit the ceiling operator $\lceil \cdot \rceil$. For situations where T is small, we can simply choose $h = 0$, which yields the same results as static methods. With an appropriately selected h, we have the following corollary.

Corollary 1 *Under assumptions of Theorem 2, when*

$$nT \gg \max\{K_*^{5/2} \sigma_*^{-3} (\mu^2 m_1 m_2)^{1/4} \log(m_1 + m_2),\ \sigma_*^2 (\mu^2 m_1 m_2)^{-1} \log(m_1 + m_2)\},$$

with probability at least $1 - 3/(m_1 + m_2)$,

$$(m_1 m_2)^{-1/2} \left\| \widetilde{M}_t^{\lambda} - M_t^0 \right\|_2 \leq C_2 \left(\frac{\sigma_*^2 r_t \log(m_1 + m_2)}{\mu^2 m_1 m_2 n T} \right)^{2/5}, \tag{13}$$

where $C_2 = 1/2\left[(2+2\sqrt{2})C_1\right]^{4/5}\left[\alpha(K)D_2\right]^{1/5}$.

The bound requirement for nT is composed of two terms: the first is to make $nTh \gg (K_*/\sigma_*)^2 \log(m_1+m_2)$ hold when h takes the form (12), and the second is to guarantee the optimal bandwidth (12) tends to 0 as $n, T \to \infty$. In (13), the upper bound is proportional to $(nT)^{-2/5}$ where nT is the total sample size. This indicates that our dynamic method makes efficient use of information from adjacent time points to improve the estimation quality. The number of observations in a single time point n and the number of time points T complement each other to reduce the upper bound (13). When the samples are extremely sparse at each time point, it is infeasible for the classical static method to reconstruct the eigenvalues and eigenvectors of M_t^0. However, the proposed method can still control the estimation error $\|\widetilde{M}_t^\lambda - M_t^0\|_2$ to a desirable level, as long as there are sufficient time points. We also mention that the two-step smoothing method performs poorly, because the first-step estimate $\widehat{M}_t^\lambda$ and the reconstruction of eigen-subspace at each time point are far from satisfaction, which cannot be fixed by the second smoothing step regardless how dense the time points are. We emphasize that Theorem 2 and Corollary 1 can be applied directly to any dynamic trace regression problems as long as X_{ji} and ξ_{ji} lie in $\mathcal{D}_{\alpha,\beta,K_1,K_2}$. Taking dynamic matrix completion for example, we have the following corollary. The corresponding result for dynamic compressed sensing can be derived similarly, which is presented in Corollary S.3 of the Supplement.

Corollary 2 *Under Assumption 1–5, when X_{ji} are i.i.d. uniformly distributed on $\mathcal{E}$, ξ_{ji} are independently follow sub-exponential mean-zero distributions, $nTh \gg (m_1 \wedge m_2)\log^{1+2/\alpha}(m_1+m_2)$, and $(m_1m_2)^{1/2}Th^2 \to \infty$ as $h \to 0, n, m_1, m_2, T \to \infty$, then with probability at least*

$1 - 3/(m_1 + m_2)$,

$$(m_1 m_2)^{-1/2} \left\| \widetilde{M}_t^{\lambda} - M_t^0 \right\|_2$$
$$\leq \frac{1}{2}\alpha(K) D_2 h^2 + (1+\sqrt{2}) C_1 (C_M \vee \sigma_\xi) \left(\frac{r_t (m_1 \vee m_2) \log(m_1 + m_2)}{nTh} \right)^{1/2} + o(h^2).$$

When $nT \gg (m_1 \vee m_2) \log^{1+5/(2\alpha)}(m_1 + m_2)$, *let*

$$h = C_h \left(\frac{(C_M \vee \sigma_\xi)^2 r_t (m_1 \vee m_2) \log(m_1 + m_2)}{nT} \right)^{1/5}, \tag{14}$$

then

$$(m_1 m_2)^{-1/2} \left\| \widetilde{M}_t^{\lambda} - M_t^0 \right\|_2 \leq C_2 (C_M \vee \sigma_\xi)^{4/5} \left(\frac{r_t (m_1 \vee m_2) \log(m_1 + m_2)}{nT} \right)^{2/5}, \tag{15}$$

where C_1, C_h, C_2 *are the same constants as above.*

Recall that the classical static result for the error bound of matrix completion in Koltchinskii et al. (2011a) is

$$(m_1 m_2)^{-1/2} \left\| \widehat{M}_t^{\lambda} - M_t^0 \right\|_2 \leq (1+\sqrt{2}) C_1 (C_M \vee \sigma_\xi) \left(\frac{r_t (m_1 \vee m_2) \log(m_1 + m_2)}{n} \right)^{1/2}.$$

The dynamic method provides a sharper upper bound than the classic one. In other words, when the sample size in each single time point is sparse, our method can borrow information from adjacent time points to improve the estimation, as long as the number of time points T is sufficiently large.

3.3 Dependent case

In parallel to the independent case, we derive the bounds of $\|\widetilde{M}_t^{\lambda} - M_t^0\|_2$ when ξ_{ji}, X_{ji} are ϕ-mixing processes, as stated in Theorem 3 below.

Theorem 3 *With Assumption 1–4 and 6, when* $Th \gg \log(m_1+m_2)$, $nTh \gg (K_*/\sigma_*)^2 \log^3(m_1+m_2)$, $h \to 0$ *and* $(m_1m_2)^{1/2}Th^2 \to \infty$ *as* $n, m_1, m_2, T \to \infty$, *with probability at least* $1 - 3/(m_1+m_2)$,

$$(m_1m_2)^{-1/2} \left\| \widetilde{M}_t^{\lambda} - M_t^0 \right\|_2 \leq \frac{1}{2}\alpha(K)D_2h^2 + (1+\sqrt{2})\mathcal{C}_1(\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})\sigma_* \left(\frac{r_t \log(m_1+m_2)}{\mu^2 m_1 m_2 nTh} \right)^{1/2} + o(h^2) \tag{16}$$

where $\mathcal{C}_1 > 0$ *is a constant defined in Lemma S.5 of Supplementary Material.*

We use $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ to measure the temporal dependence of design matrices X_{ji} and noises ξ_{ji}, respectively, where larger $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ mean stronger dependence. When $\Phi_{\mathcal{X}} = \Phi_{\mathcal{Y}} = 1$, it indicates that X_{ji}, ξ_{ji} are independent across j and the bound (16) is the same as (11) in Theorem 2 in the independent case. When $nT \gg (\mu^2 m_1 m_2)^{-1}\sigma_*^2 \log(m_1+m_2)$, if we choose

$$h = \mathcal{C}_h \left(\frac{\sigma_*^2(\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})^2 r_t \log(m_1+m_2)}{\mu^2 m_1 m_2 nT} \right)^{1/5}, \tag{17}$$

where $\mathcal{C}_h = \left[(2+\sqrt{2})\mathcal{C}_1/(\alpha(K)D_2)\right]^{2/5}$, then we have the following corollary.

Corollary 3 *Under assumptions of Theorem 3, when*

$$n^{-1}T^4 \gg \sigma_*^{-2}\mu^2 m_1 m_2 \log^4(m_1+m_2)$$

and

$$nT \gg \max\{K_*^{5/2}\sigma_*^{-3}(\mu^2 m_1 m_2)^{1/4}\log(m_1+m_2),\ \sigma_*^2(\mu^2 m_1 m_2)^{-1}\log(m_1+m_2)\},$$

with probability at least $1 - 3/(m_1+m_2)$,

$$(m_1m_2)^{-1/2} \left\| \widetilde{M}_t^{\lambda} - M_t^0 \right\|_2 \leq \mathcal{C}_2 \left(\frac{\sigma_*^2(\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})^2 r_t \log(m_1+m_2)}{\mu^2 m_1 m_2 nT} \right)^{2/5}, \tag{18}$$

where $\mathcal{C}_2 = 1/2\left[(2+\sqrt{2})\mathcal{C}_1\right]^{4/5}\left[\alpha(K)D_2\right]^{1/5}$.

When X_{ji} and/or ξ_{ji} have strong long-term dependence across j, $\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}}$ becomes large and inflates the bound (18), which means that the convergence rate of $\widetilde{M}_t^{\lambda}$ is usually slower than that in independent case. Note that the bound for the single-stage estimates at each t using static method can also be derived from Theorem 2 by using a degenerated kernel $\delta_0(x)$, given by

$$(m_1 m_2)^{-1/2} \left\| \widehat{M}_t^{\lambda} - M_t^0 \right\|_2 \leq (1+\sqrt{2}) C_1 \sigma_* \left(\frac{r_t \log(m_1+m_2)}{\mu^2 m_1 m_2 n} \right)^{1/2}.$$

Then when

$$\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}} \ll T^{1/2} \left(\frac{\sigma^{*2} r_t \log(m_1+m_2)}{\mu^2 m_1 m_2 n} \right)^{1/8}, \tag{19}$$

our dynamic method yields a sharper bound than the static one though strong dependence across time exists. With the expression of h in (17), the condition (19) is equal to $Th \gg (\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})^2$. Note that Th corresponds to the effective time points used in estimation at each t. This implies that as the number of effective time points increases, the negative effect caused by temporal correlation would be offset to some extent. For space economy, the applications of Theorem 3 and Corollary 3 to dynamic matrix completion and dynamic compressed sensing are presented as Corollary S.2 and Corollary S.4 in Supplementary Material.

3.4 Empirical error and complexity analysis

Recall that the sequence generated by the iteration of Algorithm 1 at time t are denoted by $\{M_t^{(k)}\}_k, k \in \mathbb{N}^*$, where $M_t^{(k)}$ is the output matrix after k-th iteration and $M_t^{(0)}$ is the initial input matrix. In practice, we use the output $M_t^{(k)}$ to approximate M_t^0 for a certain k. Thus, we derive the upper bound of the empirical error $\left\| M_t^{(k)} - M_t^0 \right\|_2$ in the following theorem, which takes both the algorithmic error between $M_t^{(k)}$ and $\widetilde{M}_t^{\lambda}$ and the estimation error between $\widetilde{M}_t^{\lambda}$ and M_t^0 into account.

Theorem 4 *Under assumptions of Theorem 1, for any $k > 1$ and $\lambda \geq 2\|\mathcal{W}_h \Delta_t\|_\infty$, we have*

$$\left\| M_t^{(k)} - M_t^0 \right\|_2 \leq \delta_h M_t + \min \left\{ \left((\delta_h M_t)^2 + \frac{2\lambda}{\mu} \|M_t^0\|_1 + \frac{2L_f \gamma_t^2}{\mu (k+1)^2} \right)^{1/2}, \right.$$
$$\left. \frac{1+\sqrt{2}}{2} \frac{\lambda}{\mu} \sqrt{r_t} + \left(\frac{2L_f \gamma_t^2}{\mu (k+1)^2} \right)^{1/2} \right\}$$

where $\gamma_t = \left\| M_t^{(0)} - \widetilde{M}_t^\lambda \right\|_2$ and $L_f = 2 \left\| \sum_{j=1}^T \omega_h(j-t) \cdot \frac{1}{n_j} \sum_{i=1}^{n_t} \left(\|X_{ji}\|_2 X_{ji} \right) \right\|_2$ is the Lipschitz constant of gradient function $\nabla f_t(M)$ that is finite under Assumption 4.

The empirical error bound in Theorem 4 has an additional term $2L_f \gamma_t^2 / \mu (k+1)^2$ compared to the estimation error bound (7) in Theorem 1, which is the so-called algorithmic error. The optimal choice is to stop iterating when the algorithmic error attains the same order as the estimation error. Let k_t be the minimal number of iteration steps needed to attain the desired order. From the expression of the algorithmic error, k_t depends on the distance between the initial value $M_t^{(0)}$ and the target $\widetilde{M}_t^\lambda$, denoted by γ_t. Since less iterations are required for a smaller initial distance, the naive strategy to initialize $M_t^{(0)}$ randomly at each time t would lead a large γ_t. Noting that $\widetilde{M}_t^\lambda$ is close to $\widetilde{M}_{t-1}^\lambda$, we thus make use of the neighboring information in the initialization to reduce computation cost. Hence we propose to use the output matrix at time $(t-1)$ as the initial value for estimating $\widetilde{M}_t^\lambda$ at time t, i.e. $M_t^{(0)} = M_{t-1}^{(k_{t-1})}$. Take the dependent case in Section 3.3 for instance. Recall that the optimal estimation error bound is derived in Corollary 3 with appropriate choice of the bandwidth h. Then desired empirical error bound for Theorem 4 can be derived by

$$(m_1 m_2)^{-1/2} \left\| M_t^{(k_t)} - M_t^0 \right\|_2 \leq 2\mathcal{C}_2 \left(\frac{\sigma_*^2 (\Phi_\mathcal{X} \vee \Phi_\mathcal{Y})^2 r_t \log(m_1 + m_2)}{\mu^2 m_1 m_2 n T} \right)^{2/5}. \tag{20}$$

To quantify the improvement of our initialization strategy, in Corollary 4, we compute the total number of iteration steps to attain the bound (20) at each t, and compare with that of

the random initialization. The result for independent case can be derived similarly, which is presented in Corollary S.5 of Supplementary Material for space economy.

Corollary 4 *Under the assumptions of Theorem 3, when the total number of iterations of random initialization K_0 satisfies*

$$K_0 \geq \mathcal{C}_2^{-1} \left(\frac{\sigma_*^2 (\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})^2 r_t \log(m_1 + m_2)}{\mu^2 m_1 m_2} \right)^{-2/5} \left(\frac{2L_f}{\mu m_1 m_2} \right)^{1/2} \gamma_1 n^{2/5} T^{7/5},$$

or total number of iterations of the proposed initial strategy K_1 satisfies

$$\begin{aligned} K_1 \geq & \mathcal{C}_2^{-1} \left(\frac{\sigma_*^2 (\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})^2 r_t \log(m_1 + m_2)}{\mu^2 m_1 m_2} \right)^{-2/5} \left(\frac{2L_f}{\mu m_1 m_2} \right)^{1/2} (\gamma_1 + D_1)(nT)^{2/5} \\ & + 3\sqrt{2} \left(\frac{L_f}{\mu} \right)^{1/2} T. \end{aligned}$$

the empirical error is lower than the optimal upper bound (20) *for each $t = 1, \ldots, T$.*

From Corollary 4, once the total number of iterations of random initialization attains the level $\mathcal{O}(n^{2/5}T^{7/5})$ and that of Algorithm 1 attains $\mathcal{O}(n^{2/5}T^{2/5}+T)$, the error is bounded by (20). Then the ratio of computational cost of our proposed algorithm compared to the random initialization method is $\mathcal{O}(T^{-1} \vee n^{-2/5}T^{-2/5})$. We finally mention that the classical static method need $\mathcal{O}(n^{1/2})$ iterations to attain the optimal rate $\mathcal{O}(n^{-1/2})$ for each t, and hence the single-stage method (at all T times) need $\mathcal{O}(n^{1/2}T)$ iterations in total. Therefore the ratio of computational cost of Algorithm 1 compared to the single-stage method is $\mathcal{O}(T^{-3/5} \vee n^{-1/2})$, which leads to the conclusion that our algorithm is computationally more efficient than the random initial method and the classical static methods.

4 Simulation Studies

We take the dynamic matrix completion as the instance to verify the usefulness of our method. To demonstrate the advantages, we compare the proposed estimates with three

benchmark methods: (i) the classical static estimator $\widehat{M}_t^{\lambda}$ defined as in (4), abbreviated as Static; (ii) the two-step smoothing estimator $\widehat{M}_t^{\text{t-s}}$ given in (5), abbreviated as TwoStep; (iii) the low-rank tensor completion estimator, abbreviated as Tensor,

$$\widehat{N}^{\lambda} = \underset{N \in \mathbb{R}^{T \times m_1 \times m_2}}{\operatorname{argmin}} \|\mathcal{P}_{\Omega}(N) - Y\|_2^2 + \lambda \sum_{i=1}^{3} \|N_{(i)}\|_1,$$

where $\mathcal{P}_{\Omega}$ is the projection to the observation subspace Ω, Y is the observed data tensor with element

$$[Y]_{j,s,l} = \begin{cases} \dfrac{1}{|\mathcal{A}_{j,s,l}|} \displaystyle\sum_{i \in \mathcal{A}_{j,s,l}} Y_{ji}, & \text{if } \mathcal{A}_{j,s,l} \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

for $\mathcal{A}_{j,s,l} = \left\{ i \mid X_{ji} = e_s(m_1) e_l^{\top}(m_2), 1 \leq i \leq n_j \right\}$, and $N_{(i)}$ is the mode-i unfolding of tensor N which is a matrix that arranges the mode-i fibers to be columns. We use matrix form FISTA algorithm in Toh & Yun (2010) to approximate $\widehat{M}_t^{\lambda}$, and $\widehat{M}_t^{\text{t-s}}$ has a closed form solution. In Tensor, we use AMD-TR(E) algorithm in Gandy et al. (2011) to numerically approximate $\widehat{N}^{\lambda}$.

We set $m_1 = 500, m_2 = 300, r_t = 10$ and the matrix function $M(t)$ in Assumption 2 is constructed by $M(t) = U(t)D(t)V^T(t)$ with

$$\begin{aligned} U(t) &= \cos\left(\frac{t\pi}{2}\right) U_0 + \sin\left(\frac{t\pi}{2}\right) U_1, \\ D(t) &= 10 \left(\operatorname{diag}\left\{10^2, 9^2, \ldots, 1\right\} + t \operatorname{diag}\left\{10, 9, \ldots, 1\right\} \right), \\ V(t) &= \cos\left(\frac{t\pi}{2}\right) V_0 + \sin\left(\frac{t\pi}{2}\right) V_1, \end{aligned}$$

where $U_0, U_1 \in \mathbb{R}^{m_1 \times r_t}$, $V_0, V_1 \in \mathbb{R}^{m_2 \times r_t}$ are composed of standard orthogonal basis. Here we consider the situation that each n_t is the same, i.e., $n_t = n, t = 1, 2, \cdots, T$. Denote the rate of observation samples as $\rho = n/(m_1 m_2)$ and the density of time points as $\tau = 1/T$. We use a 5-fold cross validation to select λ in the numerical experiments. Please refer to Section S.9.4 of the Supplement for more details.

For the independent case, we sample $X_{ji},\ i = 1, 2, \ldots, n$ from $\mathcal{E}$ defined in (3) independently with uniform probabilities, and let $\xi_{ji}, j = 1, 2, \ldots, T,\ i = 1, 2, \ldots, n$ i.i.d follow the normal distribution $N(0, \sigma_\xi^2)$, where $T = 100$ and $\sigma_\xi = 1$. We use the Epanechnikov kernel and set the bandwidth h by the optimal selection form (14) in which we approximate $C_M \vee \sigma_\xi$ by mean of the top 10% largest Y_{ji} and D_2 by $\sum_{j=2}^{T} |1/n_j \sum_{i=1}^{n_j} Y_{ji} - 1/n_{j-1} \sum_{i=1}^{n_{j-1}} Y_{j-1i}|$. We set $\rho = 0.2$ for our proposed dynamic low-rank (DLR) and Tensor methods. In Static and TwoStep, $\widehat{M}_t^{\lambda}$ and $\widehat{M}_t^{\text{t-s}}$ fail to recover the matrix structure under the same setting $\rho = 0.2$, so we set $\rho = 0.8$ for feasibility. Figure 1 illustrates the mean square errors

$$\text{MSE}_t = (m_1 m_2)^{-1} \left\| \widehat{M}_t - M_t^0 \right\|_2^2, \tag{21}$$

for $t = 1, 2, \cdots, T$. For the proposed DLR and TwoStep methods, the MSE_t floats up at the boundaries due to slight edge effect. The average MSE, i.e., $T^{-1} \sum_{t=1}^{T} \text{MSE}_t$ for the DLR estimate is 0.085 and for Tensor is 0.44 when $\rho = 0.2$. This implies that the tensor recovery is not effective at least when the temporal structure is smooth. For Static and TwoStep, when $\rho = 0.8$, the average MSE is 0.16 and 0.13 respectively. That is, these two methods with four times data still perform worse than the proposed DLR method. This verifies the theoretical finding that when the sample size n at a single time point is small, the classical trace regression fails to estimate M_t^0, while our dynamic method can borrow adjacent information to improve the estimation quality.

Now we study the empirical influence of ρ and τ on the estimation results. One can see from (15) that given the underlying matrix, the bound of MSE is in proportion to $(\rho/\tau)^{-4/5}$. We set eight different values for (ρ, τ) which can be divided into three groups, i.e. $\rho/\tau = 5, 10, 20$. The MSE_t across t under these settings are shown in the left panel of Figure 2, which indicates that given the matrix dimension, the empirical MSE depends on ρ/τ. We further plot the logarithm of average MSE across time versus the logarithm of

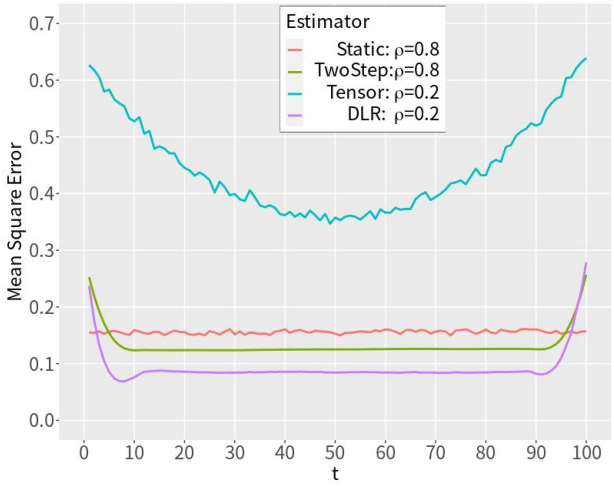

Figure 1: The comparisons of MSE_t for different methods. In DLR and Tensor, $\rho = 0.2$, and in Static and TwoStep, $\rho = 0.8$.

ρ/τ in the right panel of Figure 2, which reveals the linear relationship of the slope -4/5 between them. This further validates the conclusion (15).

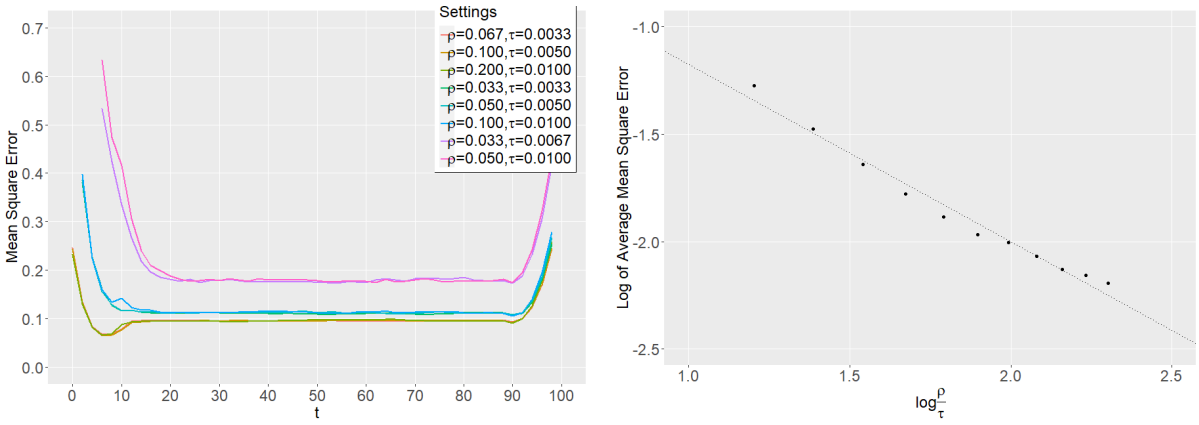

Figure 2: The left panel corresponds to MSE_t of DLR estimates under different settings of ρ and τ. The curves are clustered into three classes corresponding to $\rho/\tau = 5, 10, 20$. The right panel shows the logarithm of average MSE versus the logarithm of ρ/τ showing a linear trend with the slope -4/5.

We also investigate the influence of the dependent noise on the proposed estimator. The noises ξ_{ji} are generated by

$$
\begin{aligned}
&E_0 = U_0, \quad U_0 \sim \mathrm{WN}(0, \sigma_\xi^2 \mathbf{1}_{m_1} \mathbf{1}_{m_2}^T), \\
&E_t = \beta E_{t-1} + \sqrt{1-\beta^2} U_t, \quad U_t \sim \mathrm{WN}(0, \sigma_\xi^2 \mathbf{1}_{m_1} \mathbf{1}_{m_2}^T), \\
&\xi_{ji} = \langle E_j, X_{ji} \rangle, \quad j = 1, 2, \ldots, T, \ i = 1, 2, \ldots, n_t,
\end{aligned} \tag{22}
$$

where $\mathrm{WN}(0, \sigma_\xi^2 \mathbf{1}_{m_1} \mathbf{1}_{m_2}^T)$ generates the $m_1 \times m_2$ random matrix whose elements are i.i.d. normal with mean zero and variance σ_ξ^2. The larger β is, the stronger dependence exists among $\{\xi_{ji}\}_j$. The other parameters are set are the same as those in the independent case. Then $\Phi_{\mathcal{X}} = 1$ and $\Phi_{\mathcal{Y}}$ can be calculated from (22) which is larger than 1 when $\beta > 0$. We apply the proposed method to recover M_t^0 for $1 \le t \le T$ under the combinations of $\sigma_\xi = 1, 2$ and $\beta = 0, 0.3, 0.6, 0.9$. As illustrated in the left panel of Figure 3, the MSE increases as σ_ξ^2 and/or β enlarge(s). Note that according to (S.26) in Supplement, the MSE is proportional to $\Phi_{\mathcal{Y}}^{8/5}$ given M_t, n, T. We plot the average MSE versus $\Phi_{\mathcal{Y}}^{8/5}$ under different σ_ξ^2 in the right panel of Figure 3 which verifies the conclusion.

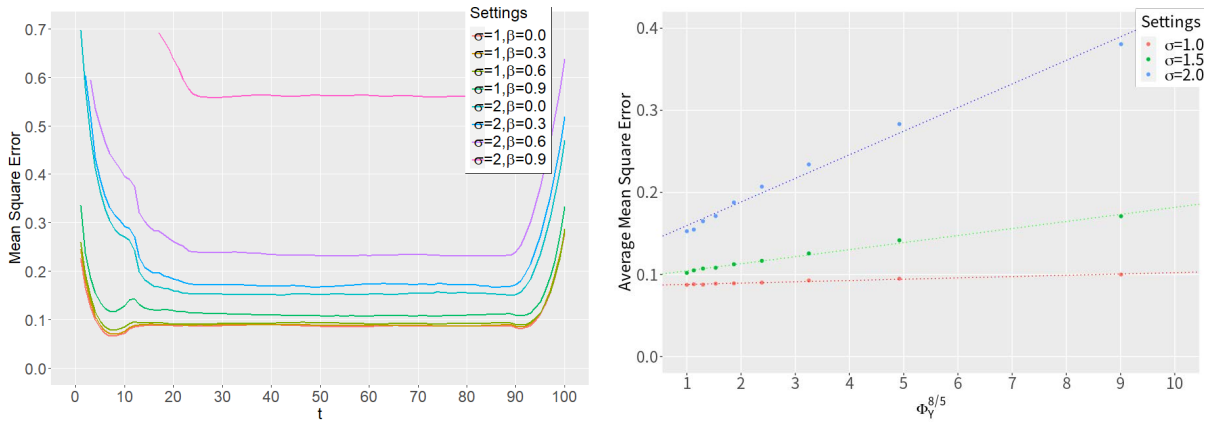

Figure 3: The left panel illustrates the MSE_t across time with different σ_ξ and β and the right panel shows that the average MSE versus $\Phi_{\mathcal{Y}}^{8/5}$ under different settings when $\Phi_{\mathcal{X}} = 1$.

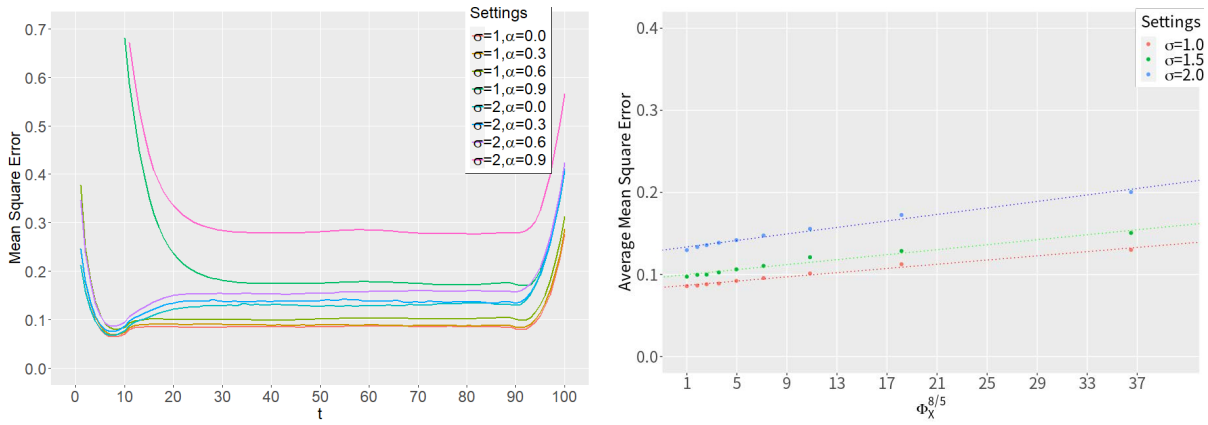

Figure 4: The left panel illustrates the MSE_t across time with different σ_ξ and α and the right panel shows that the average MSE versus $\Phi_{\mathcal{X}}^{8/5}$ under different settings when $\Phi_{\mathcal{Y}} = 1$.

Finally we study the influence of the dependence of design matrices on the estimation. We sample $X_{1i}, i = 1, \ldots, n$ from $\mathcal{E}$ with uniform probabilities independently and generate $X_{ji}, j \geq 2, i = 1, 2, \ldots, n$ by preserving αn elements randomly in $X_{j-1,i}, i = 1, 2, \ldots, n$ and choosing $(1-\alpha)n$ new elements i.i.d. from $\mathcal{E}$. Then the dependence of design matrices across time becomes stronger as α increases. We set the remaining parameters the same as those in the independent case and have $\Phi_{\mathcal{Y}} = 1$, $\Phi_{\mathcal{X}} \geq 1$, where the equality holds if and only if $\alpha = 0$. We plot the MSE_t across time under different $\sigma_\xi = 1, 2$ and $\alpha = 0, 0.3, 0.6, 0.9$ and verify the linear relationship between the average MSE and $\Phi_{\mathcal{X}}^{8/5}$ in Figure 4. We remark that the slopes of linear relationships in the right panels of Figure 3 and 4 are different. This is because the bound (S.26) in Supplement is obtained by the technique that $C_M \Phi_{\mathcal{X}} + \sigma_\xi \Phi_{\mathcal{Y}} \leq (C_M \vee \sigma_\xi)(\Phi_{\mathcal{X}} \vee \Phi_{\mathcal{Y}})$. Hence the curves of the average MSE versus $\Phi_{\mathcal{Y}}^{8/5}$ become sharper as σ_ξ increases, as shown in the right panel of Figure 3, while the slopes of the average MSE versus $\Phi_{\mathcal{X}}^{8/5}$ under different σ_ξ in the right panel of Figure 4 are similar.

5 Real Data Examples

In this section, we apply the proposed method to two real data examples for the dynamic matrix completion and compressed sensing problems, respectively.

The first example is the Netflix Prize Dataset (Netflix 2006), in which users ratings for movies received by Netflix were collected from October 1998 to December 2005 and can be downloaded at https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data. The ratings are integer-valued from 1 to 5. We preprocess the dataset to remove those movies which are watched less than 25000 times.

For user selection, we consider two different filters. Filter 1 is to choose users whose rating times are more than 700 because users who watch a larger number of movies tend to offer more reliable ratings and make better comparisons across different movies due to their broader exposure. There remain 3036 users and 1034 movies with total 2337997 ratings. We utilize a highly efficient data merging approach that requires minimal computational resources. Specifically, we divide the observations into $T = 100$ chronological time intervals. Within each interval, observations are aggregated as they occur at distinct time points. Then we randomly choose 4/5 of them in each interval as training data and set the rest as test data. At each time point t, there exists an underlying matrix $M_t \in \mathbb{R}^{m_1 \times m_2}$ to be recovered, where the rows represent users and the columns represent movies. The observed sparse matrix $Y_t \in \mathbb{R}^{m_1 \times m_2}$ has elements at position (j, k), representing the rating given by user j to movie k at time point t, or zero if no rating is available. Thus we can convert it to the trace regression model. The average sample size at each time point is about 17803 and thus the rate of observation samples in each time point is $\rho \approx 0.0057$. We use again the Epanechnikov kernel $K(x)$ and select the bandwidth h in the same way as in the simulations. Filter 2 is to select users whose rating times are more than 30 because those

users can represent the target groups of users who often using Netflix to watch movies. Then we conduct a random selection of 3000 individuals from the pool of eligible users for ease of computation.

We use 5-fold cross validation to choose tuning parameter λ and evaluate the performance of $\widetilde{M}_t^{\lambda}$ by calculating

$$\mathrm{MSE}_t^* = \frac{1}{n_t^{\mathrm{test}}} \sum_{i=1}^{n_t^{\mathrm{test}}} \left(\langle X_{t,i}^{\mathrm{test}}, \widetilde{M}_t^{\lambda} \rangle - Y_{t,i}^{\mathrm{test}} \right)^2, \tag{23}$$

using the test data and compare the proposed method with those three benchmarks mentioned in Section 4. Figure 5 shows that our dynamic method is more accurate than those benchmarks across the time domain. And the average MSE^*, i.e. $T^{-1} \sum_{t=1}^{T} \mathrm{MSE}_t^*$, of our dynamic method and benchmarks Static, TwoStep and Tensor methods are compared in Table 1.

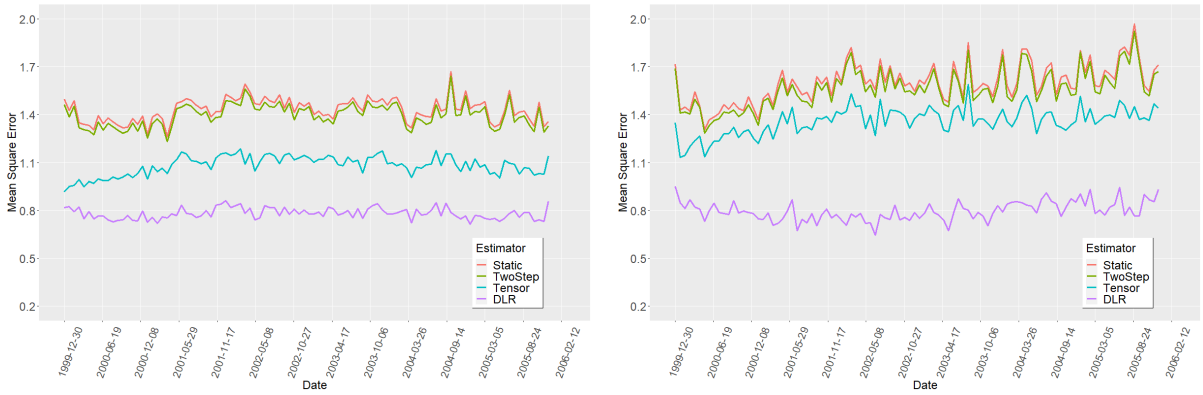

Figure 5: The MSE_t^* (23) at different time points for the proposed method and three benchmarks for the Netflix dataset. The left plot is the result for Filter 1 and the right plot is the result for Filter 2.

Another example is the compression and recovery of videos. We use the lion video from Davis 2017 dataset (Pont-Tuset et al. 2017) and treat each frame as a matrix at each time point. The dataset at https://davischallenge.org/davis2017/code.html#unsupervised

Table 1: The average MSE* for Netflix dataset.

Setting	DLR	Static	TwoStep	Tensor
Filter 1 MSE*	0.781	1.431	1.397	1.082
Filter 2 MSE*	0.796	1.601	1.565	1.360

is publicly available. For compression, we first separate the matrices M_t^0 into the sparse part S_t and the low-rank part L_t using robust principal component analysis (Candès et al. 2011). The design matrices X_{ti} are the convolution kernel matrices with random centers (a_{ti}, b_{ti}), i.e., $[X_{ti}]_{a_{ti},b_{ti}} = 4, [X_{ti}]_{a_{ti}\pm1,b_{ti}} = [X_{ti}]_{a_{ti},b_{ti}\pm1} = 2, [X_{ti}]_{a_{ti}\pm1,b_{ti}\pm1} = 1$. It is straightforward to verify that the distributions of $\{X_{ti}\}_{t,i}$ satisfy Assumption 4. Then we compress the low-rank part to obtain corresponding output $Y_{ti} = \langle X_{ti}, L_t \rangle$. For recovery, we use the dynamic trace regression to reconstruct the low-rank part, denoted as $\widetilde{L}_t^\lambda$. Then we recover each frame by $\widetilde{M}_t^\lambda = S_t + \widetilde{L}_t^\lambda$, while the sparse part S_t is retained. We emphasize that we only need store the centers (a_{ti}, b_{ti}), the output Y_{ti} and the nonzero part of S_t to recover the matrices M_t^0, which is considerably space-saving compared to storing the whole video. In the lion video, each frame has 480×854 pixels. We generate Y_{ti} with the rate of observation samples $\rho = 0.146$ and the file volume is reduced by 70% after compression. Here we use Static and TwoStep mentioned in Section 4 to reconstruct $\widehat{L}_t^\lambda, \widehat{L}_t^{\text{t-s}}$ and recover $\widehat{M}_t^\lambda = S_t + \widehat{L}_t^\lambda$, $\widehat{M}^{\text{t-s}} = S_t + \widehat{L}_t^{\text{t-s}}$ as benchmarks. Figure 6 shows the original frames M_t^0 and the estimates $\widetilde{M}_t^\lambda, \widehat{M}_t^\lambda, \widehat{M}^{\text{t-s}}$ at different time points. The background and the smooth motions of all three lions are recovered well by the proposed DLR method, which can be seen visually better than two benchmark methods. Recall the definition (23), the average $\text{MSE}^* = T^{-1}\sum_{t=1}^T \text{MSE}_t^*$ is 2.9×10^{-3} for our dynamic method while is 7.6×10^{-2} and 4.7×10^{-2} for the two benchmarks Static and TwoStep repsectively.

Figure 6: The top five pictures are the 5th, 25th, 45th, 65th and 85th original frames in the lions video, the second rows are the corresponding estimates of the proposed DLR method and the third and fourth rows are the benchmarks Static and TwoStep, respectively.

SUPPLEMENTARY MATERIAL

Dynamic_Matrix_Recovery_supp Detail proofs for all theorems and corollaries in "Dynamic Matrix Recovery", theoretical results for dynamic compressed sensing and additional numerical results. (.pdf file)

Code and Data for Dynamic Matrix Recovery R-code to implement and reproduce the simulation and real data results, corresponding output and raw datasets.

References

Agarwal, A., Negahban, S. & Wainwright, M. J. (2012), 'Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions', The Annals of Statistics **40**(2), 1171–1197.

Beck, A. & Teboulle, M. (2009), 'A fast iterative shrinkage-thresholding algorithm for linear inverse problems', SIAM journal on imaging sciences **2**(1), 183–202.

Bi, X., Tang, X., Yuan, Y., Zhang, Y. & Qu, A. (2021), 'Tensors in statistics', Annual review of statistics and its application **8**, 345–368.

Candès, E. J., Li, X., Ma, Y. & Wright, J. (2011), 'Robust principal component analysis?', Journal of the ACM (JACM) **58**(3), 1–37.

Candes, E. J. & Plan, Y. (2011), 'Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements', IEEE Transactions on Information Theory **57**(4), 2342–2359.

Candès, E. J. & Recht, B. (2009), 'Exact matrix completion via convex optimization', Foundations of Computational mathematics **9**(6), 717–772.

Csurgay, Á. I., Porod, W. & Lent, C. S. (2000), 'Signal processing with near-neighbor-coupled time-varying quantum-dot arrays', IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications **47**(8), 1212–1223.

Doukhan, P. (2012), Mixing: properties and examples, Vol. 85, Springer Science & Business Media.

Fan, J., Wang, W. & Zhu, Z. (2021), 'A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery', Annals of statistics **49**(3), 1239.

Gandy, S., Recht, B. & Yamada, I. (2011), 'Tensor completion and low-n-rank tensor recovery via convex optimization', Inverse problems **27**(2), 025010.

Gao, P. & Wang, M. (2018), Dynamic matrix recovery from partially observed and erroneous measurements, in '2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', IEEE, pp. 4089–4093.

Gross, D., Liu, Y.-K., Flammia, S. T., Becker, S. & Eisert, J. (2010), 'Quantum state

tomography via compressed sensing', Physical review letters **105**(15), 150401.

Hang, H. & Steinwart, I. (2017), 'A bernstein-type inequality for some mixing processes and dynamical systems with an application to learning', The Annals of Statistics **45**(2), 708–743.

Hu, X. & Yao, F. (2022), 'Dynamic principal component analysis in high dimensions', Journal of the American Statistical Association pp. 1–12.

Ji, S. & Ye, J. (2009), An accelerated gradient method for trace norm minimization, in 'Proceedings of the 26th annual international conference on machine learning', pp. 457–464.

Keshavan, R., Montanari, A. & Oh, S. (2009), 'Matrix completion from noisy entries', Advances in neural information processing systems **22**.

Klopp, O. (2011), 'Rank penalized estimators for high-dimensional matrices', Electronic Journal of Statistics **5**, 1161–1183.

Koltchinskii, V., Lounici, K. & Tsybakov, A. B. (2011a), 'Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion', The Annals of Statistics **39**(5).

Koltchinskii, V. & Xia, D. (2015), 'Optimal estimation of low rank density matrices.', J. Mach. Learn. Res. **16**(53), 1757–1792.

Koren, Y. (2009), Collaborative filtering with temporal dynamics, in 'Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 447–456.

Koren, Y., Bell, R. & Volinsky, C. (2009), 'Matrix factorization techniques for recommender systems', Computer **42**(8), 30–37.

Li, X. P., Huang, L., So, H. C. & Zhao, B. (2019), 'A survey on matrix completion: Perspective of signal processing', arXiv:1901.10885 .

Liu, J., Musialski, P., Wonka, P. & Ye, J. (2012), 'Tensor completion for estimating missing values in visual data', IEEE transactions on pattern analysis and machine intelligence **35**(1), 208–220.

Lois, B. & Vaswani, N. (2015), Online matrix completion and online robust pca, in '2015 IEEE International Symposium on Information Theory (ISIT)', IEEE, pp. 1826–1830.

Merlevède, F., Peligrad, M. & Rio, E. (2009), Bernstein inequality and moderate deviations under strong mixing conditions, in 'High dimensional probability V: the Luminy volume', Institute of Mathematical Statistics, pp. 273–292.

Merlevède, F., Peligrad, M. & Rio, E. (2011), 'A bernstein type inequality and moderate deviations for weakly dependent sequences', Probability Theory and Related Fields **151**(3), 435–474.

Negahban, S. & Wainwright, M. J. (2011), 'Estimation of (near) low-rank matrices with noise and high-dimensional scaling', The Annals of Statistics **39**(2), 1069–1097.

Nesterov, Y. E. (1983), A method of solving a convex programming problem with convergence rate $o(\frac{1}{k^2})$, in 'Doklady Akademii Nauk', Russian Academy of Sciences.

Netflix (2006), 'Netflix prize data', `http://netflixprize.com/index.html`.

Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A. & Van Gool, L. (2017), 'The 2017 davis challenge on video object segmentation', arXiv:1704.00675 .

Qiu, K., Mao, X., Shen, X., Wang, X., Li, T. & Gu, Y. (2017), 'Time-varying graph signal reconstruction', IEEE Journal of Selected Topics in Signal Processing **11**(6), 870–883.

Recht, B., Fazel, M. & Parrilo, P. A. (2010), 'Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization', SIAM review **52**(3), 471–501.

Toh, K.-C. & Yun, S. (2010), 'An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems', Pacific Journal of optimization **6**(615-640), 15.

Vershynin, R. (2018), High-dimensional probability: An introduction with applications in data science, Vol. 47, Cambridge university press.

Wang, X., Donaldson, R., Nell, C., Gorniak, P., Ester, M. & Bu, J. (2016), Recommending groups to users using user-group engagement and time-dependent matrix factorization, in 'Proceedings of the AAAI Conference on Artificial Intelligence'.

Xia, D. & Yuan, M. (2021), 'Statistical inferences of linear forms for noisy matrix completion', Journal of the Royal Statistical Society Series B **83**(1), 58–77.

Xu, L. & Davenport, M. (2016), 'Dynamic matrix recovery from incomplete observations under an exact low-rank constraint', Advances in Neural Information Processing Systems **29**.

Xu, X., Dong, F., Li, Y., He, S. & Li, X. (2020), Contextual-bandit based personalized recommendation with time-varying user interests, in 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 34, pp. 6518–6525.

Zhang, A. R., Luo, Y., Raskutti, G. & Yuan, M. (2020), 'Islet: Fast and optimal low-rank tensor regression via importance sketching', SIAM journal on mathematics of data science **2**(2), 444–479.

Zhang, Y., Bi, X., Tang, N. & Qu, A. (2021), 'Dynamic tensor recommender systems', The Journal of Machine Learning Research **22**(1), 3032–3066.

Zhao, T., Wang, Z. & Liu, H. (2015), 'A nonconvex optimization framework for low rank matrix estimation', Advances in Neural Information Processing Systems **28**.

Zheng, Q. & Lafferty, J. (2016), 'Convergence analysis for rectangular matrix completion using burer-monteiro factorization and gradient descent', arXiv:1605.07051 .

Zhou, H., Li, L. & Zhu, H. (2013), 'Tensor regression with applications in neuroimaging data analysis', Journal of the American Statistical Association **108**(502), 540–552.