RESEARCH

Open Access

Local low-rank approach to nonlinear matrix completion

Ryohei Sasaki^{1*} , Katsumi Konishi¹, Tomohiro Takahashi² and Toshihiro Furukawa³

*Correspondence:

clustereddata@gmail.com

¹Faculty of Computer and Information Science, Hosei University, 3-7-2 Kajino-cho Koganei-shi, Tokyo, Japan
Full list of author information is available at the end of the article

Abstract

This paper deals with a problem of matrix completion in which each column vector of the matrix belongs to a low-dimensional differentiable manifold (LDDM), with the target matrix being high or full rank. To solve this problem, algorithms based on polynomial mapping and matrix-rank minimization (MRM) have been proposed; such methods assume that each column vector of the target matrix is generated as a vector in a low-dimensional linear subspace (LDLS) and mapped to a p th order polynomial and that the rank of a matrix whose column vectors are d th monomial features of target column vectors is deficient. However, a large number of columns and observed values are needed to strictly solve the MRM problem using this method when p is large; therefore, this paper proposes a new method for obtaining the solution by minimizing the rank of the submatrix without transforming the target matrix, so as to obtain high estimation accuracy even when the number of columns is small. This method is based on the assumption that an LDDM can be approximated locally as an LDLS to achieve high completion accuracy without transforming the target matrix. Numerical examples show that the proposed method has a higher accuracy than other low-rank approaches.

Keywords: Matrix rank minimization, Nonlinear matrix completion, Differentiable manifold, Dimensionality reduction

1 Introduction

This paper deals with the following completion problem for a matrix $X \in \mathbb{R}^{M \times N}$ on a low-dimensional differentiable manifold (LDDM) $\mathcal{M}_r$:

$$\begin{aligned} &\text{Find} && X = [x_1 \ x_2 \ \cdots \ x_N] \\ &\text{subject to} && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m,n) \in \Omega \\ &&& x_i \in \mathcal{M}_r \text{ for all } i \in \mathcal{I}, \end{aligned} \quad (1)$$

where the (m,n) th element of a matrix is denoted by $(\cdot)_{m,n}$, $\mathcal{I}$ is an index set defined as $\mathcal{I} = \{1, 2, \dots, N\}$, and $\mathcal{M}_r \subset \mathbb{R}^M$, Ω , and $X^{(0)}$ denote an unknown r -dimensional differential manifold, a given index set, and a given observed matrix, respectively. In this paper, the LDDM $\mathcal{M}_r$ satisfies the following condition: on an open set $\mathcal{U}_\lambda$ satisfying $\bigcup_\lambda \mathcal{U}_\lambda = \mathcal{M}_r$, there exists a differentiable homeomorphism $\phi_\lambda : \mathcal{U}_\lambda \mapsto \mathcal{U}'_\lambda$, where $\mathcal{U}'_\lambda$ denotes an open set of $\mathbb{R}^r$. If $\mathcal{M}_r$ is an unknown low-dimensional linear subspace (LDLS), then this is a low-rank matrix completion problem. Many algorithms have

been proposed [1–6] to obtain solutions to this problem with high estimation accuracy. The low-rank matrix completion problem has various applications in the field of signal processing, including collaborative filtering [7], low-order model fitting and system identification [8], image inpainting [9], and human-motion recovery [10], all of which are formulated as signal recovery or estimation problems. However, in most practical applications, the column vectors of a matrix do not belong to an LDLS, i.e., $\mathcal{M}_r$ is not an LDLS. Therefore, these algorithms do not achieve high performance. As an example, a matrix is of high rank when its column vectors lie on a union of linear subspaces (UoLS), which the column space of the matrix is high dimension even when the dimension of the linear subspace is low. In this case, several methods have been proposed to solve this high-rank matrix completion problem [11–16], all of which are based on subspace clustering [17]. In particular, [15] proposed an algebraic variety approach known as variety-based matrix completion (VMC), which is based on the fact that the monomial features of each column vector belong to an LDLS when the column vectors belong to a UoLS. This approach solves the rank minimization problem about the Gram matrix of the monomial features by relaxing the problem into one of rank minimization of a polynomial kernel matrix. Unfortunately, these algorithms recover a matrix only when $\mathcal{M}_r$ can be approximately divided into some LDLSs and do not work well otherwise.

To solve the matrix completion problem on a general LDDM, some nonlinear methods have been proposed [18–22]. In particular, Fan et al. [19–21] have proposed a method based on a kind of kernel principal component analysis [23] that assumes that the dimension of the subspace spanned by the column vectors mapped nonlinearly is low. They formulate the matrix completion problem as a low-rank approximation problem of the kernel matrix, in common with [15]; however, they require a large number of observed entries in the matrix to solve the problem, and the matrix completion accuracy declines when the number of observed entries is small.

In the present paper, a new method is proposed that uses neither the monomial features nor the kernel method to achieve high completion accuracy. Based on an idea similar to that of locally linear embedding [24, 25], this paper assumes that an LDDM can be approximated locally as a LDLS, because there are tangent hyperplanes whose dimension is equal to that of the manifold. The matrix completion problem is then formulated as one of minimizing the rank of the local submatrix of X whose columns are local nearest neighborhoods of each other.

This paper is organized as follows. In Section 2, related works are introduced. Section 3 proposes a local low-rank approach (LLRA) to solve a matrix completion problem on an LDDM, and the convergence properties of the proposed algorithm are shown in Section 4. Finally, numerical examples are presented in Section 5 to illustrate that the proposed algorithm has a higher accuracy than other low-rank approaches.

2 Related works

Here, we focus on some matrix completion algorithms based on matrix rank minimization (MRM) on an unknown manifold, $\mathcal{M}_r$. First, this paper introduces the algorithms for the case where $\mathcal{M}_r$ is an r -dimensional linear subspace in Section 2.1; then, Section 2.2 shows the algorithms using the polynomial kernel for a UoLS and an LDDM.

2.1 Matrix rank minimization for linear subspace

Most algorithms for matrix completion deal with the case where the manifold $\mathcal{M}_r$ is an LDLS [1–3, 5]. In this case, since the dimension of r is unknown, they formulate a matrix completion problem as the following MRM problem to simultaneously estimate r and to restore X .

$$\begin{aligned} & \underset{X}{\text{Minimize}} && \text{rank}(X) \\ & \text{subject to} && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m,n) \in \Omega. \end{aligned} \quad (2)$$

Since this problem is generally NP-hard, several surrogate functions such as the nuclear norm [1] and truncated nuclear norm [5], Schatten norm [3] have been proposed. These algorithms recover X well if X can be approximated as a low-rank matrix.

2.2 High-rank matrix completion with the kernel method

To recover a high-rank matrix with columns belonging to an UoLS or an LDDM, some algorithms have been proposed that minimize the rank of its kernel matrix [15, 18–21].

In [15], the authors focused on a matrix completion problem on an union of d linear subspaces $\bigcup_{k=1}^d \mathcal{S}_k$, where $\mathcal{S}_k$ denotes an LDLS of dimension r or lower. Since the matrix rank is high or full in this problem, the MRM approach does not achieve high performance. To solve this matrix completion problem, an algebraic variety model approach was proposed based on the fact that the monomial features of each column vector ($\mathbf{x}_i \in \bigcup_{k=1}^d \mathcal{S}_k$) belong to a LDLS.

Here, the monomial features of $\mathbf{x}$ are defined as:

$$\boldsymbol{\psi}_d(\mathbf{x}) = (\mathbf{x}^\alpha)_{|\alpha| \leq d} \in \mathbb{R}^{\binom{M+d}{d}}, \quad (3)$$

$\alpha = [\alpha_1 \cdots \alpha_M]$ denotes a multi-index of non-negative integers, $\mathbf{x}^\alpha$ is defined as $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_M^{\alpha_M}$, $|\alpha| = \alpha_1 + \cdots + \alpha_M$.

Since $\mathbf{x} \in \bigcup_{k=1}^d \mathcal{S}_k$ if and only if $\prod_{k=1}^d (\mathbf{x}^T \mathbf{a}_k) = 0$ (where $\mathbf{a}_k$ denotes a vector in the orthogonal complement of $\mathcal{S}_k$), there exists a vector $\mathbf{c} \in \mathbb{R}^{\binom{M+d}{d}}$ that satisfies $\mathbf{c}^T \boldsymbol{\psi}_d(\mathbf{x}) = 0$. Hence, the matrix $\boldsymbol{\psi}_d(X) = [\boldsymbol{\psi}_d(\mathbf{x}_1) \boldsymbol{\psi}_d(\mathbf{x}_2) \cdots \boldsymbol{\psi}_d(\mathbf{x}_N)]$ is rank deficient, and the high-rank matrix completion problem is formulated as follows:

$$\begin{aligned} & \underset{X}{\text{Minimize}} && \text{rank}(\boldsymbol{\psi}_d(X)) \\ & \text{subject to} && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m,n) \in \Omega. \end{aligned} \quad (4)$$

This problem can be solved efficiently by replacing $\boldsymbol{\psi}_d(X)$ with a polynomial kernel-gram matrix and by using the Schatten norm-minimization algorithm [3]. The details are presented in [15].

Another approach to the high-rank matrix completion problem was proposed in [19–21]. The matrix $\boldsymbol{\psi}_d(X)$ is rank deficient when each column vector $\mathbf{x}_i$ is given by a polynomial mapping of latent features $\mathbf{y}_i \in \mathbb{R}^r$ ($r \ll M < N$) denoted by:

$$\mathbf{x}_i = \mathbf{U}_p \boldsymbol{\psi}_p(\mathbf{y}_i)$$

with polynomial coefficients $\mathbf{U}_p \in \mathbb{R}^{M \times \binom{r+p}{p}}$ and order $p \ll M$, because $R = \text{rank}(\boldsymbol{\psi}_d(X))$ satisfies:

$$R = \min \left\{ N, \binom{M+d}{d}, \binom{r+pd}{pd} \right\}$$

and $R < \binom{M+d}{d}$ if $r, p \ll M < N$. Therefore, the matrix $\psi_d(X)$ can be approximated by a low-rank matrix. [19–21] proposed a high-rank matrix-completion algorithm using matrix factorization in the same way as [15]; however, this algorithm requires a large number of observed entries and does not recover the matrix when only a small number are present. The algorithm restores $[\psi_d(\mathbf{x}_1) \cdots \psi_d(\mathbf{x}_N)]$ uniquely if the sample number N and the sampling rate $q = \frac{|\Omega|}{MN}$ satisfy the inequality:

$$q \geq \left(\frac{R}{N} + \frac{R}{\binom{M+d}{d}} - \frac{R^2}{N \binom{M+d}{d}} \right)^{\frac{1}{d}}.$$

For example, when $p = 3, r = 5, m = 100, d = 2$, $\binom{r+pd}{pd} = 462$ and $\binom{M+d}{d} = 5151$, although the ratio $\binom{r+pd}{pd} / \binom{M+d}{d} \ll \binom{r+p}{p} / M = 0.56$, we need $N \geq 5982$ for $q = 0.4$ and $N \geq 1362465$ for $q = 0.3$. Hence, we expect that the matrix-completion accuracy will worsen when p and r are high and N is small.

Therefore, this paper proposes a new approach that makes use of neither monomial features nor the kernel method, but which is rather based on the assumption that an LDDM can be approximated locally as an LDLS to achieve high completion accuracy with a small q and too few samples N .

3 Methods

3.1 Local low-dimensional model

First, in order to consider how the LDDM is structured when the matrix X is given, this paper assumes that some columns $\mathbf{x}_j$ are approximated by a vector within a set of tangent vectors $\mathbf{x} + U(\mathbf{x})$ at $\mathbf{x}$ in the LDDM $\mathcal{M}_r$. Here, $U(\mathbf{x})$ is defined as:

$$U(\mathbf{x}) = \{J(\mathbf{x})\Delta\mathbf{y} \in \mathbb{R}^M \mid \Delta\mathbf{y} \in \mathbb{R}^r, \|\Delta\mathbf{y}\|_2^2 \leq \epsilon\}. \quad (5)$$

$\epsilon > 0$ denotes the radius of an r -dimensional hyperball, and $J(\mathbf{x})$ denotes a Jacobian matrix defined as:

$$J(\mathbf{x}) = \left[\begin{array}{ccc} \frac{\partial \phi_{\lambda,1}^{-1}}{\partial y_1} & \frac{\partial \phi_{\lambda,1}^{-1}}{\partial y_2} & \cdots & \frac{\partial \phi_{\lambda,1}^{-1}}{\partial y_r} \\ \frac{\partial \phi_{\lambda,2}^{-1}}{\partial y_1} & \frac{\partial \phi_{\lambda,2}^{-1}}{\partial y_2} & \cdots & \frac{\partial \phi_{\lambda,2}^{-1}}{\partial y_r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial \phi_{\lambda,M}^{-1}}{\partial y_1} & \frac{\partial \phi_{\lambda,M}^{-1}}{\partial y_2} & \cdots & \frac{\partial \phi_{\lambda,M}^{-1}}{\partial y_r} \end{array} \right]_{\mathbf{y}=\phi_{\lambda}(\mathbf{x})}.$$

$\phi_{\lambda} : \mathcal{U}_{\lambda} \mapsto \mathcal{U}'_{\lambda}$ and $\phi_{\lambda}^{-1} = [\phi_{\lambda,1}^{-1} \cdots \phi_{\lambda,M}^{-1}]^T : \mathcal{U}'_{\lambda} \mapsto \mathcal{U}_{\lambda}$ denote a chart and its inverse for an index λ , with an open set $\mathcal{U}_{\lambda}$ that includes $\mathbf{x}$ satisfying $\bigcup_{\lambda} \mathcal{U}_{\lambda} = \mathcal{M}_r$ and $\mathcal{U}'_{\lambda} \subset \mathbb{R}^r$. Then, we consider that each $\mathbf{x}_j$ in a set $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ can be approximated by a vector belonging to $\bigcup_{i \neq j} (\mathbf{x}_i + U(\mathbf{x}_i))$ for all $j \in \mathcal{I}$. In other words, we assume that we have the following non-empty-index set $\mathcal{I}_i$ for $i \in \mathcal{I}$ defined as:

$$\mathcal{I}_i = \{j \in \mathcal{I} \mid \|\mathbf{x}_j - \mathbf{x}_i - \mathbf{z}_{i,j}\|_2^2 \leq \eta, \mathbf{z}_{i,j} \in U(\mathbf{x}_i)\}, \quad (6)$$

where $\eta > 0$ denotes the upper bound of the Euclidean distance between $\mathbf{x}_j - \mathbf{x}_i$ and a vector $\mathbf{z}_{i,j} \in U(\mathbf{x}_i)$. In this case, the rank of a matrix $\mathbf{Z}_i = [\mathbf{z}_{i,j_1} \mathbf{z}_{i,j_2} \cdots \mathbf{z}_{i,j_{|\mathcal{I}_i|}}]$ (where $\{j_1, j_2, \dots, j_{|\mathcal{I}_i|}\} = \mathcal{I}_i$) is less than or equal to r because of $\text{rank}(J(\mathbf{x}_i)) = r$.

Figure 1 illustrates the construction of each $\mathbf{z}_{i,j}$. From the figure, it is apparent that the $\mathbf{z}_{i,j} \in U(\mathbf{x}_i)$ can be obtained for suitable parameters ϵ and η . Therefore, the matrix-completion problem for an arbitrary LDDM (1) can be substituted with the problem of finding sets $\mathcal{I}_i, \mathbf{Z}_i$ that satisfy (6) and the missing entries of the matrix $\mathbf{X}$ with the set of tangent vectors $\mathbf{x}_i + U(\mathbf{x}_i)$ and given parameters ϵ and η .

Next, we consider how to find $\mathbf{z}_{i,j}$ and $\mathcal{I}_i$. To simplify the below explanation, we redefine a variable $\mathbf{z}_{i,j}$ as follows:

$$\mathbf{z}_{i,j} = (\mathbf{x}_j - \mathbf{x}_i)d_{i,j} + \mathbf{e}_{i,j}, \quad (7)$$

where $\mathbf{e}_{i,j}$ denotes an error vector satisfying $\|\mathbf{e}_{i,j}\|_2^2 \leq \eta$ and $d_{i,j} \in \{0, 1\}$ denotes a variable for which finding $d_{i,j}$ is equivalent to finding $\mathcal{I}_i$. In order to find a suitable solution for $d_{i,j}$, this paper formulates the following maximization problem:

$$\begin{aligned} & \underset{\substack{\mathbf{X}, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \\ \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2}}}{\text{Maximize}} && \sum_{i=1}^N \sum_{j=1}^N d_{i,j} \\ & \text{subject to} && \mathbf{z}_{i,j} \in U(\mathbf{x}_i) \\ & && \|(\mathbf{x}_j - \mathbf{x}_i)d_{i,j} - \mathbf{z}_{i,j}\|_2^2 \leq \eta \\ & && d_{i,j} \in \{0, 1\} \text{ for } (i,j) \in \mathcal{I}^2 \\ & && (\mathbf{X})_{m,n} = (\mathbf{X}^{(0)})_{m,n} \text{ for } (m,n) \in \Omega, \end{aligned} \quad (8)$$

where $\mathbf{Z}_i \in \mathbb{R}^{M \times N}$ is a matrix whose j th column vector is $\mathbf{z}_{i,j}$. Since the problem (8) cannot be solved because of $U(\mathbf{x}_i)$ when the LDDM $\mathcal{M}_r$ is unknown (as is often the case in actual problems), this paper reformulates the constraint condition $\mathbf{z}_{i,j} \in U(\mathbf{x}_i)$ as two constraint conditions: (1) $\text{rank}(\mathbf{Z}_i) \leq r$, because the span of $U(\mathbf{x}_i)$ is an r -dimensional linear subspace, and (2) $\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \leq \epsilon_i$ if $d_{i,j} = 1$. Because it is difficult to estimate the radius of an ellipsoid since each $\mathbf{J}(\mathbf{x}_i)$ is arbitrary and unknown, this paper uses the

Euclidean distance of $\mathbf{x}_j - \mathbf{x}_i$ and gives the radius of the hyperball ϵ_i for each $\mathbf{x}_i$. Thus, this paper reformulates the problem (8) with the given parameters r and $\{\epsilon_i\}_{i \in \mathcal{I}}$ as:

$$\begin{aligned}
 & \underset{\substack{X, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \\ \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2}}}{\text{Maximize}} && \sum_{i=1}^N \sum_{j=1}^N d_{i,j} \\
 & \text{subject to} && \text{rank}(\mathbf{Z}_i) \leq r \text{ for } i \in \mathcal{I}, \\
 & && (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{i,j} \leq 0 \\
 & && \|(\mathbf{x}_j - \mathbf{x}_i) d_{i,j} - \mathbf{z}_{i,j}\|_2^2 \leq \eta \\
 & && d_{i,j} \in \{0, 1\} \text{ for } (i, j) \in \mathcal{I}^2 \\
 & && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m, n) \in \Omega,
 \end{aligned} \tag{9}$$

where the 2nd constraint condition is the same as $\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \leq \epsilon_i$ if $d_{i,j} = 1$. Thus, we obtain the formulation for finding $\mathbf{z}_{i,j}$ on each $U(\mathbf{x}_i)$ and the set $\mathcal{I}_i$ without understanding $J(\mathbf{x})$. However, it is difficult to solve the problem (9) for $X, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2}$ at the same time due to the condition $\text{rank}(\mathbf{Z}_i) \leq r$. Actual applications may not be able to find a suitable dimension r . In order to solve this issue, the present paper explains how to obtain the solution using a MRM technique in Section 3.2.

3.2 Local low-rank approximation algorithm

First, we consider how to estimate the dimension of the LDDM, r , with an arbitrary matrix X . We can estimate r simply using a principal-component analysis if we obtain $d_{i,j}$; however, the lower the rank of the matrix $\mathbf{Z}_i$, the lower that the number of solutions to $d_{i,j} = 1$ becomes for the solution of (9). It can be seen that there is a trade-off between the dimension r and the number of solutions to $d_{i,j} = 1$. Therefore, this paper formulates the following problem:

$$\begin{aligned}
 & \underset{\substack{X, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \\ \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2}}}{\text{Minimize}} && \sum_{i=1}^N \left\{ \alpha \text{rank}(\mathbf{Z}_i) - (1 - \alpha) \sum_{j=1}^N d_{i,j} \right\} \\
 & \text{subject to} && \|(\mathbf{x}_j - \mathbf{x}_i) d_{i,j} - \mathbf{z}_{i,j}\|_2^2 \leq \eta, \\
 & && (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{i,j} \leq 0 \\
 & && d_{i,j} \in \{0, 1\} \text{ for } (i, j) \in \mathcal{I}^2 \\
 & && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m, n) \in \Omega,
 \end{aligned} \tag{10}$$

where $0 \leq \alpha \leq 1$ denotes a given trade-off parameter, which is the ratio of the decreasing rank of $\mathbf{Z}_i$ to the sum of $d_{i,j}$. Because solving the problem (10) is NP-hard due to $\text{rank}(\mathbf{Z}_i)$, this paper reformulates the problem as one of relaxation:

$$\begin{aligned}
 & \underset{\substack{X, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \\ \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2}}}{\text{Minimize}} && f_{\beta, \gamma} \left(X, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2} \right) \\
 & \text{subject to} && (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{i,j} \leq 0 \\
 & && d_{i,j} \in [0, 1] \text{ for } (i, j) \in \mathcal{I}^2 \\
 & && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m, n) \in \Omega,
 \end{aligned} \tag{11}$$

where f is defined as:

$$\begin{aligned}
 & f_{\beta, \gamma} \left(X, \{\mathbf{Z}_i\}_{i \in \mathcal{I}}, \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2} \right) \\
 & = \sum_{i=1}^N \left\{ \gamma \|\mathbf{Z}_i\|_* + \frac{1}{2} \left\| (X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i - \mathbf{Z}_i \right\|_F^2 \right\} \\
 & \quad - \beta \text{trace}(\mathbf{D}_i)
 \end{aligned} \tag{12}$$

Here, $\beta, \gamma \geq 0$ denote the given parameters, function $\|\cdot\|_*$ denotes the nuclear norm, $\mathbf{1}_N \in \mathbb{R}^N$ denotes the vector whose elements are all 1, $\mathbf{D}_i$ denotes a diagonal matrix whose diagonal elements $(\mathbf{D}_i)_{jj}$ each equal d_{ij} , and $\text{trace}(\mathbf{Y})$ denotes the sum of all diagonal elements of $\mathbf{Y}$.

Next, this paper presents a technique to solve the problem (11) using alternating optimization. Firstly, we consider how to solve the problem (11) for $\mathbf{Z}_i$ and d_{ij} with a fixed $\mathbf{X}$. We repeat the following schemes until a termination condition is satisfied with respect to $\mathbf{Z}_i$ and d_{ij} :

$$\begin{aligned} 1. & d_{ij} \leftarrow h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_{ij}) \quad \text{for } (i, j) \in \mathcal{I}^2, \\ 2. & \mathbf{Z}_i \leftarrow \mathcal{T}_\gamma \{(\mathbf{X} - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i\} \quad \text{for } i \in \mathcal{I}, \end{aligned} \quad (13)$$

where $h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_{ij})$ is defined as:

$$h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_{ij}) = \begin{cases} 0 & : \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 > \epsilon_i \\ 1 & : \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 = 0 \\ \text{sat} \left(\frac{\langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{z}_{ij} \rangle + \beta}{\|\mathbf{x}_j - \mathbf{x}_i\|_2^2} \right) & : \text{otherwise} \end{cases} \quad (14)$$

$\langle \mathbf{a}, \mathbf{b} \rangle$ denotes the inner product of $\mathbf{a}$ and $\mathbf{b}$, $\text{sat}(c) = \max(0, \min(1, c))$, and $\mathcal{T}_\tau$ denotes the matrix-shrinkage operator for the nuclear norm-minimization problem [1]. Each step of (13) minimizes the objective function (12) for $\mathbf{Z}_i$ and d_{ij} . Then, we consider minimizing the objective function (12) for $\mathbf{X}$ with fixed $\mathbf{Z}_i$ and d_{ij} . Since the objective function (12) is quadratic for $\text{vec}(\mathbf{X}) = \mathbf{x}$, we obtain the following solution to the quadratically constrained quadratic program for a given $\mathbf{Z}_i$ and d_{ij} :

$$\begin{aligned} & \underset{\mathbf{x}}{\text{argmin}} \quad \mathbf{x}^T (\mathbf{L} \otimes \mathbf{I}_{M,M}) \mathbf{x} - 2\mathbf{x}^T \mathbf{c} \\ & \text{subject to } (\mathbf{X})_{m,n} = (\mathbf{X}^{(0)})_{m,n} \quad \text{for } (m, n) \in \Omega \\ & \quad \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \leq \epsilon_i \quad \text{if } d_{ij} > 0, \end{aligned} \quad (15)$$

where $\mathbf{I}_{M,M} \in \mathbb{R}^{M \times M}$ is the identity matrix, $\otimes$ is the Kronecker product of two matrices, and $\mathbf{L} \in \mathbb{R}^{N \times N}$ is a graph Laplacian defined as:

$$\begin{aligned} \mathbf{L} &= \text{diag}(\hat{\mathbf{D}} \mathbf{1}_N) - \hat{\mathbf{D}}, \\ (\hat{\mathbf{D}})_{ij} &= d_{ij}^2 + d_{ji}^2 \quad \text{for } (i, j) \in \mathcal{I}^2. \end{aligned}$$

$\mathbf{c} = [\mathbf{c}_1^T \ \mathbf{c}_2^T \ \cdots \ \mathbf{c}_M^T]^T \in \mathbb{R}^{MN}$ is defined as:

$$\begin{aligned} \mathbf{c}_l &= (\tilde{\mathbf{D}} \odot \tilde{\mathbf{Z}}_l - (\tilde{\mathbf{D}} \odot \tilde{\mathbf{Z}}_l)^T) \mathbf{1}_N, \\ (\tilde{\mathbf{Z}}_l)_{ij} &= (\mathbf{Z}_i)_{lj} \quad \text{for } (i, j) \in \mathcal{I}^2, \\ (\tilde{\mathbf{D}})_{ij} &= d_{ij} \quad \text{for } (i, j) \in \mathcal{I}^2, \end{aligned}$$

for $l = 1, \dots, M$. $\odot$ denotes the Hadamard product. Thus, we can alternately optimize for each of $\mathbf{Z}_i, d_{ij}$ and $\mathbf{X}$ in the problem (11).

3.3 Truncated nuclear norm-minimization approach

The solution to the problem (11) is obtained by minimizing the function (12). However, the norm of the solution $\mathbf{X}$ might be below the true value, since nuclear norm minimization decreases not only the $(r+1)$ th biggest singular values, but also the 1st to the r th

biggest singular values. Therefore, this paper reformulates the problem and the evaluation function as follows:

$$\begin{aligned}
 & \underset{\substack{X, \{Z_i\}_{i \in \mathcal{I}}, \\ \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2}}}{\text{Minimize}} && g_{\beta, \gamma, r} \left(X, \{Z_i\}_{i \in \mathcal{I}}, \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2} \right) \\
 & \text{subject to} && (\|x_j - x_i\|_2^2 - \epsilon_i) d_{i,j} \leq 0 \\
 & && d_{i,j} \in [0, 1] \text{ for } (i, j) \in \mathcal{I}^2 \\
 & && (X)_{m,n} = (X^{(0)})_{m,n} \text{ for } (m, n) \in \Omega,
 \end{aligned} \tag{16}$$

$$\begin{aligned}
 & g_{\beta, \gamma, r} \left(X, \{Z_i\}_{i \in \mathcal{I}}, \{d_{i,j}\}_{(i,j) \in \mathcal{I}^2} \right) \\
 &= \sum_{i=1}^N \left\{ \gamma \|Z_i\|_{*,r} + \frac{1}{2} \|(X - x_i \mathbf{1}_N^T) D_i - Z_i\|_F^2 \right. \\
 & \quad \left. - \beta \text{trace}(D_i) \right\},
 \end{aligned} \tag{17}$$

where $r \in \{0, 1, \dots, M\}$ is a given parameter and the function $\|Z\|_{*,r}$ represents the truncated nuclear norm, which is defined with the k th biggest singular value σ_k of Z . The details of the truncated nuclear norm and the optimization technique are given in [Appendix](#). Note that the truncated nuclear norm with $r = 0$ is equal to the nuclear norm. In this case, the problem (11) is same as the problem (16). When the variables X and D_i are constant, the optimal solution for each Z_i is obtained by $Z_i = \mathcal{T}_{r, \gamma} \{(X - x_i \mathbf{1}_N^T) D_i\}$. In the same way, to solve the problem (11), this paper describes Algorithm 1 using iterative partial matrix shrinkage (IPMS) [5] for the problem (16), which contains the algorithm for (11). Here, $\mathbf{0}_{M,N} \in \mathbb{R}^{M \times N}$ denotes a zero matrix, η_1, η_2 denote lower limits for the termination conditions $\|D_i^{\text{old}} - D_i\|_F / \|D_i\|_F \leq \eta_1$, and $\|X^{\text{old}} - X\|_F / \|X\|_F \leq \eta_2$ and $\beta^{(k)}, \gamma^{(k)}, r^{(k)}$ denote given parameters that satisfy $0 < \beta^{(0)} \leq \beta^{(1)} \leq \dots \leq \beta^{\max}, \gamma_0 \geq \gamma_1 \geq \dots \geq \gamma_{\min} \geq 0, 0 \leq r_0 \leq r_1 \leq \dots \leq r_{\max} \leq M$.

Algorithm 1 LLRA algorithm for matrix completion with given r (LLRAr)

Require: $X^{(0)}, \Omega, \{\beta^{(k)}, \gamma^{(k)}, r^{(k)}\}_{k \in \mathbb{N}}, \{\epsilon_i\}_{i \in \mathcal{I}}, \eta_1, \eta_2$

- 1: $X \leftarrow X^{(0)}$
- 2: $Z_i \leftarrow \mathbf{0}_{M,N}$ for $i \in \mathcal{I}$
- 3: $D_i \leftarrow \mathbf{0}_{N,N}$ for $i \in \mathcal{I}$
- 4: **repeat**
- 5: $X^{\text{old}} \leftarrow X$
- 6: $\beta \leftarrow \beta^{(k)}, \gamma \leftarrow \gamma^{(k)}, r \leftarrow r^{(k)}$
- 7: **repeat**
- 8: $D_i^{\text{old}} \leftarrow D_i$ for $i \in \mathcal{I}$
- 9: $d_{i,j} \leftarrow h_{\beta, \epsilon_i}(x_j, x_i, z_{i,j})$ for $(i, j) \in \mathcal{I}^2$
- 10: $D_i \leftarrow \text{diag}([d_{i,1} \dots d_{i,N}]^T)$ for $i \in \mathcal{I}$
- 11: $Z_i \leftarrow \mathcal{T}_{r, \gamma} \{(X - x_i \mathbf{1}_N^T) D_i\}$ for $i \in \mathcal{I}$
- 12: **until** $\|D_i^{\text{old}} - D_i\|_F / \|D_i\|_F \leq \eta_1$
- 13: $x \leftarrow (15)$
- 14: $X \leftarrow \text{vec}^{-1}(x)$
- 15: **until** $\|X^{\text{old}} - X\|_F / \|X\|_F \leq \eta_2$

Ensure: X .

4 Convergence analysis

This section presents the convergence property of Algorithm 1.

First, let us define the following schemes with regard to the t th iteration of the second iteration statements in Algorithm 1:

$$\begin{cases} d_{ij}^{(t)} = h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_{ij}^{(t)}) & \text{for } (i, j) \in \mathcal{I}^2, \\ \mathbf{Z}_i^{(t+1)} = \mathcal{T}_{r, \gamma} \left\{ (X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i^{(t)} \right\} & \text{for } i \in \mathcal{I}, \end{cases} \quad (18)$$

$u(t_1, t_2)$ behaves as:

$$u(t_1, t_2) = g_{\beta, \gamma, r} \left(X, \left\{ \mathbf{Z}_i^{(t_1)} \right\}_{i \in \mathcal{I}}, \left\{ d_{ij}^{(t_2)} \right\}_{(i,j) \in \mathcal{I}^2} \right),$$

for $t, t_1, t_2 \geq 0$ and a given $X \in \mathbb{R}^{M \times N}$.

Lemma 1 For $t \geq 0$ and a given $X \in \mathbb{R}^{M \times N}$, the $d_{ij}^{(t)}$ generated by the update schemes (18) satisfies:

$$u(t, t) - u(t+1, t+1) \geq \frac{1}{2} \sum_{(i,j) \in \mathcal{I}^2} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \left(d_{ij}^{(t)} - d_{ij}^{(t+1)} \right)^2.$$

Proof $u(t_1, t_2)$ satisfies $u(t, t) \geq u(t+1, t) \geq u(t+1, t+1) \geq \dots \geq -\beta N^2$ for $t \geq 0$, since $d_{ij}^{(t)} = h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_{ij}^{(t)})$ is the closed-form optimal solution of the convex quadratic-minimization problem with linear constraints for fixed $\mathbf{Z}_i^{(t)}$, and $\mathbf{Z}_i^{(t+1)}$ represents the optimal solution for fixed $d_{ij}^{(t)}$ (from Theorem 1 of [5]). Each $d_{ij}^{(t+1)}$ satisfies the following KKT condition of problem (16) with $(\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{ij}^{(t+1)} \leq 0, d_{ij}^{(t)} - 1 \leq 0, -d_{ij}^{(t+1)} \leq 0$ for $(i, j) \in \mathcal{I}^2$:

$$\begin{cases} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 d_{ij}^{(t+1)} - \langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{z}_{ij}^{(t+1)} \rangle - \beta \\ = \mu_{1,ij}^{(t+1)} (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) + \mu_{2,ij}^{(t+1)} - \mu_{3,ij}^{(t+1)}, \\ \mu_{1,ij}^{(t+1)} (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{ij}^{(t+1)} = 0, \\ \mu_{2,ij}^{(t+1)} (d_{ij}^{(t+1)} - 1) = 0, \\ \mu_{3,ij}^{(t+1)} (-d_{ij}^{(t+1)}) = 0, \\ \mu_{1,ij}^{(t+1)}, \mu_{2,ij}^{(t+1)}, \mu_{3,ij}^{(t+1)} \geq 0 \end{cases}$$

where $\mu_{1,ij}^{(t+1)}, \mu_{2,ij}^{(t+1)}$ and $\mu_{3,ij}^{(t+1)}$ denote KKT multipliers for $d_{ij}^{(t+1)}$. Therefore, $u(t_1, t_2)$ satisfies:

$$\begin{aligned}
& u(t+1, t) - u(t+1, t+1) \\
&= \sum_{(i,j) \in \mathcal{I}^2} \left\{ \begin{aligned} & \frac{1}{2} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 (d_{ij}^{(t)^2} - d_{ij}^{(t+1)^2}) \\ & - \langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{z}_{ij}^{(t+1)} \rangle (d_{ij}^{(t)} - d_{ij}^{(t+1)}) \\ & - \beta (d_{ij}^{(t)} - d_{ij}^{(t+1)}) \end{aligned} \right\} \\
&= \sum_{(i,j) \in \mathcal{I}^2} \frac{1}{2} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 (d_{ij}^{(t)} - d_{ij}^{(t+1)})^2 \\
&+ \sum_{(i,j) \in \mathcal{I}^2} (d_{ij}^{(t)} - d_{ij}^{(t+1)}) \left(\begin{aligned} & \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 d_{ij}^{(t+1)} \\ & - \langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{z}_{ij}^{(t+1)} \rangle \\ & - \beta \end{aligned} \right) \\
&= \sum_{(i,j) \in \mathcal{I}^2} \frac{1}{2} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 (d_{ij}^{(t)} - d_{ij}^{(t+1)})^2 \\
&- \sum_{(i,j) \in \mathcal{I}^2} (d_{ij}^{(t)} - d_{ij}^{(t+1)}) \left\{ \begin{aligned} & \mu_{1,ij}^{(t+1)} (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) \\ & + \mu_{2,ij}^{(t+1)} - \mu_{3,ij}^{(t+1)} \end{aligned} \right\} \\
&= \sum_{(i,j) \in \mathcal{I}^2} \left\{ \begin{aligned} & \frac{1}{2} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 (d_{ij}^{(t)} - d_{ij}^{(t+1)})^2 \\ & + \mu_{1,ij}^{(t+1)} (\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) (d_{ij}^{(t+1)} - d_{ij}^{(t)}) \\ & + \mu_{2,ij}^{(t+1)} (d_{ij}^{(t+1)} - 1 - d_{ij}^{(t)} + 1) \\ & + \mu_{3,ij}^{(t+1)} (-d_{ij}^{(t+1)} + d_{ij}^{(t)}) \end{aligned} \right\},
\end{aligned}$$

Since $(\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{ij}^{(t+1)} = 0$ if $\mu_{1,ij}^{(t+1)} > 0$, $d_{ij}^{(t+1)} - 1 = 0$ if $\mu_{2,ij}^{(t+1)} > 0$ and $-d_{ij}^{(t+1)} = 0$ if $\mu_{3,ij}^{(t+1)} > 0$, and each $d_{ij}^{(t)}$ satisfies the constraint condition:

$$\begin{aligned}
& u(t, t) - u(t+1, t+1) \geq \\
& u(t+1, t) - u(t+1, t+1) \geq \\
& \frac{1}{2} \sum_{(i,j) \in \mathcal{I}^2} \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 (d_{ij}^{(t)} - d_{ij}^{(t+1)})^2.
\end{aligned}$$

Therefore, each sequence $\{d_{ij}^{(t)}\}$ converges to a limit point $\bar{d}_{ij}$ if $u(0, 0) < \infty$, because $d_{ij}^{(t)} - d_{ij}^{(t+1)} \rightarrow 0$ when $t \rightarrow \infty$ if $\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 > 0$ and $d_{ij}^{(t)} - d_{ij}^{(t+1)} = 1 - 1 = 0$, even if $\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 = 0$ for $t \geq 0$, $(i, j) \in \mathcal{I}^2$. Then, each sequence $\{\mathbf{Z}_i^{(t)}\}$ converges to a limit point $\bar{\mathbf{Z}}_i$ because each $\mathbf{Z}_i^{(t+1)}$ can be obtained by the soft-thresholding operator using fixed $d_{ij}^{(t)}$ for $t \geq 0$, $i \in \mathcal{I}$. $\square$

Lemma 2 If $\beta \geq \epsilon_i$, the optimal solution of (17) under the constraint conditions for d_{ij} and $\mathbf{Z}_i$ can be obtained by initializing $\mathbf{Z}_i^{(0)}$ as $\mathbf{Z}_i^{(0)} = \mathbf{0}_{M \times N}$ and updating $d_{ij}^{(0)}$ and $\mathbf{Z}_i^{(1)}$ using the update schemes (18) for a given $\mathbf{X} \in \mathbb{R}^{M \times N}$.

Proof From Theorem 1 of [5], any $\mathbf{X} \in \mathbb{R}^{M \times N}$ and each optimal solution $\bar{\mathbf{Z}}_i$ and $\bar{D}_i$ satisfies $\bar{\mathbf{Z}}_i = \mathcal{T}_{r,\gamma} \{(\mathbf{X} - \mathbf{x}_i \mathbf{1}_N^T) \bar{D}_i\}$. For a given $d_{ij} \geq 0$, a matrix $\mathbf{Z}_i = \mathcal{T}_{r,\gamma} \{(\mathbf{X} - \mathbf{x}_i \mathbf{1}_N^T) D_i\}$ satisfies $0 \leq \langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{z}_{ij} \rangle$ because, when $d_{ij} > 0$,

$$\begin{aligned}
\langle \mathbf{x}_j - \mathbf{x}_i, \mathbf{z}_{i,j} \rangle d_{i,j} &= \langle \mathbf{y}_{i,j}, \mathbf{z}_{i,j} \rangle \\
&= \sum_{l=1}^r \sigma_l^2 (V)_{j,l}^2 \\
&\quad + \sum_{l=r+1}^M \sigma_l (\sigma_l - \gamma)_+ (V)_{j,l}^2 \\
&\geq 0.
\end{aligned}$$

Here, $\mathbf{y}_{i,j}$ denotes the j th column of $\mathbf{Y}_i = (\mathbf{X} - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i = \mathbf{U} \text{diag}(\boldsymbol{\sigma}) \mathbf{V}^T$ and $\boldsymbol{\sigma}, \mathbf{U}, \mathbf{V}$ denotes the singular values and vectors of $(\mathbf{X} - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i$; when $d_{i,j} = 0$, $\langle \mathbf{y}_{i,j}, \mathbf{z}_{i,j} \rangle = 0$ because of $\mathbf{y}_{i,j} = \mathbf{0}_M$, where $\mathbf{0}_M \in \mathbb{R}^M$ denotes the zero vector. Then, $\bar{d}_{i,j}$ satisfies:

$$\bar{d}_{i,j} = h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \bar{\mathbf{z}}_{i,j}) = \begin{cases} 0 & : \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 > \epsilon_i \\ 1 & : \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \leq \epsilon_i \end{cases},$$

because $\beta \geq \epsilon_i \geq \|\mathbf{x}_j - \mathbf{x}_i\|_2^2$, which does not depend on $\bar{\mathbf{Z}}_i$. Therefore, $d_{i,j}^{(0)} = h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{0}_M) \in \{0, 1\}$ and $\mathbf{Z}_i^{(1)} = \mathcal{T}_{r, \gamma} \left\{ \mathbf{X} - \mathbf{x}_i \mathbf{1}_N^T \right\} \mathbf{D}_i^{(0)}$ is the optimal solution for (16). $\square$

Next, let us define the following schemes with regard to the k th iteration of the first-iteration statements in Algorithm 1 with $\beta^{(k)}, \gamma^{(k)}, r^{(k)}$ for $k \geq 0$,

$$\begin{cases} d_{i,j}^{(k)} = \bar{d}_{i,j} \text{ for } (i,j) \in \mathcal{I}^2, \\ \mathbf{Z}_i^{(k)} = \bar{\mathbf{Z}}_i \text{ for } i \in \mathcal{I}, \\ \mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\text{argmin}} \mathbf{x}^T (\mathbf{L}^{(k)} \otimes \mathbf{I}_{M,M}) \mathbf{x} - 2\mathbf{x}^T \mathbf{c}^{(k)} \\ \quad \text{s.t. } (\mathbf{X})_{m,n} = (\mathbf{X}^{(0)})_{m,n} \text{ for } (m,n) \in \Omega \\ \quad \|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \leq \epsilon_i \text{ if } d_{i,j}^{(k)} > 0, \end{cases} \quad (19)$$

where $\bar{d}_{i,j}$ and $\bar{\mathbf{Z}}_i$ are the t th elements of the sequences obtained by the schemes (18) with $\beta^{(k)}, \gamma^{(k)}, r^{(k)}, \mathbf{X}^{(k)}$, and vector $\mathbf{c}^{(k)}$ as:

$$\begin{aligned}
\mathbf{c}^{(k)} &= \begin{bmatrix} \mathbf{c}_1^{(k)T} & \mathbf{c}_2^{(k)T} & \dots & \mathbf{c}_M^{(k)T} \end{bmatrix}^T \in \mathbb{R}^{MN} \\
\mathbf{c}_l^{(k)} &= \left(\tilde{\mathbf{D}}^{(k)} \odot \tilde{\mathbf{Z}}_l^{(k)} - \left(\tilde{\mathbf{D}}^{(k)} \odot \tilde{\mathbf{Z}}_l^{(k)} \right)^T \right) \mathbf{1}_N \\
&\in \mathbb{R}^N \text{ for } l = 1, 2, \dots, M.
\end{aligned}$$

Here, $\tilde{\mathbf{D}}^{(k)} \in \mathbb{R}^{N \times N}$ and $\tilde{\mathbf{Z}}_l^{(k)} \in \mathbb{R}^{N \times N}$ denote matrices defined as $(\tilde{\mathbf{D}}^{(k)})_{ij} = d_{i,j}^{(k)}$ and $(\tilde{\mathbf{Z}}_l^{(k)})_{ij} = (\mathbf{Z}_i^{(k)})_{lj}$ for $(i,j) \in \mathcal{I}^2$, and the graph Laplacian $\mathbf{L}^{(k)}$ is:

$$\mathbf{L}^{(k)} = \text{diag} \left(\hat{\mathbf{D}}^{(k)} \mathbf{1}_N \right) - \hat{\mathbf{D}}^{(k)}$$

where $\hat{\mathbf{D}}^{(k)} \in \mathbb{R}^{N \times N}$ denotes a matrix whose every element is given by $(\hat{\mathbf{D}}^{(k)})_{ij} = d_{i,j}^{(k)2} + d_{j,i}^{(k)2}$.

Lemma 3 For $k \geq 0$, $\mathbf{L}^{(k)}$ satisfies $\text{kernel}(\mathbf{L}^{(k)}) \supseteq \text{kernel}(\mathbf{L}^{(k+1)})$.

Proof Since a vector $\mathbf{a} \in \text{kernel}(\mathbf{L}^{(k)})$ satisfies:

$$\mathbf{a}^T \mathbf{L}^{(k)} \mathbf{a} = \sum_{(i,j) \in \mathcal{I}^2} (d_{ij}^{(k)^2} + d_{ji}^{(k)^2}) (a_i - a_j)^2 = 0,$$

$\text{kernel}(\mathbf{L}^{(k)})$ is written as:

$$\begin{aligned} & \text{kernel}(\mathbf{L}^{(k)}) \\ &= \left\{ \mathbf{a} \in \mathbb{R}^N \mid a_i = a_j \text{ for } (i,j) \text{ s.t. } d_{ij}^{(k)} + d_{ji}^{(k)} > 0 \right\}. \end{aligned}$$

Since $d_{ij}^{(k+1)}$ and $d_{ij}^{(k)}$ generated by the schemes (18) and (19) satisfy $d_{ij}^{(k+1)} > 0$ when $d_{ij}^{(k)} > 0$, $\mathbf{L}^{(k)}$ satisfies $\text{kernel}(\mathbf{L}^{(k)}) \supseteq \text{kernel}(\mathbf{L}^{(k+1)})$. $\square$

Now, let us describe the properties of the sequences generated by Algorithm 1 $\{X^{(k)}\}, \{Z_i^{(k)}\}, \{d_{ij}^{(k)}\}$. We define the evaluation function:

$$\begin{aligned} & \nu(k_1, k_2, k_3, k_4) = \\ & g_{\beta^{(k_1)}, \gamma^{(k_1)}, r^{(k_1)}} \left(X^{(k_2)}, \{Z_i^{(k_3)}\}_{i \in \mathcal{I}}, \{d_{ij}^{(k_4)}\}_{(i,j) \in \mathcal{I}^2} \right) \end{aligned}$$

and replace the linear-constraint condition $(X^{(k)})_{m,n} = (X^{(0)})_{m,n}$ for $(m,n) \in \Omega$ with $A\mathbf{x}^{(k)} = \mathbf{b}$, where $\mathbf{b} \in \mathbb{R}^{|\Omega|}$ denotes a vector whose elements are observed values $\{(X^{(0)})_{m,n}\}_{(m,n) \in \Omega}$ and $A \in \{0, 1\}^{|\Omega| \times MN}$ denotes a selector matrix.

Theorem 1 The sequences $\{X^{(k)}\}, \{Z_i^{(k)}\}$ and $\{d_{ij}^{(k)}\}$ converge to the limit points $\bar{X}, \bar{Z}_i$, and $\bar{d}_{ij}$ under repetition of the iteration schemes of (19) when $\text{kernel}(\tilde{\mathbf{L}}^{(0)}) \cap \nu(A) = \{\mathbf{0}_{MN}\}$, where $\tilde{\mathbf{L}}^{(k)} = \mathbf{L}^{(k)} \otimes \mathbf{I}_{M,M}$.

Proof The scheme (15) can be written as:

$$\begin{aligned} & \underset{\mathbf{x}}{\text{argmin}} \quad \mathbf{x}^T (\mathbf{L} \otimes \mathbf{I}_{M,M}) \mathbf{x} - 2\mathbf{x}^T \mathbf{c} \\ & \text{subject to} \quad A\mathbf{x} = \mathbf{b} \\ & \quad \|\mathbf{Q}_{i,j}\mathbf{x}\|_2^2 - \epsilon_i \leq 0 \text{ if } d_{ij} > 0, \end{aligned}$$

where $\mathbf{Q}_{i,j} \in \mathbb{R}^{M \times MN}$ denotes a matrix defined as $\mathbf{Q}_{i,j} = \mathbf{q}_{ij}^T \otimes \mathbf{I}_{M,M}$ and $\mathbf{q}_{ij} \in \mathbb{R}^N$ is defined such that the i th element is 1, the j th element is -1 , and the others are 0 ($\mathbf{Q}_{i,j}$ satisfies $\|\mathbf{Q}_{i,j}\mathbf{x}\|_2^2 = \|\mathbf{x}_j - \mathbf{x}_i\|_2^2$ for $\mathbf{x} \in \mathbb{R}^{MN}$). Since $\mathbf{x}^{(k+1)}$ satisfies the following KKT condition for $\nu(k, k+1, k, k)$:

$$\begin{cases} \tilde{\mathbf{L}}^{(k)} \mathbf{x}^{(k+1)} - \mathbf{c}^{(k)} + \lambda^{(k+1)} A^T A \mathbf{x}^{(k+1)} \\ + \sum_{d_{ij}^{(k)} > 0} \mu_{ij}^{(k+1)} \mathbf{Q}_{i,j}^T \mathbf{Q}_{i,j} \mathbf{x}^{(k+1)} = \mathbf{0}_{MN}, \\ A\mathbf{x}^{(k+1)} = \mathbf{b}, \\ \mu_{ij}^{(k+1)} (\|\mathbf{Q}_{i,j}\mathbf{x}^{(k+1)}\|_2^2 - \epsilon_i) = 0 \text{ for } d_{ij}^{(k)} > 0, \\ \mu_{ij}^{(k+1)} \geq 0, \end{cases}$$

where $\lambda^{(k+1)}$ and $\mu_{ij}^{(k+1)}$ denote the KKT multipliers, $v(k_1, k_2, k_3, k_4)$ satisfies:

$$\begin{aligned} & 2v(k, k, k, k) - 2v(k, k+1, k, k) \\ &= \mathbf{x}^{(k)T} \tilde{\mathbf{L}}^{(k)} \mathbf{x}^{(k)} - \mathbf{x}^{(k+1)T} \tilde{\mathbf{L}}^{(k)} \mathbf{x}^{(k+1)} \\ & \quad - 2\mathbf{x}^{(k)T} \mathbf{c}^{(k)} + 2\mathbf{x}^{(k+1)T} \mathbf{c}^{(k)} \\ &= (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)})^T \tilde{\mathbf{L}}^{(k)} (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}) \\ & \quad - 2\mathbf{x}^{(k)T} \sum_{d_{ij}^{(k)} > 0} \mu_{ij}^{(k+1)} \mathbf{Q}_{ij}^T \mathbf{Q}_{ij} \mathbf{x}^{(k+1)} \\ & \quad + 2\mathbf{x}^{(k+1)T} \sum_{d_{ij}^{(k)} > 0} \mu_{ij}^{(k+1)} \mathbf{Q}_{ij}^T \mathbf{Q}_{ij} \mathbf{x}^{(k+1)}, \end{aligned}$$

where the second equality uses the fact that $\mathbf{A}\mathbf{x}^{(k)} = \mathbf{b}$. Since $\|\mathbf{Q}_{ij}\mathbf{x}^{(k+1)}\|_2^2 = \epsilon_i$ when $\mu_{ij}^{(k+1)} > 0$,

$$\begin{aligned} & v(k, k, k, k) - v(k, k+1, k, k) \\ &= \frac{1}{2} (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)})^T \tilde{\mathbf{L}}^{(k)} (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}) \\ & \quad + \sum_{d_{ij}^{(k)} > 0} \mu_{ij}^{(k+1)} \left\{ \epsilon_i - \mathbf{x}^{(k)T} \mathbf{Q}_{ij}^T \mathbf{Q}_{ij} \mathbf{x}^{(k+1)} \right\} \\ &\geq \frac{1}{2} (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)})^T \tilde{\mathbf{L}}^{(k)} (\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)}). \end{aligned}$$

The second inequality uses:

$$\left| \langle \mathbf{Q}_{ij}\mathbf{x}^{(k)}, \mathbf{Q}_{ij}\mathbf{x}^{(k+1)} \rangle \right| \leq \|\mathbf{Q}_{ij}\mathbf{x}^{(k)}\|_2 \|\mathbf{Q}_{ij}\mathbf{x}^{(k+1)}\|_2 \leq \epsilon_i.$$

Obviously, $v(k, k+1, k, k) \geq v(k+1, k+1, k, k)$ because the parameters $\{\beta^{(k)}, \gamma^{(k)}, r^{(k)}\}$ decrease the objective function (17), and $v(k+1, k+1, k, k) \geq v(k+1, k+1, k+1, k+1)$ from Lemma 1. Since the sequence $\{v(k, k, k, k)\}$ generated by (19) converges to a limit point because of:

$$\begin{aligned} v(k, k, k, k) &\geq v(k, k+1, k, k) \\ &\geq v(k+1, k+1, k, k) \\ &\geq v(k+1, k+1, k+1, k+1) \\ &\vdots \\ &\geq -\beta_{\max} N^2, \end{aligned}$$

$\mathbf{x}^{(k)} - \mathbf{x}^{(k+1)} \rightarrow \mathbf{0}_{MN}$ when $k \rightarrow \infty$ and $v(0, 0, 0, 0) < \infty$ because each $\mathbf{L}^{(k)}$ satisfies $\text{kernel}(\tilde{\mathbf{L}}^{(k)}) \cap \text{kernel}(\mathbf{A}) = \{\mathbf{0}_{mn}\}$ for $k \geq 0$ if $\text{kernel}(\tilde{\mathbf{L}}^{(0)}) \cap \text{kernel}(\mathbf{A}) = \{\mathbf{0}_{MN}\}$ from Lemma 3. $\mathbf{X}^{(k)}$ reaches a limit point $\bar{\mathbf{X}}$; then, the sequence $\{\mathbf{Z}_i^{(k)}\}$ and $\{d_{ij}^{(k)}\}$ converges to limit points $\bar{\mathbf{Z}}_i$ and $\bar{d}_{ij}$ with a fixed $\bar{\mathbf{X}}$ from Lemma 1. $\square$

Finally, some improvements to Algorithm (1) are offered in this section. First, the dimension of the LDDM is unknown in actual applications, although Algorithm 1 requires a suitable r . In order to solve this issue, we adopt a method that estimates the dimension r based on the ratio of the singular value σ_r/σ_1 , just as [5] did for each column $i \in \mathcal{I}$. Second, we consider ways to reduce the computational complexity. Two key possibilities are considered: one is to ignore the quadratic-constraint condition $(\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 - \epsilon_i) d_{ij} \leq 0$ when we update $\mathbf{X}$ and the other is to update $\mathbf{X}$ for only the columns in the i th

neighborhoods, for example, by minimizing the only i th Frobenius norm term of (17) $\|(X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i - \mathbf{Z}_i\|_F^2$ with regard to the column $\mathbf{x}_i$, which is expected to work like a stochastic gradient-descent algorithm. Furthermore, this paper utilizes the parameter $\beta = \max_i \epsilon^{(i)}$ because the update schemes (18) yield limit points for $\mathbf{Z}_i$ and $d_{i,j}$ only once for each $i \in \mathcal{I}$ from Lemma 2. Thus, this paper proposes a heuristic algorithm for reducing the calculation time, as shown in Algorithm 2. There, the parameters satisfy $1 > \alpha^{(0)} \geq \alpha^{(1)} \geq \dots \geq \alpha_{\min} > 0$ for $k = 0, 1, \dots, k_{\max}$ and $\delta > 0$, just as in [5]. We consider here the time and space complexities of Algorithm 2. The major computa-

Algorithm 2 LLRA using a stochastic-gradient-descent-like algorithm (LLRASGD)

Require: $X^{(0)}$, Ω , $\{\alpha^{(k)}\}_{k=0}^{k_{\max}}$, $\{\epsilon_i\}_{i \in \mathcal{I}}$, δ , η , $k_{\max}$

```

1:  $X \leftarrow X^{(0)}$ 
2:  $\mathbf{Z}_i \leftarrow \mathbf{0}_{M,N}$  for  $i \in \mathcal{I}$ 
3:  $\beta \leftarrow \max_i \epsilon^{(i)}$ 
4:  $k \leftarrow 0$ 
5: repeat
6:   for  $i \in \mathcal{I}$  do
7:      $d_{i,j} \leftarrow h_{\beta, \epsilon_i}(\mathbf{x}_j, \mathbf{x}_i, \mathbf{z}_{i,j})$  for  $j \in \mathcal{I}$ 
8:      $\mathbf{D}_i \leftarrow \text{diag}([d_{i,1} \dots d_{i,N}]^T)$ 
9:      $\boldsymbol{\sigma} \leftarrow \text{svd}((X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i)$ 
10:     $r \leftarrow \underset{r}{\text{argmax}} \sigma_r \geq \alpha^{(k)} \sigma_1$ 
11:     $\gamma \leftarrow \delta \sigma_r$ 
12:     $\mathbf{Z}_i \leftarrow \mathcal{T}_{r,\gamma} \{(X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i\}$ 
13:     $X \leftarrow X(\mathbf{I}_{N,N} - \mathbf{D}_i) + (\mathbf{Z}_i + \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i$ 
14:     $(X)_{m,n} \leftarrow (X^{(0)})_{m,n}$  for  $(m, n) \in \Omega$ 
15:  end for
16: until  $k < k_{\max}$  or  $\|X_{\text{old}} - X\|_F / \|X\|_F \leq \eta$ 

```

Ensure: X .

tional cost of Algorithm 2 is derived from computing the singular value decomposition of $(X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i$ for all $i = 1, 2, \dots, N$ at each iteration. For simplicity, this paper assumes that the number of non-zero vectors of $(X - \mathbf{x}_i \mathbf{1}_N^T) \mathbf{D}_i$ is M for each iteration and each i . Then, since the algorithm requires the singular value decomposition of the $M \times M$ matrix, the time and space complexities of Algorithm 2 are $O(M^3 N)$ and $O(M^2)$ for each iteration. As written in [20], since the method VMC [15] requires the time complexity $O(N^3 + MN^2)$ and the space complexity $O(N^2)$, the time and space complexities of Algorithm 2 are lower than those of VMC when the numbers of rows M and columns N satisfy $M^3 < N^2$. Hence, Algorithm 2 is effective for datasets such as those used in Section 5.2.

5 Results and discussion

5.1 Synthetic data

This section presents several numerical examples for the matrix completion problem (1). In this section, each i th column of $X^{(0)}$ is generated by $\mathcal{F}_p : \mathbb{R}^r \mapsto \mathbb{R}^M$ with mapping function (3) as:

$$\mathbf{x}_i^{(0)} = \mathcal{F}_p(\mathbf{y}_i^{(0)}) = \mathbf{U}_p \boldsymbol{\psi}_p(\mathbf{y}_i^{(0)}). \quad (20)$$

Using $\mathbf{U}_p \in \mathbb{R}^{M \times \binom{r+p}{p}}$ and $\mathbf{Y}^{(0)} = [\mathbf{y}_1^{(0)} \mathbf{y}_2^{(0)} \cdots \mathbf{y}_N^{(0)}] \in \mathbb{R}^{r \times N}$ generated by an i.i.d. continuous uniform distribution whose supports are $[-0.5, 0.5]$ and $[-1, 1]$, the elements of $\mathbf{Y}^{(0)}$ are normalized as $\max |(Y^{(0)})_{ij}| = 1$. The index set Ω is generated using the Bernoulli distribution with the given probability q , for which an index (i, j) belongs to Ω . This paper uses relative recovery error as:

$$\text{RE} [\%] = \frac{\|\mathbf{X}^{(0)} - \mathbf{X}\|_F}{\|\mathbf{X}^{(0)}\|_F} \times 100$$

to evaluate each algorithm. All numerical experiments were run in MATLAB 2017b on a PC with an Intel Core i7 3.1 GHz CPU, 8 GB of RAM, and no swap memory.

This paper applies some low-rank matrix completion algorithms including singular value thresholding (SVT) [1], the fixed-point continuation algorithm (FPCA) [2], the short IRLS-0 (sIRLS-0) method [3], IPMS [5], the nonlinear matrix completion method VMC [15], and the proposed LLRASGD method to several matrix completion problems with $M = 100$, $N = 4,000$, and $d = 3, 5$ for (20). A maximum iteration number of $k_{\max} = 1000$ is used for LLRA, IPMS, sIRLS-0, and SVT, and the termination condition is $\|\mathbf{X}^{(k)} - \mathbf{X}^{(k+1)}\|_F / \|\mathbf{X}^{(k+1)}\|_F \leq 10^{-5}$ for all algorithms. The parameters for LLRASGD and IPMS are given as $\alpha^{(k)} = 10^{-\frac{4k}{k_{\max}}}$ and $\delta = 10^{-2}$; those for SVT are $\tau^{(k)} = 10^{-2}\sigma_1^{(k)}$; those for sIRLS-0 and VMC are $\gamma^{(k)} = 10^{2-\frac{6k}{k_{\max}}}$; those for VMC are $p = 0.5$ and $d = 3$; and those for FPCA are $\tau = 1$ and $\mu^{(k)} = (0.25)^k \geq \bar{\mu} = 10^{-8}$. The condition $\sigma_l^{(k)} \geq 10^{-2}\sigma_1^{(k)}$ is used to choose r for FPCA in this paper. We set the initial value of $\{(X)_{m,n}\}_{(m,n) \notin \Omega}$ to 0 for SVT, FPCA, sIRLS-0, and IPMS. The values $\mathbf{X}$ and ϵ_i are estimated using IPMS for VMC and LLRASGD such that the total number satisfying the condition $\|\mathbf{x}_j - \mathbf{x}_i\|_2^2 \leq \epsilon_i$ equals 50 with an estimated value of $\mathbf{X}$ using IPMS for LLRASGD.

The results are shown in Tables 1, 2, and 3 for $q \in \{0.2, 0.3, 0.4\}$ and $r \in \{2, 3, 4, 5, 6\}$. As can be seen, estimation accuracy of LLRASGD is better than the others for $r = 5, 6$,

Table 1 Results of the algorithms for problem (1) with $p = 3$ for (20)

Problem		Algorithm					
q	r	SVT	FPCA	sIRLS-0	IPMS	VMC	LLRASGD
0.2	2	17.21%	2.33%	1.32%	1.40%	0.08%	0.92%
0.2	3	32.61%	32.19%	16.04%	22.38%	7.61%	4.04%
0.2	4	49.25%	49.31%	47.79%	44.48%	36.07%	5.77%
0.2	5	59.42%	59.52%	64.72%	56.03%	76.64%	17.59%
0.2	6	66.76%	66.85%	71.41%	65.03%	85.83%	39.90%
0.3	2	9.50%	0.28%	0.09%	0.09%	0.03%	0.03%
0.3	3	17.54%	9.27%	1.97%	2.11%	0.10%	0.12%
0.3	4	34.45%	34.48%	22.45%	26.05%	7.11%	1.24%
0.3	5	55.30%	55.40%	59.46%	51.47%	70.70%	17.72%
0.3	6	66.76%	66.85%	71.41%	65.03%	85.83%	39.90%
0.4	2	5.98%	0.05%	0.07%	1.75%	0.04%	0.07%
0.4	3	10.47%	0.50%	0.15%	0.15%	0.07%	0.03%
0.4	4	22.52%	22.39%	5.94%	9.70%	0.21%	0.29%
0.4	5	36.18%	36.30%	31.42%	29.10%	7.29%	2.96%
0.4	6	45.81%	45.91%	49.21%	40.89%	34.89%	9.87%

Numbers in boldface show the best results

Table 2 Results of the algorithms for problem (1) with $p = 5$ for (20)

Problem		Algorithm					
q	r	SVT	FPCA	sIRLS-0	IPMS	VMC	LLRASGD
0.2	2	21.62%	18.55%	7.22%	10.09%	0.73%	2.81%
0.2	3	44.03%	44.07%	36.67%	38.53%	24.87%	5.42%
0.2	4	60.19%	60.31%	63.43%	58.02%	73.79%	15.60%
0.2	5	69.95%	70.06%	73.60%	69.35%	86.51%	41.26%
0.2	6	76.71%	76.79%	80.82%	77.19%	88.10%	61.95%
0.3	2	12.60%	2.06%	0.92%	0.95%	0.08%	0.12%
0.3	3	29.30%	29.23%	21.08%	21.28%	3.84%	0.69%
0.3	4	46.92%	47.05%	47.22%	41.78%	31.61%	5.36%
0.3	5	58.52%	58.63%	61.18%	55.62%	72.23%	19.60%
0.3	6	66.40%	66.50%	69.97%	65.28%	80.52%	38.15%
0.4	2	8.47%	1.27%	0.11%	0.13%	0.08%	0.04%
0.4	3	19.70%	19.36%	12.07%	12.37%	0.10%	0.11%
0.4	4	36.67%	36.79%	34.88%	30.57%	9.14%	2.46%
0.4	5	48.97%	49.08%	51.09%	44.84%	43.18%	11.38%
0.4	6	57.47%	57.58%	60.38%	55.16%	69.21%	25.69%

Numbers in boldface show the best results

$q = 0.2, 0.3, 0.4$ and $d = 3, 5, 7$, and $r = 3, 4, 5, 6$ and $q = 0.2$ especially. Figures 2, 3, and 4 compare all algorithms with $q = 0.3$. In Figs. 2 and 3, the recovery errors of LLRASGD tend not to decay more than other algorithms. From this result, the proposed method is more effective for the case in which the missing rate or the latent dimension is high.

5.2 CMU motion capture data

This paper considers the matrix completion on motion capture data, which consists of time-series trajectories of human motions such as running and jumping. Similar to [15], this paper uses the trial #6 of subject #56 of the CMU motion capture dataset. The data has measurements from $M = 62$ sensors at 6784 time instants, which the data matrix is

Table 3 Results of the algorithms for problem (1) with $p = 7$ for (20)

Problem		Algorithm					
q	r	SVT	FPCA	sIRLS-0	IPMS	VMC	LLRASGD
0.2	2	25.55%	24.37%	11.61%	15.93%	5.10%	3.51%
0.2	3	51.01%	51.11%	48.52%	47.36%	44.97%	6.88%
0.2	4	66.77%	66.89%	69.89%	66.87%	83.93%	28.63%
0.2	5	76.12%	76.23%	80.19%	76.82%	87.89%	55.80%
0.2	6	81.96%	82.02%	85.72%	82.98%	88.71%	74.73%
0.3	2	14.42%	6.83%	3.16%	4.15%	0.05%	0.24%
0.3	3	36.47%	36.53%	30.56%	29.60%	11.59%	1.48%
0.3	4	54.34%	54.47%	56.77%	51.32%	57.27%	11.91%
0.3	5	65.44%	65.55%	69.06%	64.35%	79.28%	32.70%
0.3	6	72.42%	72.51%	76.19%	72.65%	81.99%	53.60%
0.4	2	9.51%	2.55%	0.76%	1.05%	0.05%	0.06%
0.4	3	26.18%	26.19%	19.59%	19.02%	1.64%	0.34%
0.4	4	44.22%	44.34%	46.79%	39.62%	26.51%	6.23%
0.4	5	56.14%	56.25%	59.02%	53.75%	63.94%	21.16%
0.4	6	63.81%	63.91%	67.08%	63.09%	73.61%	39.25%

Numbers in boldface show the best results

known as high-rank matrix. In this experiment, the sequence is downsampled by factor 2, which the data matrix has $M = 62$ rows and $N = 3392$ columns. Then, the elements of the data matrix were randomly observed with the ratio $q \in \{0.1, 0.2, 0.3, 0.4\}$, and this paper applied the matrix completion algorithms with the same parameters which is used in the Section 5.1.

The average recovery errors for 10 trials are shown in Fig. 5. Similar to the results on synthetic data, the estimation accuracy of LLRASGD is better than the others. Especially, the recovery errors of LLRASGD are much lower than the others when the missing ratio is

very high (such as $q = 0.1, 0.2$). From these results, the proposed method is more effective for not only synthetic data but also real-world dataset.

The average computational time costs for all observed ratio $q \in \{0.1, 0.2, 0.3, 0.4\}$ are shown in Table 4. This result indicates that the computation time of LLRASGD is about 200 to 500 times longer than that of the conventional MRM methods, and the computation time of VMC is about 2.4 times longer than that of LLRASGD for the same number of iterations. However, VMC and LLRASGD have sufficiently high estimation accuracy even with a small number of iterations. Figure 6 shows the results of VMC and LLRASGD

Table 4 The average computational time cost (second) of the algorithms for CMU motion capture data recovery

Algorithm					
SVT	FPCA	sIRLS-0	IPMS	VMC	LLRASGD
2.43s	2.19s	5.66s	4.51s	2871.54s	1273.35s

for the maximum iteration number of $k_{\max} \in \{10, 20, 40, 100, 1000\}$ in the observed ratio $q = 0.2$. As can be seen in Fig. 6, the recovery error of LLRASGD converges sufficiently in $k_{\max} = 40$. In this maximum iteration number, although the computational time of LLRASGD is about 16 times longer than that of IPMS, the recovery error of LLRASGD is less than half that of IPMS. We can also see that the recovery error of LLRASGD is less than that of VMC for all $k_{\max} \in \{10, 20, 40, 100, 1000\}$.

6 Conclusion

This paper proposed a local low-rank approach (LLRA) for a matrix-completion problem in which the columns of the matrix belong to an LDDM. The convergence properties of this approach were also presented. The proposed method is based on the idea of tangent hyperplanes of dimension equal to that of the LDDM with respect to each column of the matrix. It is assumed that each hyperplane is of low dimension and that the sum of the rank of each local submatrix with respect to each column belonging to the set of nearest neighborhoods of each column is minimized. Numerical examples show that the proposed algorithm offers higher accuracy for matrix completion than other algorithms in the case where each column vector is given by a p th order polynomial mapping of a latent feature. In particular, the proposed method is suitable when the order p and the dimension of the latent space are high.

Appendix

In this section, this paper introduces the truncated nuclear norm and the minimization technique by IPMS [5].

The truncated nuclear norm $\|Z\|_{*,r}$ is defined with the k th biggest singular value σ_k of Z as:

$$\|Z\|_{*,r} = \sum_{k=r+1}^M \sigma_k.$$

The truncated nuclear norm is used as the substitution function of matrix rank, and the solution of $\frac{1}{2}\|Y - Z\|_F^2 + \gamma\|Z\|_{*,r}$ with a given matrix Y and a given parameter $\gamma > 0$ can be solved as follows:

$$\begin{aligned} Z &= \underset{Z}{\operatorname{argmin}} \frac{1}{2}\|Y - Z\|_F^2 + \gamma\|Z\|_{*,r} \\ &= \mathcal{T}_{r,\gamma}(Y), \end{aligned}$$

where $\mathcal{T}_{r,\gamma}$ denotes the matrix-shrinkage operator defined as:

$$\begin{aligned} \mathcal{T}_{r,\gamma}(Y) &= U \operatorname{diag}(\sigma_{r,\gamma}) V^T \\ \sigma_{r,\gamma} &= [\sigma_1 \cdots \sigma_r (\sigma_{r+1} - \gamma)_+ \cdots (\sigma_M - \gamma)_+]^T \end{aligned}$$

and $(c)_+ = \max(0, c)$ with regard to the singular value decomposition $Y = U \operatorname{diag}(\sigma) V^T$.

In the matrix completion problem (2), the IPMS algorithm solves the relaxation problem by iterating the following update schemes:

$$\begin{cases} Z & \leftarrow \mathcal{T}_{r,\gamma}(X), \\ (X)_{m,n} & \leftarrow (X^{(0)})_{m,n} \text{ for } (m, n) \in \Omega. \end{cases}$$

Since the truncated nuclear norm requires the value of r regarding with a matrix rank, the IPMS estimates a matrix rank r during iterations by using the scheme:

$$r \leftarrow \underset{r}{\operatorname{argmin}} \sigma_r \text{ s.t. } \sigma_r \leq \alpha \sigma_1,$$

where $0 \leq \alpha < 1$ is a given constant. The details of the IPMS algorithm are written in [5].

Abbreviations

LDDM: Low-dimensional differentiable manifold; MRM: Matrix rank minimization; LDLS: Low-dimensional linear subspace; UoLS: Union of linear subspaces; VMC: Variety-based matrix completion; LLRA: Local low-rank approach; SVT: Singular value thresholding; FPCA: Fixed-point continuation algorithm; IRLS: Iterative reweighted least squares; IPMS: Iterative partial matrix shrinkage

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of this manuscript.

Authors' contributions

All authors have contributed equally. All authors have read and approved the final manuscript.

Funding

This work was supported by the JSPS KAKENHI Grant Number JP19H02163.

Availability of data and materials

Please contact the author for data requests.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Faculty of Computer and Information Science, Hosei University, 3-7-2 Kajino-cho Koganei-shi, Tokyo, Japan. ²Faculty of Information Science and Technology, Tokai University, 4-1-1 Kitakaname Hiratsuka-shi, Kanagawa, Japan. ³Faculty of Engineering, Tokyo University of Science, 6-3-1 Nijuku Katsushika-ku, Tokyo, Japan.

Received: 19 August 2020 Accepted: 20 January 2021

Published online: 12 February 2021

References

1. J. F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **20**(4), 1956–1982 (2010)
2. D. Goldfarb, S. Ma, Convergence of fixed point continuation algorithms for matrix rank minimization. *Found. Comput. Math.* **11**(2), 183–210 (2011)
3. K. Mohan, M. Fazel, Iterative reweighted algorithms for matrix rank minimization. *J. Mach. Learn. Res. (JMLR)*. **13**(1), 3441–3473 (2012)
4. D. Zhang, Y. Hu, J. Ye, X. Li, X. He, in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit*, Matrix completion by truncated nuclear norm regularization, (2012), pp. 2192–2199
5. K. Konishi, K. Uruma, T. Takahashi, T. Furukawa, Iterative partial matrix shrinkage algorithm for matrix rank minimization. *Signal Process.* **100**, 124–131 (2014)
6. J. Gotoh, A. Takeda, K. Tono, DC formulations and algorithms for sparse optimization Problems. *J. Math. Program.* **169**(1), 141–176 (2018)
7. X. Guan, C. T. Li, Y. Guan, Matrix factorization with rating completion: an enhanced SVD model for collaborative filtering recommender systems. *IEEE Access*. **5**, 27668–27678 (2017)
8. M. Verhaegen, A. Hansson, N2SID: nuclear norm subspace identification of innovation models. *Autom.* **72**, 57–63 (2016)
9. K. H. Jin, J. C. Ye, Annihilating filter-based low-rank Hankel matrix approach for image inpainting. *IEEE Trans. Image Process.* **24**(11), 3498–3511 (2015)
10. Q. Zhao, D. Meng, Z. Xu, W. Zuo, Y. Yan, L_1 -norm low-rank matrix factorization by variational Bayesian method. *IEEE Trans. Neural Netw. Learn. Syst.* **26**(4), 825–839 (2015)
11. B. Eriksson, L. Balzano, R. Nowak, High rank matrix completion. *Int. Conf. Artif. Intell. Stat.* **22**, 373–381 (2012)
12. C. Yang, D. Robinson, R. Vidal, in *Proc. of the 32th Int. Conf. on Machine Learning (PMLR)*, Sparse subspace clustering with missing entries, vol. 37 (PMLR, Lille, 2015), pp. 2463–2472
13. C.-G. Li, R. Vidal, A structured sparse plus structured low-rank framework for subspace clustering and completion. *IEEE Trans. Signal Process.* **64**(24), 6557–6570 (2016)
14. E. Elhamifar, in *Proc. 28th Adv. Neural Inf. Process. Syst.*, High-rank matrix completion and clustering under selfexpressive models (Curran Associates Inc., Barcelona, 2016), pp. 73–81
15. G. Ongie, R. Willett, R. D. Nowak, L. Balzano, in *Proc. of the 34th Int. Conf. on Machine Learning (PMLR)*, Algebraic variety models for high-rank matrix completion, vol. 70 (PMLR, Sydney, 2017), pp. 2691–2700
16. G. Ongie, L. Balzano, D. Pimentel-Alarcón, R. Willett, R. D. Nowak, Tensor methods for nonlinear matrix completion. *arXiv preprint arXiv:1804.10266* (2018). <https://arxiv.org/abs/1804.10266>
17. R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (GPCA). *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(12), 1945–1959 (2005)
18. X. Alameda-Pineda, E. Ricci, Y. Yan, N. Sebe, in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit*, Recognizing emotions from abstract paintings using non-linear matrix completion (IEEE, Las Vegas, 2016), pp. 5240–5248
19. J. Fan, T. W. S. Chow, Non-linear matrix completion. *Pattern Recogn.* **77**, 378–394 (2018)
20. J. Fan, M. Udell, in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit*, Online high rank matrix completion (IEEE, Long Beach, 2019), pp. 8682–8690
21. J. Fan, Y. Zhang, M. Udell, in *Proc. AAAI*, Polynomial matrix completion for missing data imputation and transductive learning, vol. 34 (AAAI Press, New York, 2020), pp. 3842–3849
22. J. Fan, J. Cheng, Matrix completion by deep matrix factorization. *Neural Netw.* **98**, 34–41 (2018)
23. B. Schölkopf, A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. (MIT Press, Cambridge, 2002). https://books.google.co.jp/books?id=y8ORL3DWt4sC&hl=ja&source=gbp_book_other_versions
24. S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding. *Sci.* **290**(5500), 2323–2326 (2000)
25. M. Winlaw, D. L. Samimi, A. Ghodsi, Robust locally linear embedding using penalty functions. *Int. Joint Conf. Neural Netw.*, 2305–2312 (2011)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.