Matrix Completion via Non-Convex Relaxation and Adaptive Correlation Learning

Xuelong Li, Fellow, IEEE, Hongyuan Zhang, and Rui Zhang, Member, IEEE

Abstract—The existing matrix completion methods focus on optimizing the relaxation of rank function such as nuclear norm, Schatten-*p* norm, *etc.* They usually need many iterations to converge. Moreover, only the low-rank property of matrices is utilized in most existing models and several methods that incorporate other knowledge are quite time-consuming in practice. To address these issues, we propose a novel non-convex surrogate that can be optimized by closed-form solutions, such that it empirically converges within dozens of iterations. Besides, the optimization is parameter-free and the convergence is proved. Compared with the relaxation of rank, the surrogate is motivated by optimizing an upper-bound of rank. We theoretically validate that it is equivalent to the existing matrix completion models. Besides the low-rank assumption, we intend to exploit the column-wise correlation for matrix completion, and thus an adaptive correlation learning, which is scaling-invariant, is developed. More importantly, after incorporating the correlation learning, the model can be still solved by closed-form solutions such that it still converges fast. Experiments show the effectiveness of the non-convex surrogate and adaptive correlation learning.

Index Terms—Artificial Intelligence, Pattern Recognition, Matrix Completion, Non-Convex Surrogate, Adaptive Correlation Learning, Parameter-Free Optimization.

1 INTRODUCTION

Matrix is a fundamental element in machine learning and the low-rank property of matrix has been applied in many practical applications [1]–[3]. Low-rank matrix completion (LRMC) [4], [5], aiming to recover a low-rank matrix according to the observed entries, plays an important role in many fields, such as image recovery [6], [7], recommendation systems [8], robust principal component analysis [9]– [11], multi-task learning [12]–[14], *etc*. The main motivation behind low-rank is from the observation that a part of principal components of a matrix usually contain most of the information, especially in optical imagery (shown in Figure 1). In other words, the distribution of singular values of an image matrix is often heavy-tailed.

The original LRMC model [4] intends to optimize the nuclear norm of matrix, the convex envelope of rank. To improve the performance, plenty of works focus on optimizing the nuclear norm and its variants such as truncated version [15], weighted version [16], *etc.* As the nuclear norm is the ℓ_1 -norm of singular values, the nuclear norm relaxed problem can be generalized to the Schatten-p norm. With 0 , the Schatten-<math>p norm approximates rank better than the nuclear norm. It should be emphasized that Schatten-p is not convex when 0 . In particular, solutions of LRMC with Schatten-<math>1/2 and Schatten-2/3 are derived in [17]–[19]. Additionally, several different surrogates are also developed to obtain a better approximation

of rank [20], [21]. Besides, diverse factorization models are derived from these surrogates. For instance, the factored nuclear norm [22], [23] transforms it into two Frobenius norm terms while RegL1 [24] tries to improve the robustness of the noisy model. Factored group-sparse regularization [25] proves that the Schatten-p norm is equivalent to the sum of two group-sparse norms. Bilinear model [26] shows that the Schatten-p norm can be converted into nuclear norms of two factored matrices. To improve the results of matrix completion, the models proposed in [27]-[29] incorporate the similarity as the prior information. However, the similarity is given as the prior information such that the model fails to work on general cases. Besides, several works [30], [31] focus on how to integrate various kinds of information. The main barrier of these hybrid models is inefficient optimization. They usually consume significant amounts of time to train, which results in unavailability in practice.

To optimize the proposed models, several optimization techniques are applied such as the semidefinite programming (SDP) [4], augmented Lagrange multiplier method (ALM) [24], alternative direction method of multipliers (ADMM) [32], re-weighted method [33], [34], etc. SDP is time-consuming especially when m and n are not tiny values (e.g., m = n = 100) [35]. ALM [36] and ADMM [32] are the most popular methods in matrix completion but it needs lots of iterations to converge. To accelerate the optimization, auxiliary variables are frequently introduced [24], [25] and the linearized ADMM [37] is widely applied since the direct subproblem of ADMM may have no closed-form solution. Non-factored models usually depend on singular value decomposition (SVD) which causes computational complexity $O(mn^2)$ per iteration. Factored models usually need O(mnd) time per iteration where d is the column number of factored matrices. For the noisy extension, the

The authors are with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, P.R. China. They are also with the Key Laboratory of Intelligent Interaction and Applications (Northwestern Polytechnical University), Ministry of Industry and Information Technology, Xi'an 710072, P. R. China.

This work is supported by The National Natural Science Foundation of China (No. 61871470).

Corresponding author: R. Zhang and X. Li

E-mail: li@nwpu.edu.cn; hyzhang98@gmail.com; ruizhang8633@gmail.com

proximal gradient method is widely used to solve the nonsmooth term, *i.e.*, the nuclear norm (denoted by $\|\cdot\|_*$). However, *d* is hard to set and experiments show that smaller *d* may lead to slower convergence in some cases. In other words, ADMM requires more iterations to converge, which implies expensive costs.

In sum, the existing LRMC models rely on parametric algorithms (*e.g.*, gradient-based methods, ADMM-based methods, *etc.*), which require lots of iterations to converge. Additionally, most of them only focus on the lowrank property. Different from the existing models, we propose a model with a novel Non-Convex surrogate and Adaptive corRelation Learning (*NCARL*) for LRMC problem, to achieve faster convergence and exploit the hidden information of the matrix. Besides the low-rank assumption, NCARL incorporates an adaptive correlation learning mechanism to exploit correlation and mines the potential information column-wisely. The main contributions are listed as follows:

- We aim to optimize an upper-bound of rank(·) via the full-rank factorization, which provides a novel non-convex surrogate. Compared with other factored methods, our model does not need the initial rank *d*. Surprisingly, it can be solved by closed-form solutions without linearization and auxiliary variables, such that its optimization is totally parameter-free.
- Besides the low-rank assumption, the potential information of columns is exploited by our model via learning column-wise correlation adaptively. Owing to the smooth surrogate, the model still has closedform solutions after incorporating the adaptive correlation learning, which implies the two parts are compatible.
- The proposed algorithm usually converges within 20 iterations such that it is competitive in the terms of efficiency compared with factored models, even though our model does not need the initial rank. Although factored models seem to need less time per iteration, they require a large number of iterations to converge. Experiments support the computational efficiency of our model.

1.1 Notations

In this paper, \boldsymbol{m}^i and \boldsymbol{m}_j denote the *i*-th row and *j*-th column of M, respectively. M^{\dagger} is the Moore-Penrose pseudo-inverse. $\mathcal{R}(M)$ represents the space spanned by columns of M. diag(\boldsymbol{m}) represents the diagonal matrix with diagonal entries \boldsymbol{m} . \mathbb{S}^n_+ and \mathbb{S}^n_{++} denote the set of positive semi-definite and positive definite $n \times n$ matrices, respectively. \odot represents the Hadamard product. Without additional statements, given a matrix $M \in \mathbb{R}^{m \times n}$, we assume $m \geq n$. Given a square matrix $Q \in \mathbb{R}^{n \times n}$ and non-zero binary vectors $\boldsymbol{p}, \boldsymbol{q} = \{0,1\}^n$ where $\|\boldsymbol{p}\|_0 = k_1$ and $\|\boldsymbol{q}\|_0 = k_2$, $[Q]_{\boldsymbol{p},\boldsymbol{q}} \in \mathbb{R}^{k_1 \times k_2}$ represents the sub-matrix where the *i*-th row and *j*-th column are deleted from Q if $p_i = 0$ and $q_j = 0$. Specially, for a vector \boldsymbol{v} , $[\boldsymbol{v}]_{\boldsymbol{p}}$ represents the sub-vector that removes the *i*-th entry from \boldsymbol{v} if $\boldsymbol{v}_i = 0$. 1 denotes the vector whose entries are all 1 and $\bar{\boldsymbol{p}} = \mathbf{1} - \boldsymbol{p}$. $\ell_{2,0}$ -norm and $\ell_{2,1}$ -norm of M are respectively defined as $\|M\|_{2,0} = \sum_{i=1}^m \|\mathbf{m}^i\|_2 \neq 0$] and $\|M\|_{2,1} = \sum_{i=1}^m \|\mathbf{m}^i\|_2$



Fig. 1. An illustration of singular values of a natural image.

where m^i is the *i*-th row of M. All proofs are summarized in appendix.

2 RELATED WORK

First, we provide the formal definition of LRMC. Suppose that we have observed some entries denoted by a matrix $M \in \mathbb{R}^{m \times n}$, where indexes are represented by Ω . Particularly, $M_{ij} = 0$ if $(i, j) \notin \Omega$. The known LRMC attempts to recover a low-rank matrix X from the observations. LRMC can be formulated as

$$\min_{\mathbf{v}} \operatorname{rank}(X), \quad s.t. \; X_{ij} = M_{ij} \; \forall (i,j) \in \Omega.$$
(1)

Assume that $m \ge n$ holds, then we could rewrite the constraint as a concise matrix form via introducing a filter matrix P as

$$P_{ij} = \begin{cases} 1, & (i,j) \in \Omega; \\ 0, & (i,j) \notin \Omega. \end{cases}$$
(2)

Accordingly, the LRMC problem can be rewritten as

$$\min_{X} \operatorname{rank}(X), \quad s.t. \ X \odot P = M, \tag{3}$$

where \odot represents the Hadamard product. Instead of solving the NP-hard problem caused by rank(\cdot), the existing models [4], [25], [38] aim to optimize a relaxed function $\varphi(\cdot)$ of rank(\cdot) such that the objective is converted into

$$\min \varphi(X), \quad s.t. \ X \odot P = M. \tag{4}$$

For instance, the classical LRMC model [4] uses the nuclear norm, $\|\cdot\|_*$, as $\varphi(\cdot)$. Max norm [39] is also investigated for LRMC. As $||X||_* = \sum_{i=1} \sigma_i$ where σ_i represents the singular value, an important variant is to employ the Schattenp norm [17]–[19], [25], [38], $||X||_{S_p} = (\sum_{i=1} \sigma_i^p)^{\frac{1}{p}}$. Based on these surrogates, some models design more complicated surrogates, such as truncated nuclear norm [15], weighted nuclear norm [16], etc. Besides, several models [27], [30], [31] also integrate other kinds of information. S³LR introduces the popular subspace exploitation into the mechanism, while the prior graph information is utilized in the model proposed by [27]. To accelerate the optimization and avoid searching the rank of matrix in all possible values, factored models [22], [23], [25] have been widely investigated. The core idea of factored models is to assume that recovered matrix can be factored as two small matrices. For example, the factored nuclear norm model [22] is defined as

$$\min_{X=AB, X \odot P=M} \frac{1}{2} (\|X\|_*) \Leftrightarrow \min_{AB \odot P=M} \frac{1}{2} (\|A\|_F^2 + \|B\|_F^2).$$
(5)

FGSR [25] aims to factorize Schatten-1/2 and Schatten-2/3 as two convex surrogates instead.

A well-known extension of LRMC is the noisy version. If the contaminated case regarding polluted observations is considered, *i.e.*, $M_{ij} = (X_{ij})_* + \varepsilon_{ij}$ where $(X_{ij})_*$ and ε_{ij} denote the true value and noise respectively, the recovery task indicates the simultaneous minimization of residuals and rank,

$$\min_{\mathbf{v}} \|X \odot P - M\|_F^2 + \gamma \cdot \varphi(X), \tag{6}$$

where γ is the hyper-parameter to leverage residuals and rank. Some works [15], [24] focus on improving the noisy model as well. Specifically, RegL1 [24] utilizes ℓ_1 -norm to replace the Frobenius norm and ensure $\{\varepsilon_{ij}\}_{i,j}$ sparse. To retrieve a simple discussion, we only focus on the noiseless case at first and the model can be easily extended into the noisy case.

3 PROBLEM REFORMULATION

Unlike most LRMC models, we do not relax $rank(\cdot)$ as the nuclear norm. Instead, for an arbitrary matrix $X \in \mathbb{R}^{m \times n}$, we can apply full-rank factorization and have

$$X = W^T U^T, (7)$$

where $U \in \mathbb{R}^{n \times n}$ is an orthogonal matrix. Note that the full-rank factorization is not unique. On the one hand, $\operatorname{rank}(X) = \operatorname{rank}(W) \leq ||W||_{2,0}$ since $\operatorname{rank}(W) \leq ||W||_{2,0}$ holds for any W. In other words, $\ell_{2,0}$ -norm is an upperbound of $\operatorname{rank}(\cdot)$. Therefore, we can optimize the following upper-bound as the objective function

$$\min_{W,U} \|W\|_{2,0}, \quad s.t. \ (W^T U^T) \odot P = M, U^T U = I.$$
(8)

On the other hand, $\{u_i\}_{i=1}^n$ denotes an orthonormal basis while w_j can be regarded as the coordinate of $(x^j)^T$ under $\{u_i\}_{i=1}^n$. The low-rank property of X indicates that only a few basis vectors are activated. In sum, $||W||_{2,0}$ is a rational replacement of rank(X). The following theorem rigorously shows the connection between the following problem and the original one.

Theorem 1. Problem (8) is equivalent to problem (3). In other words, $X_* = W_*^T U_*^T$ where X_* , W_* , and U_* are the optimal solutions of two problems, respectively.

Accordingly, we can focus on how to optimize problem (8). Likewise, since the optimization of $\ell_{2,0}$ -norm is NP-hard and $\ell_{2,1}$ -norm is convex envelope of it, problem (8) can be relaxed into

$$\min_{W,U} \|W\|_{2,1}, \quad s.t. \ (W^T U^T) \odot P = M, U^T U = I;$$

$$\Rightarrow \min_{W,U} \|W\|_{2,1}^2, \quad s.t. \ (W^T U^T) \odot P = M, U^T U = I.$$
(9)

The following theorem demonstrates that the above objective function can be converted into a smooth function which has continuous first-order derivative.

Theorem 2. Define Ψ and Ψ' as follows

$$\begin{cases} \Psi = \{ (X, D) | X \odot P = M, \operatorname{tr}(D^{\dagger}) = 1, D \in \mathbb{S}_{+}^{n} \}, \\ \Psi' = \{ (W, U) | (W^{T}U^{T}) \odot P = M, U^{T}U = I \}. \end{cases}$$
(10)

Algorithm 1 Algorithm to solve problem (11).

Input: Mask matrix *P*, observed entries *M*, perturbation coefficient $\delta = 10^{-6}$, and maximum iterations $t_m = 50$. $X \leftarrow M$. **repeat**

Update D by Eq. (15). $\hat{D} \leftarrow D + \delta I$. $F_i \leftarrow P_i \hat{D}^{-1} P_i$. Update X according to Eq. (20): $x^i \leftarrow m^i (F_i)^{p^i} + \hat{D}^{-1}$. until convergence or exceeding maximum iteration t_m . Output: Recovered matrix X.

Then problem (9) is equivalent to

$$\min_{X,D} \operatorname{tr}(XDX^T), \quad s.t. \ (X,D) \in \Psi.$$
(11)

Meanwhile, the relation of optimal solutions of the two problems can be established as

$$X_* = W_*^T U_*^T, D_* = U_* \Lambda U_*^T, \Lambda = \operatorname{diag}(\frac{\|\boldsymbol{w}_*^i\|_2}{\|W_*\|_{2,1}})^{\dagger}, \quad (12)$$

where $U_*\Lambda U_*^T$ is the eigenvalue decomposition of D_* ,

$$\begin{cases} (X_*, D_*) = \arg\min_{(X,D)\in\Psi} \operatorname{tr}(XDX^T), \\ (W_*, U_*) = \arg\min_{(W,U)\in\Psi'} \|W\|_{2,1}^2. \end{cases}$$
(13)

Proposition 1. *Problem (11) is non-convex. In particular, the subproblem regarding X is convex.*

In spite of the non-convexity, $tr(XDX^T)$ is smooth compared with $||X||_*$. In the following subsection, an efficient gradient-free algorithm, which can converge into the global optimum, is developed. Besides, in the next section, we will find that this surrogate is more compatible with additional mechanisms.

3.1 Optimization of Problem (11)

Since the problem is non-convex and the subproblem regarding X is convex, we optimize problem (11) by an alternative method. Inspired by [13], Theorem 3 provides a closed-form solution for the subproblem regarding D.

Theorem 3. If X is fixed as constant, the optimum of problem (11) is $||X||_{*}^2$ i.e.,

$$||X||_*^2 = \min_D \operatorname{tr}(XDX^T), \quad s.t. \operatorname{tr}(D^{\dagger}) = 1, D \in \mathbb{S}^n_+, \quad (14)$$

where the optimal D is given as

$$D = \left(\frac{(X^T X)^{\frac{1}{2}}}{\operatorname{tr}((X^T X)^{\frac{1}{2}})}\right)^{\dagger}.$$
 (15)

Accordingly, we can focus on how to optimize $\min_X \operatorname{tr}(XDX^T)$ subject to $X \odot P = M$. The Lagrangian function can be represented as

$$\mathcal{L} = \operatorname{tr}(XDX^T) + \operatorname{tr}(V^T(X \odot P - M)), \qquad (16)$$

where $V \in \mathbb{R}^{m \times n}$ denotes Lagrange multipliers. Note that only $|\Omega|$ multipliers are needed due to the fact that $X_{ij}P_{ij} = M_{ij}$ always holds if $(i, j) \notin \Omega$. The KKT conditions are

$$\begin{cases} \nabla_X \mathcal{L}(X, V) = 2XD + V \odot P = 0, \\ X \odot P = M. \end{cases}$$
(17)

To obtain the closed-form solution, we replace D with $\hat{D} = D + \delta I$ ($\delta > 0$) since \hat{D} is invertible. Note that $\boldsymbol{x}^i \odot \boldsymbol{p}^i = \boldsymbol{x}^i \operatorname{diag}(\boldsymbol{p}^i)$. To keep notations uncluttered, let $P_i = \operatorname{diag}(\boldsymbol{p}^i)$. Accordingly, we have

$$\begin{cases} \boldsymbol{x}^{i} = -\frac{1}{2} \boldsymbol{v}^{i} P_{i} \hat{D}^{-1} \\ \boldsymbol{x}^{i} P_{i} = \boldsymbol{m}^{i} \end{cases} \Rightarrow -\frac{1}{2} \boldsymbol{v}^{i} P_{i} \hat{D}^{-1} P_{i} = \boldsymbol{m}^{i}.$$
(18)

Let $F_i = P_i \hat{D}^{-1} P_i$. Lemma 1 shows that $[F_i]_{p^i,p^i}$ is invertible.

Lemma 1. For any $Q \in \mathbb{S}_{++}^n$ and any non-zero binary vector $p \in \{0,1\}^n$, $[Q]_{p,p}$ is positive definite.

Definition 1. Given a binary vector $\mathbf{p} \in \{0,1\}^n$ and a square matrix $Q \in \mathbb{R}^{n \times n}$, suppose that $[Q]_{\mathbf{p},\mathbf{p}}$ is invertible. We define $Q^{\mathbf{p}+} \in \mathbb{R}^{n \times n}$ as a matrix which satisfies $[Q^{\mathbf{p}+}]_{\mathbf{p},\mathbf{p}} = [Q]_{\mathbf{p},\mathbf{p}}^{-1}$ and the other entries are 0.

Accordingly, V and X can be approximately solved by

$$\begin{cases} v_i = -2m^i(F_i)^{p^i +} \\ x^i = m^i(F_i)^{p^i +} \hat{D}^{-1}. \end{cases}$$
(19)

However, the residual caused by Eq. (19) can not be guaranteed to be upper-bounded when $\exists i, \hat{D}_{ii} = 0$. To address this issue, define $H_i \in \mathbb{R}^{n \times n}$ as a diagonal matrix such that $[H_i]_{ii} = \mathbb{1}[\hat{D}_{ii} \neq 0]$ and the other diagonal entries are 1. If the solution is modified as

$$\begin{cases} v_i = -2m^i (\hat{F}_i)^{p_i +}, \\ x^i = m^i (\hat{F}_i)^{p_i +} (H_i \hat{D} H_i)^{p_i +}, \end{cases}$$
(20)

where $\hat{F}_i = P_i \hat{D}^{h_i +} P_i$ and $h_i = \text{diag}(H_i)$, Theorem 4 shows that the residual between approximate solution and real solution is related to δ .

Theorem 4. Let \hat{X} and \hat{V} denote the approximate solutions defined in Eq. (20). There exists a constant u, which is independent on δ , such that $\hat{X} \odot P = M$ and $\|\nabla_X \mathcal{L}(\hat{X}, \hat{V})\| \le 2\delta u \|M\|$.

Therefore, with $\delta \to 0$, $\nabla_X \mathcal{L} \to 0$. In our experiments, we set $\delta = 10^{-6}$. To compute D, we need $O(mn^2)$ time. Since we have to compute the inverse of \hat{D} , $O(n^3)$ are needed to calculate X at least. Recall that $m \ge n$ and thus the computational complexity is $O(mn^2)$. The algorithm to solve problem (11) is summarized in Algorithm 1. In Section 5, we can see that our method can converge within 20 iterations. Compared with other methods that require hundreds even thousands of iterations to converge, the consuming time of the proposed model is less even though the computational complexity of each iteration is $O(mn^2)$. Theoretically, combining with Theorem 3 and 4, we have the following proposition,

Corollary 1. If $\delta \rightarrow 0$, then Algorithm 1 will approach the global minimum of problem (11).

3.2 Recovery Bound

As the recovery bound, which exposes the upper bound of errors between the recovered matrix and the real matrix, is an important part in field of matrix completion, we also provide recovery bounds about our model. Inspired by Theorem 3, the following theorem is the critical part to show recovery bounds.



Fig. 2. Two images from MSRC-v2 which are used for image recovery.

Theorem 5. Problem (9) is equivalent to

$$\min_{X} \|X\|_{*}, \ s.t. \ X \odot P = M.$$
(21)

Therefore, recovery bounds for the nuclear relaxation model [4], [39] can be applied to our model. For instance, the famous recovery bound proposed by [4] is available for our model:

Lemma 2. [4] Let $X_0 \in \mathbb{R}^{m \times n}$ be the real matrix, and $N = \max(m, n)$. Suppose that $|\Omega|$ entries of X_0 are observed uniformly at random. Then there are constants c_1 and c_2 such that if $|\Omega| \ge c_1 N^{5/4} r \log N$, the unique minimizer equals with X_0 with probability at least $1 - c_2 N^{-3} \log N$.

Remark: One may concern the significance of our relaxation since the equivalence between problem (9) and the nuclear norm surrogate. Roughly speaking, the main merits include the tractable optimization and scalability of our model. On the one hand, the optimization is completely parameter-free. Compared with gradient-based methods and ADMM-based methods, no hyper-parameters (e.g., step-size in gradient-based methods, increasing coefficient in ADMM-based methods, etc.) are required, which leads to the simple optimization. On the other hand, our model is well formulated from the mathematical aspect, since the mathematical form is common in machine learning. Therefore, it is easy to incorporate other mechanisms without obvious expenses and extra derivation. As we show in the next section, the optimization of the whole model is analogous to the one of problem (9) after introducing the correlation learning of columns. Contrastively, the existing models that attempt to integrate the additional information of matrices and the nuclear norm surrogate (e.g., LRFD [30], $S^{3}LR$ [31], *etc.*) needs a great deal of time to train.

3.3 Noisy Case

Similarly, the noisy matrix completion can be modeled by adding an extra regularization,

$$\min_{X,D} \frac{\mathcal{J}}{\|X \odot P - M\|_F^2 + \gamma \operatorname{tr}(XDX^T)}, \quad (22)$$
s.t. $\operatorname{tr}(D^{\dagger}) = 1, D \in \mathbb{S}_+^n.$

Without any additional proofs, Theorem 2 and 3 can be easily extended into the noisy case. The solution of the subproblem regarding D is given by Eq. (15). Since the

subproblem to solve X is an unconstrained problem, we can take the derivative and set it to 0,

$$\nabla_X \mathcal{J} = 2X \odot P - 2M \odot P + 2\gamma X D = 0.$$
 (23)

Similar with Eq. (20), X can be solved analytically by adding a perturbation,

$$\boldsymbol{x}^{i} = \boldsymbol{m}^{i} P_{i} (P_{i} + \gamma \hat{D})^{-1}.$$
(24)

Motived by Theorem 4, $\|\nabla_X \mathcal{J}\| \le 2\delta u \|M\|$ always holds if *X* is approximately computed by Eq. (24). Obviously, the computational cost of every iteration is $O(mn^2)$. Figure 5 gives a vivid illustration of the result of the noisy model.

Interestingly, if $D \in \mathbb{S}_{++}^n$, then problem (22) can be regarded as a Maximum A Posterior (*MAP*) model from the probabilistic perspective. In the probabilistic model, Xis a random variable, and M is regarded as supervised information. Hence, the objective is to solve

$$\max_{X} p(X|M) \Leftrightarrow \max_{X} p(M|X) \cdot P(X).$$
(25)

Recall that $M_{ij} = (X_{ij})_* + \varepsilon_{ij}$ if $(i, j) \in \Omega$. Suppose that $\varepsilon_{ij} \sim \mathcal{N}(0, I)$. Therefore, $p(M_{ij}|X_{ij})$ can be modeled as $\mathcal{N}(X_{ij}, I)$ if $(i, j) \in \Omega$. To be convenient, $p(M_{ij}|X_{ij}) = 1$ if $(i, j) \notin \Omega$. We further assume that the prior distribution of X is a matrix Gaussian distribution, *i.e.*, $p(X) = \mathcal{MN}(0, I, \frac{1}{2\gamma}D^{-1})^{-1}$. Take the log of Eq. (25),

$$\log p(M|X) + \log(X) = \sum_{i,j} \log p(M_{ij}|X_{ij}) + \log \mathcal{MN}(0, I, \frac{1}{2\gamma}D^{-1}) = \sum_{(i,j)\in\Omega} \log \mathcal{N}(0, 1) + \log \mathcal{MN}(0, I, \frac{1}{2\gamma}D^{-1}) = - \|X \odot P - M\|_F^2 - \gamma \operatorname{tr}(XDX^T) - \frac{n}{2}\log|D^{-1}| + C,$$
(26)

where C denotes the constant term. Therefore, Eq. (18) is equivalent to

$$\min_{X,D} \|X \odot P - M\|_F^2 + \gamma \operatorname{tr}(X D X^T) + \frac{n}{2} \log |D^{-1}|.$$
 (27)

Note that $\log |D^{-1}|$ can be viewed as a penalty term such that eigenvalues of D^{-1} will not be too large. To simplify the model, the constraint $\operatorname{tr}(D^{-1})$ is used to replace $|D^{-1}|$, which can restrict eigenvalues as well. Specifically speaking, if $\operatorname{tr}(D^{-1}) = 1$, $|D^{-1}| < \operatorname{tr}(D^{-1}) = 1$ always holds. In sum, Eq. (22) is thus derived from the probabilistic aspect.

4 ADAPTIVE CORRELATION LEARNING: COMPLE-TION CORRELATION OF COLUMNS

Although the matrix completion is usually formulated as a brief optimization problem, the concise formulation may hide some important properties in practice. In this section, we will design a rational mechanism to utilize some additional information to improve the performance of matrix

1. $\mathcal{MN}(E, \Sigma_1, \Sigma_2) = \frac{\exp(-\frac{1}{2}\operatorname{tr}(\Sigma_2^{-1}(X-E)^T\Sigma_1^{-1}(X-E)))}{(2\pi)^{mn/2}|\Sigma_1|^{n/2}|\Sigma_2|^{m/2}}$ where $E \in \mathbb{R}^{m \times n}$ is the mean, $\Sigma_1 \in \mathbb{R}^{m \times m}$ and $\Sigma_2 \in \mathbb{R}^{n \times n}$ represent covariance matrix of row and column, respectively.

Algorithm 2 Algorithm to solve NCARL (defined in Eq. (33)).

 $\begin{aligned} & \lim_{l_{ij}} \leftarrow \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2. \\ & \text{Update } S \text{ by Eq. (32).} \\ & S \leftarrow \frac{S+S^T}{2}. \\ & \text{Update } D \text{ by Eq. (15).} \\ & L \leftarrow D_S - S. \\ & \hat{Q} \leftarrow D + \alpha L + \delta I. \\ & F_i \leftarrow P_i \hat{Q}^{-1} P_i. \end{aligned}$

Update X according to Eq. (20): $x^i \leftarrow m^i(F_i)^{p^i} + \hat{Q}^{-1}$. until convergence or exceeding maximum iteration t_m . Output: Recovered matrix X.

completion. Meanwhile, it also verifies the compatibility of the surrogate proposed in the previous section.

In practical applications, columns probably have underlying connections with each other. For instance, in recommendation systems, a column vector may represent the preferences of a user to diverse items. According to the obtained information (*i.e.*, the observed entries), we can judge whether two users are similar. Therefore, the two recovered user vectors, which are highly similar, should be more analogous. Inspired by this, we have the following assumption,

Assumption 1. Two vectors are similar if the Euclidean distance between them is small. Formally, given x_i , x_j , and x_k , x_i are more similar with x_j compared with x_k if $||x_i - x_j||_2 < ||x_i - x_k||_2$.

Suppose that we have obtained similarities of some pairs of column vectors as the prior knowledge. Formally, let $S_{ij} \ge 0$ denote the similarity of $(\boldsymbol{x}_i)_*$ and $(\boldsymbol{x}_j)_*$ where $(\boldsymbol{x}_i)_*$ is the *i*-th column of the optimal matrix X_* . Clearly, S should be symmetric. Naturally, the recovered matrix Xshould keep these similarities. More formally,

$$\min_{X} \sum_{i,j=1}^{n} S_{ij} \| \boldsymbol{x}_{i} - \boldsymbol{x}_{j} \|_{2}^{2} \Leftrightarrow \min_{X} \operatorname{tr}(XLX^{T}), \quad (28)$$

where $L = D_S - S$ and D_S is a diagonal matrix where $(D_S)_{ii} = \sum_{j=1}^{n} S_{ij}$. The matrix, L, is usually called Laplacian matrix in spectral graph theory [40]. Hence, the noiseless model is formulated as

$$\min_{X,D} \operatorname{tr}(XDX^T) + \alpha \operatorname{tr}(XLX^T),$$

s.t. $X \odot P = M, \operatorname{tr}(D^{\dagger}) = 1, D \in \mathbb{S}^n_+.$ (29)

However, the similarity matrix S is frequently unavailable in most matrix completion scenarios. Inspired by self-supervised learning [41], we compute S adaptively in the training phase. Suppose that the recovered matrix is $X^{(t)}$ at step t. Then, the similarity S is updated according to $X^{(t)}$. Accordingly, the key is how to obtain a rational similarity matrix S based on Assumption 1. Intuitively, for every column vector, the number of correlated columns should not

Algorithm 3 Algorithm to solve NCARL-noisy.

Input: The tradeoff hyper-parameter α and γ , sparsity k, mask matrix P, observed entries M, perturbation coefficient $\delta = 10^{-6}$, and maximum iterations $t_m = 50$. $X \leftarrow M$.

repeat $l_{ij} \leftarrow ||\mathbf{x}_i - \mathbf{x}_j||_2^2$. Update *S* by Eq. (32). $S \leftarrow \frac{S+S^T}{2}$. Update *D* by Eq. (15). $L \leftarrow D_S - S$. $\hat{Q} \leftarrow P + D + \delta I + \alpha L$. Update *X* according to Eq. (20): $\mathbf{x}^i \leftarrow \mathbf{m}^i P_i \hat{Q}^{-1}$. until convergence or exceeding maximum iteration t_m . Output: Recovered matrix *X*.

be too large. In other words, s^i should be sparse. Besides, we normalize *S* such that $S\mathbf{1} = \mathbf{1}$. In this paper, we design a novel point-wise similarity learning model, which can precisely control the sparsity degree. The designed correlation learning model is

$$\min_{S1=1,S\geq 0} \sum_{i,j=1}^{n} S_{ij} \|\boldsymbol{x}_{i}^{(t)} - \boldsymbol{x}_{j}^{(t)}\|_{2}^{2} + \|\boldsymbol{\mu}^{T}S\|_{F}^{2}$$

$$\Leftrightarrow \min_{S1=1,S\geq 0} \operatorname{tr}(X^{(t)}LX^{(t)T}) + \|\boldsymbol{\mu}^{T}S\|_{F}^{2},$$
(30)

where $\mu \in \mathbb{R}^n$ represents hyper-parameters for *n* columns. Theorem 6 states that the *n* hyper-parameters can be converted into one parameter if we assume that the sparsity degrees of columns are identical. According to Theorem 6, Corollary 2 demonstrates that the correlation learning mechanism is scaling-invariant.

Theorem 6. In problem (30), each row of the optimal S is k-sparse (i.e., $\forall i, \|s^i\|_0 = k$) if μ satisfies

$$\frac{1}{2}(k\boldsymbol{l}_{i}^{(k)} - \sum_{v=1}^{k} \boldsymbol{l}_{i}^{(v)}) < \mu_{i}^{2} \le \frac{1}{2}(k\boldsymbol{l}_{i}^{(k+1)} - \sum_{v=1}^{k} \boldsymbol{l}_{i}^{(v)}), \quad (31)$$

where $i = 1, 2, \dots, n$, $l_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2$ and $\boldsymbol{l}_i^{(k)}$ is the kth smallest value in $\{l_{ij}\}_{j=1}^n$. Moreover, if $\mu_i^2 = \frac{1}{2}(k \boldsymbol{l}_i^{(k+1)} - \sum_{v=1}^k \boldsymbol{l}_i^{(v)})$, S can be solved by

$$S_{ij} = \left(\frac{\boldsymbol{l}_i^{(k+1)} - \boldsymbol{l}_{ij}}{\sum\limits_{v=1}^k \boldsymbol{l}_i^{(k+1)} - \boldsymbol{l}_i^{(v)}}\right)_+.$$
 (32)

Corollary 2. The adaptive graph learning process is scalinginvariant. In other words, if $\hat{X} = kX$, then the learned similarity \hat{S} satisfies $\hat{S} = S$.

To keep *S* symmetric, we set $S \leftarrow (S + S^T)/2$. By unifying the above correlation learning and the matrix completion model proposed in Section 3, the final objective of *NCARL* can be formulated as

$$\min_{X,D,S} \operatorname{tr}(X(D+\alpha L)X^T) + \alpha \|\boldsymbol{\mu}^T S\|_F^2,$$

s.t. $X \odot P = M, \operatorname{tr}(D^{\dagger}) = 1, D \in \mathbb{S}^n_+, S\mathbf{1} = \mathbf{1}, S \ge 0.$ (33)

It should be pointed out that the added correlation learning mechanism does not impact the optimization of *D*. The only difference is that *D* should be replaced by $D + \alpha L$ in Eq. (20) when optimizing *X*. Accordingly, the adaptive correlation learning is compatible with the non-convex surrogate such that problem (33) can be optimized by the alternative method as well. The entire procedure is summarized in Algorithm 2.

Similarly, the model for noisy case, *NCARL-noisy*, is given as

$$\min_{X,D} \|X \odot P - M\|_F^2 + \gamma \operatorname{tr}(XDX^T) + \alpha(\operatorname{tr}(XLX^T) + \|\boldsymbol{\mu}^T S\|_F^2),$$
(34)
s.t. $\operatorname{tr}(D^{\dagger}) = 1, D \in \mathbb{S}_+^n, S\mathbf{1} = \mathbf{1}, S \ge 0.$

Analogous to Eq. (24), we have $x^i = m^i P_i (P_i + \gamma D + \alpha L)^{-1}$. The whole procedure to solve NCARL-noisy is similar with Algorithm 2, and the concrete algorithm is stated in Algorithm 3. The computational cost of updating X and D does not change. The optimization of S requires $O(n^2 \log n + n^2)$ and thus, the time complexity of each iteration is still $O(mn^2)$.

5 EXPERIMENTS

In this section, we test the performance of NCARL and its noisy extension on several real datasets. The experimental details, results, and analysis are reported as follows. All codes are implemented by MATLAB 2019b.

5.1 Baseline Methods

In our experiments, 8 representative models are compared with our model, including classical nuclear norm [4], factored nuclear norm (*F*-Nuclear) [22], Schatten-p norm (S_p norm) [38], weighted nuclear norm (WNNM) [16], factored model with ℓ_1 -norm (*RegL1*) [24], factored group-sparse regularization (FGSR) [25], LRFD [30], and S³LR [31]. Specifically, S³LR introduces the sparse subspace learning [42]. Note that the subspace learning can also be reformulated as the compatible form of our proposed surrogate. The authors [31] use the traditional nuclear norm and develop the optimization based on LADMM [37]. For Schatten-p norm, we set p as 1/2 and 2/3. In particular, F-nuclear, RegL1, and FGSR are factored models, which are usually more efficient on large scale matrices. The hyper-parameters of these methods are searched in the same way recorded in corresponding papers. All codes are downloaded from the authors' homepages.

5.2 Datasets and Evaluation Metric

NCARL is first tested on 3 synthetic matrices with different scale, including 500×300 , 1500×1000 , and 4000×3000 . The rank of these matrices are set as 100, 200, and 300, respectively. As all models performs well on the clean data when the missing entries is not too large, the missing rate, ϵ , is set as 0.8 and 0.9. Moreover, we add tiny noise on 20 percent of observed entries to disturb models. Then, we test different models on two real images (denoted by *Image-1*)





(e) Image-1 with block mask



(f) Recovered Image-1



(g) Image-2 with block mask



(h) Recovered Image-2

Fig. 3. Recovered images under different kinds of noise. For random noise, the missing rate, ϵ , is set as 0.5, while the missing block is 50×50 for the block mask. The colorful image is obtained by recovering images from three channels individually.

TABLE 1 MSE and consuming seconds on three synthetic matrices. ϵ represents the missing rate.

Size	ϵ	Metric	Nuclear	F-Nuclear	$S_{1/2}$	$S_{2/3}$	WNNM	RegL1	FGSR	LRFD	S ³ LR	NCARL
5 00 900	0.0	MSE	0.0428	0.0515	0.0410	0.0347	0.0958	0.0473	0.0549	0.1921	0.0387	0.0344
	0.0	Time	3.9667	4.9850	3.7477	3.6688	18.274	34.134	9.7684	116.76	69.783	3.2725
000 × 000	0.0	MSE	0.0446	0.3484	0.0991	0.0709	0.0952	0.0492	0.0556	0.2996	0.0406	0.0397
	0.9	Time	3.7660	6.6923	1.6861	1.6662	11.778	40.323	7.4459	119.77	85.002	1.6175
1500 × 1000	0.8	MSE	0.0409	0.1079	0.0433	0.0357	0.1006	0.0584	0.0377	0.5788	0.0385	0.0328
		Time	37.993	33.935	56.461	52.000	610.76	165.63	32.941	> 3000	1612.7	26.262
1500 × 1000	0.0	MSE	0.0404	0.5451	0.0599	0.0450	0.1028	0.0527	0.0453	0.6978	0.0389	0.0326
	0.9	Time	42.730	36.529	29.242	28.973	237.84	126.18	86.099	> 3000	1556.8	27.176
	0.8	MSE	0.0383	0.1640	0.0445	0.0343	-	-	0.0630	-	-	0.0255
4000×3000		Time	2042.9	374.65	1290.3	1393.1	-	-	713.15	-	-	229.50
	0.9	MSE	0.0360	0.5988	0.0529	0.0408	-	-	0.0474	-	-	0.0287
		Time	2007.2	369.95	564.55	547.60	-	-	633.48	-	-	189.91



 943×1682 matrix while MovieLens-1M is a 6040×3952 matrix. For the images, we convert the image into grey pictures to obtain a matrix and mask ϵ (0 < ϵ < 1) of pixels randomly. For MovieLens-100K, since only parts of entries are known, we mask ϵ of known entries randomly. The normalized mean-squared-error (MSE) is employed to measure the performance of various models. The definition of MSE is

$$MSE = \sqrt{\frac{\sum_{(i,j)\in\Upsilon\backslash\Omega} (X_{ij} - (X_{ij})_*)^2}{\sum_{(i,j)\in\Upsilon\backslash\Omega} (X_{ij})_*^2}},$$
 (35)

Fig. 4. Experimental results on MovieLens-1M with different ϵ . Note that the nuclear norm consumes more than 3000s on this dataset.

and Image-2) from MSRC-v2² [43] and two recommendation system datasets, MovieLens-100K and MovieLens-1M³ [44]. The colorful images, which are stored as $320 \times 240 \times 3$ tensors in computer, are shown in Figure 2. Two MovieLens datasets are two large matrices. Specifically, MovieLens-100K is a

where X_* is the true matrix and Υ is the set of known entries. Besides the recovery quality, the consuming time is also a vital metric in our paper.

5.3 Experimental Setup

There are two hyper-parameters (α and k) to tune in NCARL. It should be emphasized that the two hyperparameters are introduced via incorporating correlation learning into the proposed surrogate that can be optimized via a parameter-free algorithm. Therefore, the hyper-

^{2.} research.microsoft.com/en-us/projects/objectclassrecognition/

^{3.} grouplens.org/datasets/movielens/

TABLE 2

MSE on two noiseless datasets. *Image-1* and *Image-2* denote the images chosen from MSRC-V2 while *ML-100K* denotes the recommendation system dataset, MovieLens-100K. *ε* represents the missing rate.

	ε	Nuclear	F-Nuclear	$S_{1/2}$	$S_{2/3}$	WNNM	RegL1	FGSR	LRFD	S ³ LR	NCARL
Image-1	0.4	0.1285	0.1285	0.1298	0.1259	0.1930	0.1285	0.1460	0.1824	0.1223	0.1267
	0.5	0.1338	0.1338	0.1359	0.1311	0.1926	0.1338	0.1492	0.2455	0.1268	0.1309
	0.6	0.1396	0.1396	0.1441	0.1380	0.1967	0.1396	0.1555	0.2785	0.1328	0.1385
	0.7	0.1468	0.1468	0.1559	0.1464	0.2065	0.1468	0.1621	0.3374	0.1397	0.1451
Image-2	0.4	0.1880	0.1880	0.1967	0.1877	0.2911	0.1880	0.2152	0.2933	0.1786	0.1766
	0.5	0.1913	0.1913	0.2031	0.1925	0.2847	0.1913	0.2169	0.7527	0.1813	0.1804
	0.6	0.1975	0.1975	0.2119	0.2002	0.2908	0.1975	0.2176	0.8971	0.1886	0.1892
	0.7	0.2068	0.2067	0.2266	0.2117	0.3003	0.2067	0.2247	0.9773	0.1973	0.1991
ML-100K	0.4	0.2799	0.3570	0.3818	0.3342	0.3877	0.2763	0.3147	0.3962	0.2746	0.2733
	0.5	0.2859	0.3782	0.4183	0.3650	0.3915	0.2815	0.3189	0.4219	0.2803	0.2787
	0.6	0.2936	0.4050	0.4465	0.3918	0.4005	0.2878	0.3650	0.4463	0.2917	0.2848
	0.7	0.3083	0.4503	0.4927	0.4386	0.4050	0.3002	0.3670	0.4511	0.3029	0.2953

TABLE 3 Consuming seconds of various models.

	ϵ	Nuclear	F-Nuclear	$S_{1/2}$	$S_{2/3}$	WNNM	RegL1	FGSR	LRFD	S ³ LR	NCARL
IMG-1	0.4	1.5624	0.7689	4.9093	4.8850	9.4661	15.7755	1.1416	32.5176	45.3176	0.6179
	0.5	1.5574	0.7913	4.0239	4.1486	17.7139	18.2746	1.1728	33.8724	44.3514	0.9211
	0.6	1.6201	0.8880	3.3858	3.3786	21.0650	19.8961	1.0485	44.7844	46.4201	1.0525
	0.7	1.5986	0.6103	2.6372	2.6854	13.8199	22.8290	0.9676	46.2483	47.9612	0.9640
IMG-2	0.4	1.7186	1.1745	7.1439	7.5223	11.1063	15.4871	3.2615	67.1065	69.5478	1.3193
	0.5	1.9493	1.4213	7.7525	7.5294	12.2682	17.2063	2.8130	71.0738	66.4808	1.4625
	0.6	1.9046	1.7328	5.4397	5.5025	25.9807	18.8793	2.5439	94.2695	62.4828	1.4859
	0.7	1.8639	1.0327	4.5917	4.3262	17.1553	21.3594	2.1404	91.5482	61.4658	1.2273
ML-100K	0.4	32.5527	38.8636	43.2083	42.8679	180.6862	1333.5766	53.9980	> 3000	> 3000	7.9339
	0.5	34.9918	39.2142	41.6637	41.5215	144.4549	1538.9810	46.3761	> 3000	> 3000	8.0379
	0.6	34.2989	38.8689	40.5688	40.2747	109.9873	1677.0022	44.2300	> 3000	> 3000	7.7124
	0.7	34.7444	38.5751	38.5936	38.6748	86.8998	1920.2985	38.8642	> 3000	> 3000	8.4206

parameters in NCARL do not contradict our claim about parameter-free optimization. Specifically, α is searched from $\{10^0, 10^1, \dots, 10^4\}$ and k is searched from $\{5, 10, 20, 50\}$. Note that the perturbation coefficient δ is set as 10^{-6} and the maximum iteration t_m is set as 50. For factored models, the upper-bounds of rank are identical to each other. On synthetic datasets, the upper-bounds are set as the exact rank. On two images, they are fixed as 200 while they are set as 500 on MovieLens datasets. To ensure fairness, all methods with randomness are run 5 times and the average results are reported.

5.4 Experimental Results

5.4.1 Synthetic Datasets

The results on synthetic low-rank matrices are summarized in Table 1. Since WNNM, RegL1, LRFD, and S^3LR require too much time to converge, we use dash marks to represent the unavailability of these methods. The optimal and suboptimal results are bolded. From Table 1, NCARL shows the strong stability and impressive efficiency due to the fast convergence. It should be emphasized that S^3LR usually provides the competitive results but with too much time. It also provides convincing evidence about the effectiveness of extra mechanisms for matrix completion. The major barrier to introduce additional information is the complicated and inefficient optimization. Therefore, the proposed surrogate, which is compatible with diverse models in machine learning, is meaningful.

5.4.2 Real Datasets

For all real datasets, we run all methods under various missing rates (or named as the mask rate), $\epsilon \in$ $\{0.4, 0.5, 0.6, 0.7\}$. MSEs are recorded in Table 2. For the two images and MovieLens-100K, the best results and second ones are highlighted in the boldface, while the consuming time is reported in Table 3. Due to the large scale of MovieLens-1M, several compared methods (Nuclear, WNNM, RegL1, LRFD, and S³LR) become inefficient and thereby we only employ F-nuclear, Schatten-p norm, and FGSR as our main competitors. Although the classical nuclear norm model is quite time-consuming on MovieLens-1M, it acts as a baseline model. MSE and time on MovieLens-1M are shown in Figure 4. From Table 2, Table 3, and Figure 4, we conclude that NCARL obtains preferable performance on all datasets with the least time. As we expect, factored models like FGSR and F-Nuclear are more efficient especially on two MovieLens datasets. Although the performance of S³LR is usually impressive compared with other competitors due to the additional subspace exploration, its time cost is extremely expensive. Specifically, it needs more than 3000s to converge, which is



Fig. 5. Recovered images employing noisy model with different γ . r is the average rank of recovered images from three channels. The mask rate, ϵ , is set as 0.5.

TABLE 4 Ablation Experiments of NCARL ($\epsilon = 0.5$): Correlation represents the correlation preserving term and Adaptive denotes the adaptive learning mechanism.

	Correlation	Adaptive	Image-1		Image-2		MovieLens-100K		MovieLens-1M	
	Correlation		MSE	Time (s)	MSE	Time (s)	MSE	Time (s)	MSE	Time (s)
Method-A	×	×	0.1484	0.8297	0.2030	1.2890	0.3905	5.9932	0.6481	236.4501
Method-B	\checkmark	×	0.1476	0.8098	0.2027	1.2236	0.2922	7.0448	0.2721	241.2230
NCARL	\checkmark	\checkmark	0.1309	0.7302	0.1804	1.4625	0.2787	8.0379	0.2542	260.1178

unacceptable in practice, even though it returns the secondbest results. By contrast, NCARL performs significantly in terms of both MSE and time.

To test the recovery quality under different kinds of noise, images are contaminated by random noise and block noise. NCARL works well under both noises and the recovered images are shown in Figure 3. Besides, the performance of noisy extension, NCARL-noisy, is also illustrated in Figure 5. To establish the low-rank property (indicated by Theorem 5) of our surrogate meanwhile, we show the recovery results with different γ . Note that the closed-form solution is approximate, singular values of the recovery images approach 0 numerically. Therefore, we regard the singular values smaller than 10^{-3} as 0 and mark the number of singular values larger than 10^{-3} as the rank *r*. Obviously, the rank of the obtained image becomes smaller with the growth of γ from Figure 5.

To testify the rapid convergence of NCARL, we show the objective value of NCARL on Image-1 and MovieLens-100K in Figure 6. Clearly, NCARL converges fast within 20 iterations. This attractive trait is more apparent on images. Contrasively, the other models, which are solved by gradient-based methods or ADMM-based methods, require hundreds or thousands of iterations to converge. Accordingly, NCARL can be used in large scale datasets (like MovieLens-1M) as well, though the computational complexity of each iteration is $O(mn^2)$.

To study the effect of sparsity k, results with different k are shown in Figure 7, where α is assigned as the best



Fig. 6. Convergence of Algorithm 2 on Image-1 and MovieLens-100K with $\epsilon=0.5.$ Clearly, NCARL converges rapidly on both datasets, especially Image-1

value from $\{10^0, 10^1, \dots, 10^4\}$. In our experiments, α is set as 100 on Image-1 and 10^2 on MovieLens-100K. From Figure 7, it is not hard to find that the proposed model will get the best performance when k is not too large and the optimal k becomes larger with the increase of matrix scale. On Image-1, the best sparsity is 10 while k = 100 will lead to the best MSE on MovieLens-100K. Moreover, the increase of k will not burden the time cost obviously.

5.5 Ablation Analysis

To test the impact of different parts, we design ablation experiments on all datasets. There are totally 2 mechanisms to testify, column correlation preserving term and adaptive correlation learning mechanism. Accordingly, we conduct



Fig. 7. The influence of sparsity k to MSE on Image-1 and MovieLens-100K with $\epsilon=0.5.$

experiments to study the role of two parts and the results are recorded in Table 4. On the one hand, we conclude that the correlation preserving is useful for LRMC, especially on recommendation system datasets. Specifically speaking, the correlation preserving decreases MSE by about 0.1 and 0.37 on MovieLens-100K and MovieLens-1M, respectively. On the other hand, the adaptive learning mechanism further promotes the performance of our model. Compared with the method only with the correlation preserving, the adaptive learning reduces MSE by about 0.2 on two MovieLens datasets.

In particular, the two mechanisms do not burden the time cost significantly. In contrast, S³LR, which introduces sparse subspace learning into the nuclear norm model, requires much more time to train than the original nuclear model.

6 CONCLUSION

In this paper, we propose a novel model for low-rank matrix completion. Rather than the nuclear norm, a nonconvex surrogate is developed. Although the surrogate is non-convex, it is easy to optimize and extend since the optimization consists of multiple closed-form solutions. Based on the proposed relaxation, we introduce an adaptive correlation learning to explore the underlying information of the matrix, which is inspired by recommendation systems. Although the computational complexity of each iteration is $O(mn^2)$, the algorithm converges so fast that it needs less time than the existing methods. We conduct experiments on 2 real images and 2 recommendation system datasets and the superiority of our model is supported on both recovery quality and consuming time. In the future work, we will focus on the investigation about the convergence rate since the rapid convergence is only verified empirically.



Fig. 8. Proof sketch of the main theoretical results.

APPENDIX A PROOFS

In this part, proofs of the above theorems and propositions are elaborated successively.

A.1 Proof of Theorem 1

Proof. Let

$$\begin{cases}
\mathcal{J}_1 = \min_{U,W} \|W\|_{2,0}, \quad s.t. \ (W,U) \in \Psi', \\
\mathcal{J}_2 = \min_X \operatorname{rank}(X), \quad s.t. \ X \odot P = M.
\end{cases}$$
(36)

On the one hand, for the optimal (W_*, U_*) , we can easily find an $X = (U_*W_*)^T \in \{X | X \odot P = M\}$, which indicates $\mathcal{J}_2 \leq \mathcal{J}_1.$

On the other hand, for the optimal X_* , we can apply the full-rank factorization on X_* , and thus we can get $\mathcal{J}_1 \leq \mathcal{J}_2$.

Overall, we have $\mathcal{J}_1 = \mathcal{J}_2$.

A.2 Proof of Theorem 2

The following theorem demonstrates that the above objective function can be converted into a smooth function that has a continuous first-order derivative.

Lemma 3. Given k non-negative constants c_i and variable $x \in$ $\{\boldsymbol{x}|\boldsymbol{x}^T\boldsymbol{1}=1, \boldsymbol{x}>0\}$, the following inequality holds

$$\sum_{i=1}^{k} \frac{c_i^2}{x_i} \ge (\sum_{i=1}^{k} c_i)^2.$$
(37)

The equality holds if and only if $x_i = \frac{c_i}{\sum_{i=1}^k c_i}$.

Proof. Let $f(\boldsymbol{x}) = \sum_{i=1}^{k} \frac{c_i^2}{x_i}$ where $\boldsymbol{x} > 0$. At first, we will show that $f(\boldsymbol{x})$ is convex. Clearly,

$$\nabla f(\boldsymbol{x}) = -\begin{bmatrix} \frac{c_1^2}{x_i^2} \\ \vdots \\ \frac{c_k^2}{x_k^2} \end{bmatrix}, \nabla^2 f(\boldsymbol{x}) = 2\begin{bmatrix} \frac{c_1^2}{x_1^3} & & \\ & \ddots & \\ & & \frac{c_k^2}{x_k^3} \end{bmatrix} \in \mathbb{S}_{++}^k.$$
(38)

The convex property indicates that the objective is equivalent to prove $(\sum_{i=1}^{k} c_i)^2$ is the infimum of f(x) for $x \in$ $\{x | x^T \mathbf{1} = 1, x > 0\}$. Now, we solve

$$\min_{\boldsymbol{x}^T \mathbf{1} = 1, \boldsymbol{x} \ge 0} \sum_{i=1}^k \frac{c_i^2}{x_i},\tag{39}$$

via Lagrangian method. Let λ and $\eta \geq 0$ be Lagrangian multipliers,

$$\mathcal{L} = \sum_{i=1}^{k} \frac{c_i^2}{x_i} + \lambda (\sum_{i=1}^{k} x_i - 1) - \boldsymbol{\eta}^T \boldsymbol{x}.$$
 (40)

Therefore, the KKT conditions can be formulated as

$$\begin{cases} -\frac{c_i^2}{x_i^2} + \lambda - \eta_i = 0, \\ \sum_{i=1}^k x_i = 1, \\ \eta_i x_i = 0, \end{cases} \Rightarrow \begin{cases} \eta_i = 0, \\ \lambda = (\sum_{i=1}^k c_i)^2, \\ x_i = \frac{c_i}{k}. \\ \sum_{i=1}^k c_i \end{cases}$$
(41)

Substitute the solution into f(x) and we get $f_*(x) =$ $(\sum_{i=1}^{k} c_i)^2$. Hence, the lemma is proved.

Proof of Theorem 1. To keep simplicity, let

$$\begin{cases} \mathcal{J}_0 = \| (W^T U^T) \odot P - M \|_F^2 + \gamma \| W \|_{2,1}^2, \\ \mathcal{J}_1 = \| X \odot P - M \|_F^2 + \gamma \operatorname{tr}(X D X^T). \end{cases}$$
(42)

On the one hand, for any W and orthogonal matrix U, we can construct $X = W^T U^T$, $D = U \Lambda U^T$, and $\Lambda =$ $\operatorname{diag}(\frac{\|\boldsymbol{w}^i\|_2}{\|W\|_{2,1}})^{\dagger}.$ Then,

$$\operatorname{tr}(XDX^{T}) = \operatorname{tr}(XU\Lambda U^{T}X^{T})$$
$$=\operatorname{tr}(W^{T}\Lambda W) = \operatorname{tr}(\sum_{i} \Lambda_{ii} \boldsymbol{w}^{iT} \boldsymbol{w}^{i})$$
$$=\sum_{i} \operatorname{tr}(\Lambda_{ii} \| \boldsymbol{w}^{i} \|_{2}^{2}) = \| W \|_{2,1}^{2}.$$
(43)

Hence, we have $\min \mathcal{J}_1 \leq \min \mathcal{J}_0$. On the other hand, for any X and D, we can perform full rank factorization to X and eigenvalue factorization to D, which means X = $W^T U^T$ and $D = U \Lambda U^T$. Similarly,

$$\operatorname{tr}(XDX^T) = \sum_{i} \operatorname{tr}(\Lambda_{ii} \| \boldsymbol{w}^i \|_2^2).$$
(44)

According to Lemma 3, we have $tr(XDX^T)$ > $(\sum_i \|\boldsymbol{w}^i\|_2^o)^2 = \|W\|_{2,1}^2$. The equality holds if and only if $\Lambda_{ii} = \left(\frac{\|\boldsymbol{w}^i\|_2}{\|\boldsymbol{W}\|_{2,1}}\right)^{\dagger}. \text{ Hence, } \min \mathcal{J}_1 \geq \min \mathcal{J}_0.$ In sum, the theorem is proved.

A.3 Proof of Proposition 1

Proof. We can consider the special case that $X = x \in \mathbb{R}$, $D = y \in \mathbb{R}$, and P = M = 0. Accordingly,

$$\nabla \operatorname{tr}(XDX^{T}) = \begin{bmatrix} 2xy \\ x^{2} \end{bmatrix}$$

$$\Rightarrow \nabla^{2}\operatorname{tr}(XDX^{T}) = \begin{bmatrix} 2y & 2x \\ 2x & 0 \end{bmatrix} = H.$$
(45)

When x = y = 1 and v = [1; -1], we have $v^T H v = -1 < 0$. Hence, the problem is non-convex. Note that the constraint $tr(D^{\dagger}) = 1$ is not convex such that the problem regarding D is non-convex. When D is fixed, the subproblem,

$$\min_{X} \operatorname{tr}(XDX^{T}), \quad s.t. \ X \odot P = M,$$
(46)

is convex. To show it, we rewrite $tr(XDX^T)$ as

$$\operatorname{tr}(XDX^T) = \sum_{i} \boldsymbol{x}^i D(\boldsymbol{x}^i)^T.$$
(47)

Expand *X* according to rows, and we have

$$\nabla \operatorname{tr}(XDX^{T}) = \begin{bmatrix} D\\ D\\ \vdots\\ D \end{bmatrix}$$

$$\Rightarrow \nabla^{2} \operatorname{tr}(XDX^{T}) = \begin{bmatrix} D & 0 & \cdots & 0\\ 0 & D & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & D \end{bmatrix} \in \mathbb{S}^{mn}_{+}.$$
(48)

Hence, the proposition is proved.

A.4 Proof of Theorem 3

Proof. According to the Cauchy-Schwarz inequality, we have

$$\operatorname{tr}(XDX^{T}) = \operatorname{tr}(D^{\frac{1}{2}}(X^{T}X)^{\frac{1}{2}}(D^{\frac{1}{2}}(X^{T}X)^{\frac{1}{2}})^{T}) \cdot \operatorname{tr}((D^{\dagger})^{\frac{1}{2}}(D^{\dagger})^{\frac{1}{2}})$$
(49)

$$\geq \operatorname{tr}(D^{\frac{1}{2}}(X^{T}X)^{\frac{1}{2}}(D^{\dagger})^{\frac{1}{2}})^{2}.$$

The equality holds if and only if

$$D^{\frac{1}{2}}(X^T X)^{\frac{1}{2}} = k(D^{\dagger})^{\frac{1}{2}}.$$
(50)

Since $X = W^T U^T$ and $D = U \Lambda U^T$, we can rewrite the right hand as

$$\operatorname{tr}(XDX^{T}) \geq \operatorname{tr}(U\Lambda^{\frac{1}{2}}U^{T}(X^{T}X)^{\frac{1}{2}}U(\Lambda^{\frac{1}{2}})^{\frac{1}{2}}U^{T})^{2}$$

=
$$\operatorname{tr}(U\hat{I}U^{T}(X^{T}X)^{\frac{1}{2}})^{2}$$

=
$$(\sum_{i} \mathbb{1}[\frac{\|\boldsymbol{w}^{i}\|_{2}}{\|W\|_{2,1}} > 0]\operatorname{tr}(\boldsymbol{u}_{i}\boldsymbol{u}_{i}^{T}(X^{T}X)^{\frac{1}{2}}))^{2}.$$
(51)

where $\hat{I} = \Lambda \Lambda^{\dagger}$. Note that $tr(\boldsymbol{u}_i \boldsymbol{u}_i^T (X^T X)^{\frac{1}{2}})$ can be rewritten as

$$\operatorname{tr}(\boldsymbol{u}_{i}\boldsymbol{u}_{i}^{T}(X^{T}X)^{\frac{1}{2}}) = \operatorname{tr}(\boldsymbol{u}_{i}\boldsymbol{u}_{i}^{T}(UWW^{T}U^{T})^{\frac{1}{2}})$$
$$= \operatorname{tr}(\boldsymbol{u}_{i}^{T}U(WW^{T})^{\frac{1}{2}}U^{T}\boldsymbol{u}_{i}) \qquad (52)$$
$$= \operatorname{tr}(\boldsymbol{e}_{i}^{T}VS^{\frac{1}{2}}V^{T}\boldsymbol{e}_{i}),$$

where $\boldsymbol{e}_i = U^T \boldsymbol{u}_i$ and $WW^T = VSV^T$. Note that $\boldsymbol{e}_i^T VSV^T \boldsymbol{e}_i = 0$, $\boldsymbol{e}_i^T VS^{\frac{1}{2}}V^T \boldsymbol{e}_i = 0$. Hence, we have

$$\operatorname{tr}(XDX^{T}) \geq (\sum_{i} \mathbb{1}[\frac{\|\boldsymbol{w}^{i}\|_{2}}{\|W\|_{2,1}} > 0]\operatorname{tr}(\boldsymbol{u}_{i}\boldsymbol{u}_{i}^{T}(X^{T}X)^{\frac{1}{2}}))^{2}$$
$$= \operatorname{tr}((X^{T}X)^{\frac{1}{2}})^{2} = \|X\|_{*}^{2},$$
(53)

where $\mathbb{1}[\cdot]$ is the indicator function. Formally, $\mathbb{1}[\cdot] = 1$ if \cdot is true; otherwise, $\mathbb{1}[\cdot] = 0$.

A.5 Proof of Lemma 1

Proof. Given a matrix $Q \in \mathbb{S}^n_{++}$ and arbitrary vector $\boldsymbol{x} \in \mathbb{R}^n$, we have

$$\boldsymbol{x}^T Q \boldsymbol{x} > 0. \tag{54}$$

For any binary vector \boldsymbol{p} , suppose that $\|\boldsymbol{p}\|_0 = k$. Accordingly, the sub-matrix $[Q]_{\boldsymbol{p},\boldsymbol{p}}$ is a $k \times k$ matrix. Given an arbitrary vector $\boldsymbol{y} \in \mathbb{R}^k$, we can construct a *n*-dimension

vector, v, such that $v_{[p]} = y$ and $[v]_{\bar{p}} = 0$. Accordingly, we have that

$$0 < \boldsymbol{v}^T Q \boldsymbol{v} = \sum_{i,j=1}^n v_i v_j Q_{ij} = \boldsymbol{y}^T [Q]_{\boldsymbol{p},\boldsymbol{p}} \boldsymbol{y}.$$
 (55)

Hence, the theorem is proved.

A.6 Proof of Theorem 4

To prove Theorem 4 completely, we will prove it by two sub-theorems. First, if $D_{ii} \neq 0$ for any *i*, we aim to prove the following theorem.

Theorem 7. Let \hat{X} and \hat{V} denote the approximate solutions defined as

$$\begin{cases} \hat{\gamma}_{i} = -2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}_{i}+}, \\ \hat{\boldsymbol{x}}^{i} = \boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}_{i}+}\hat{D}^{-1}. \end{cases}$$
(56)

where $F_i = P_i \hat{D}^{-1} P_i$, $H_i = [F_i]_{\boldsymbol{p}_i, \boldsymbol{p}_i} \in \mathbb{R}^{r_i \times r_i}$ and $r_i = \|\boldsymbol{p}_i\|_0$. If $\forall i, D_{ii} \neq 0$, then there exists a constant u, which is independent on δ , such that $\hat{X} \odot P = M$ and $\|\nabla_X \mathcal{L}(\hat{X}, \hat{V})\| \leq 2\delta u \|M\|$.

The above theorem proves Theorem 4 partially. Then, we will prove the more general case when there exists i which satisfies $D_{ii} = 0$, which completes the proof.

A.6.1 Proof of Theorem 7

Lemma 4. Given a binary vector $\mathbf{p} \in \mathbb{R}^n$ and a square matrix $S \in \mathbb{R}^{n \times n}$, suppose that $[S]_{\mathbf{p},\mathbf{p}}$ is invertible. Then we have $PSPS^{\mathbf{p}+} = P$ where $P = \text{diag}(\mathbf{p})$.

Proof. Let A = PSP and $B = S^{p+}$. Then we have

$$(AB)_{ij} = \sum_{k} a_{ik} b_{ik}.$$
(57)

It is not hard to see that

$$AB = P. (58)$$

Lemma 5. For any $S \in \mathbb{S}_{++}^n$ and $i \in \{1, 2, \cdots, n\}$, $S_{ii} \neq 0$.

Proof. Use the eigenvalue decomposition, and we can factor S as

$$S = U^{T} \Lambda U. \tag{59}$$

Then, we have

$$S_{ii} = \boldsymbol{u}_i^T \Lambda \boldsymbol{u}_i > 0.$$
 (60)

Lemma 6. Given a matrix $A \in \mathbb{R}^{m \times n}$ and $x \in \mathbb{R}^n$, the following inequality,

$$\|A\boldsymbol{x}\|_{2} \leq \|A\|_{F} \|\boldsymbol{x}\|_{2}.$$
(61)

Lemma 7. [45] Let $A \in \mathbb{R}^{n \times n}$ be an invertible matrix, and $B = A^{-1}$. If $b_{qp} \neq 0$ for any $p, q \in \{1, 2, \dots, n\}$, then $A_{\bar{p}, \bar{q}} \in \mathbb{R}^{(n-1) \times (n-1)}$ is invertible. Let $M = A_{\bar{p}, \bar{q}}^{-1}$, and then we have

$$m_{ij} = b_{ij} - \frac{b_{ip}b_{qj}}{b_{qp}}.$$
(62)

The lemma is proved in literature [45]. To keep the notations uncluttered, we use $\|\cdot\|$ to replace $\|\cdot\|_F$ for short.

Lemma 8. For $\forall S \in \mathbb{S}_{++}^n$ and binary vector \boldsymbol{p} which satisfies $\|\boldsymbol{p}\|_0 = n - 1$, $\|S^{\boldsymbol{p}+}S\|^2 < (n-1) + \frac{\|\boldsymbol{b}_k\|_2^2}{b_{kk}^2}$ where $p_k = 0$ and $B = S^{-1}$.

Proof. Without loss of generality, we assume that k = n, *i.e.*, $p_n = 0$. To be simple, let $M = S^{p+}$. According to Lemma 7, we have

$$m_{ij} = \begin{cases} b_{ij} - \frac{b_{in}b_{nj}}{b_{nn}}, & i, j \le n-1; \\ 0, & \text{otherwise.} \end{cases}$$
(63)

Let $T = S^{p+}S$. Without formal proof, $t^n = 0$ and $[T]_{p,p} = I$. Now, we focus on t_n using the above equation. For $i \neq n$,

$$t_{in} = \sum_{l} m_{il} a_{ln} = \sum_{l} (b_{il} - \frac{b_{in} b_{nl}}{b_{nn}}) a_{ln}$$

= $\sum_{l} b_{il} a_{ln} - \frac{b_{in}}{b_{nn}} \sum_{l} b_{nl} a_{ln}.$ (64)

Due to AB = I, we have

$$(AB)_{in} = \sum_{l} b_{il} a_{ln} = \begin{cases} 1, & i = n; \\ 0, & i \neq n. \end{cases}$$
(65)

Accordingly,

$$t_{in} = -\frac{b_{in}}{b_{nn}}.$$
(66)

Hence, we have

$$\|S^{\mathbf{p}+}S\|^2 = \|T\|^2 = (n-1) + \sum_{i \neq n} \frac{b_{in}^2}{b_{nn}^2} < (n-1) + \frac{\|\mathbf{b}_n\|_2^2}{b_{nn}^2}.$$
(67)

Hence, the lemma is proved.

Lemma 9. If $\forall i, D_{ii} \neq 0$, then

$$\frac{\|\hat{\boldsymbol{d}}_{i}\|_{2}^{2}}{\hat{d}_{ii}^{2}} < \frac{\|\boldsymbol{d}_{i}\|_{2}^{2}}{d_{ii}^{2}}.$$
(68)

Since the lemma is obvious, we omit the corresponding proof.

Now, we begin the proof of Theorem 7.

Proof. First, we show that $\hat{X} \odot P = M$. Note that we only need to prove that

$$\hat{\boldsymbol{x}}^i \boldsymbol{P}_i = \boldsymbol{m}^i. \tag{69}$$

Note that

$$(F_i)^{p^i} + P_i = (F_i)^{p^i} + .$$
 (70)

Combine the above equation and Eq. (56), and we have

$$x^{i}P_{i} = m^{i}(F_{i})^{p^{i}} \hat{D}^{-1}P_{i} = m^{i}(F_{i})^{p^{i}} P_{i}\hat{D}^{-1}P_{i}$$

= $m^{i}(F_{i})^{p^{i}} F_{i} = m^{i}P_{i} = m^{i}.$ (71)

where we use the lemma and fact $\hat{X} \odot P = M$. Hence, we prove that $\hat{X} \odot P = M$ holds.

Now, we focus on how to prove $\|\nabla_X \mathcal{L}(\hat{X}, \hat{V})\| \leq 2\delta u \|M\|$. Let $G = \nabla_X \mathcal{L}(\hat{X}, \hat{V})$ represent the gradient. Substitute Eq. (56) into G, and we have

$$\begin{aligned}
\boldsymbol{g}^{i} &= 2\boldsymbol{x}^{i}\boldsymbol{D} + \boldsymbol{v}^{i}\boldsymbol{P}_{i} = 2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \hat{\boldsymbol{D}}^{-1}\boldsymbol{D} - 2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \boldsymbol{P} \\
&= 2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{D}} - 2\delta\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \hat{\boldsymbol{D}}^{-1} - 2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \boldsymbol{P} \\
&= 2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} - 2\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} - 2\delta\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \hat{\boldsymbol{D}}^{-1} \\
&= -2\delta\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \hat{\boldsymbol{D}}^{-1} \\
&= -2\delta\boldsymbol{m}^{i}(P_{i}\hat{\boldsymbol{D}}^{-1}P_{i})^{\boldsymbol{p}^{i}} + \hat{\boldsymbol{D}}^{-1}.
\end{aligned}$$
(72)

According to Lemma 6, we have

$$\|\boldsymbol{g}^{i}\| = 2\delta \|\boldsymbol{m}^{i}(F_{i})^{\boldsymbol{p}^{i}} + \hat{D}^{-1}\| \le 2\delta \|\boldsymbol{m}^{i}\| \|(F_{i})^{\boldsymbol{p}^{i}} + \hat{D}^{-1}\|.$$
(73)

Now, we need to prove that there exists a constant u such that $||(F_i)^{p^i} + \hat{D}^{-1}|| \le u$. Let $S = \hat{D}^{-1}$, and we have

$$\begin{aligned} &\|(F_i)^{p+}\hat{D}^{-1}\| \\ &= \|(S)^{p_+}S\| \\ &= \|(S)^{p_1+}P_1P_2\cdots S\| \\ &= \|(S)^{p_1+}P_1SP_1(S)^{p_1+}P_2SP_2(S)^{p_2+}\cdots S\| \\ &\leq \|(S)^{p_1+}P_1SP_1\| \cdot \|(S)^{p_1+}P_2SP_2\| \cdots \|(S)^{p_t+}S\|. \end{aligned}$$
(74)

where $[p_1, p_2, \dots, p_t]$ is a sequence, $p_1 = p$, and $||p_t||_0 = n - 1$. p_{i+1} is constructed by replace a zero entry of p_i . According to Lemma 8, we have

$$||(S)^{p_t+}S||^2 < (n-1) + \frac{||\boldsymbol{d}_k||_2^2}{\hat{d}_{kk}^2}.$$
(75)

where $(\mathbf{p}_t)_k = 0$. According to the Lemma 9, there exists a constant c_t such that

$$\frac{\|\hat{d}_{k}\|_{2}^{2}}{\hat{d}_{kk}^{2}} \le c_{t}.$$
(76)

Hence,

$$||(S)^{\mathbf{p}_t} + S||^2 < (n-1) + nc_t^2.$$
(77)

Similarly, it is not hard to find that

$$\|(S)^{\boldsymbol{p}_{i}} + P_{i+1}SP_{i+1}\|^{2} < (n-1) + nc_{i}^{2}.$$
 (78)

$$u = \sqrt{\prod_{i} [(n-1) + nc_i^2]}.$$
 (79)

Hence, the theorem is proved.

A.6.2 Proof of the General Case

For any matrix A, let A_{-i} denote a sub-matrix which is obtained by deleting the *i*-th column. Besides, define π_i as a vector where its *i*-th entry is 0 and others are 1.

The following lemma explains the situation when $D_{ii} = 0$.

Lemma 10. As D is computed by Algorithm 1, $D_{ii} = 0$ if and only if $x_i = 0$.

Proof. On the one hand, if $\boldsymbol{x}_i = 0$, then $(X^T X)_{ii} = 0$. Let $X^T X = V^T \Lambda V$. Note that

$$(X^T X)_{ii} = \boldsymbol{v}_i^T \Lambda \boldsymbol{v}_i = 0.$$
(80)

Clearly, $(X^T X)^{\frac{1}{2}} = V^T \Lambda^{\frac{1}{2}} V$. Hence, we have

$$((X^T X)^{\frac{1}{2}})_{ii} = \boldsymbol{v}_i^T \Lambda^{\frac{1}{2}} \boldsymbol{v}_i = 0.$$
 (81)

Similarly,

$$((X^T X)^{\frac{1}{2}})_{ii} = \boldsymbol{v}_i^T \Lambda^{\frac{1}{2}} \boldsymbol{v}_i.$$
(82)

On the other hand, the proof of $D_{ii} = 0 \Rightarrow x_i = 0$ is similar.

Corollary 3. As D is computed by Algorithm 1, if $D_{ii} = 0$, then $\|\boldsymbol{\gamma}_i\|_0 = \|\boldsymbol{\gamma}^i\|_0 = 0$.

Lemma 11. For any *i*, define $\pi_i \in \mathbb{R}^{n+1}$. Given an arbitrary matrix $Q \in \mathbb{S}^n_+$, let A be an $(n + 1) \times (n + 1)$ matrix where $[A]_{\pi_i,\pi_i} = Q$ and $[A]_{\pi_i,\pi_i} = 0$. Then we will have $A \ge 0$, $[A^{\frac{1}{2}}]_{\pi_i,\pi_i} = Q^{\frac{1}{2}}$ and $[A^{\frac{1}{2}}]_{\pi_i,\pi_i} = 0$.

The proof is similar with Lemma 4.

Lemma 12. Let $X_* = \arg \min_{X \odot P = M} \|X\|_*$. If $\forall i, (i, j) \notin \Omega$, then $\|(x_j)_*\|_0 = 0$.

Proof. Let

$$R = \arg \min_{X_{-i} \odot P_{-i} = M_{-i}} \|X\|_*.$$
 (83)

If $(X_{-i})_* \neq R$, then we have $||R||_* < ||(X_{-i})_*||_*$. Thereby we can construct a matrix X_0 that satisfies

$$[X_0]_{-i} = R, [X_0]_{i,\cdot} = 0.$$
(84)

According to Lemma 11, we have

$$||X_0||_* = \operatorname{tr}[(X_0^T X_0)^{\frac{1}{2}}] = \operatorname{tr}[(R^T R)^{\frac{1}{2}}] = ||R||_* > ||(X_{-i})_*||_*,$$
(85)

which results in a conflict.

Hence,
$$(X_{-i})_* = R$$
. In other words, $(\boldsymbol{x}_i)_* = 0$.

Proof of Theorem 4. As is shown in Algorithm 1 and Lemma 10, if $\forall i, (i, j) \notin \Omega$, then x_i , computed by

$$x^{i} = \hat{m}^{i}(\hat{F}_{i})^{p_{i}+}(H_{i}\hat{D}H_{i})^{p_{i}+},$$
 (86)

will always be 0. According to Lemma 12, the neglect of the *i*-th column is sound due to there being no observed entry.

To complete the proof of Theorem 4, we can convert the situation, where $\exists i, D_{ii} = 0$, into the simple case which has been discussed in the last subsection. Specifically speaking, provided that $\forall k, (k, i) \notin \Omega$, the original problem can be transformed into

$$\min_{X_{-i},[D]_{\boldsymbol{\pi}_{i},\boldsymbol{\pi}_{i}}} \operatorname{tr}(X_{-i}[D]_{\boldsymbol{\pi}_{i},\boldsymbol{\pi}_{i}}X_{-i}^{T}),$$
s.t. $X_{-i} \odot P_{-i} = M_{-i}, [D]_{\boldsymbol{\pi}_{i},\boldsymbol{\pi}_{i}} \ge 0, \operatorname{tr}([D]_{\boldsymbol{\pi}_{i},\boldsymbol{\pi}_{i}}^{\dagger}) = 1.$
(87)

The above transformation can be performed multiple times until all diagonal entries of D are non-zero. Let X_- , D_- , and Γ_- (Lagrangian multipliers) be the corresponding matrices when all unobserved columns are removed. Clearly, we have

$$\|\nabla_X \mathcal{L}(X, \Gamma)\| = \|\nabla_{X_-} \mathcal{L}(X_-, \Gamma_-)\|.$$
(88)

Combining with the conclusion of Theorem 7, the theorem is thus proved. $\hfill \Box$

A.7 Proof of Corollary 1

The proof of Corollary 1 relies on Theorem 5 while the latter does not rely on the former. Therefore, we just employ the conclusion of Theorem 5.

Proof. The proof is similar to the one for Theorem 5. If each step decreases the objective value, the algorithm has to approach a local minimum since the loss is lower-bounded.

Clearly, if $\delta \rightarrow 0$, then the solution that Algorithm 1 tends toward will be a valid solution for problem (90). For this convex problem, the solution is indeed its optimum. As is shown in the proof of Theorem 5, it is also the global optimum for the proposed surrogate.

Hence, the theorem is proved. \Box

A.8 Proof of Theorem 5

Proof. According to Proposition 1, problem (11) in the main paper is a non-convex optimization problem. From Theorem 2, we find that if the model converges, the optimal solutions, X_* and D_* , should satisfy

$$\mathcal{R}((X_*)^T) \subseteq \mathcal{R}((D_*)^{\dagger}), \tag{89}$$

where $\mathcal{R}(\cdot)$ represents the space spanned by columns. Let $G = D^{\dagger}$. Interestingly, the above condition indicates that problem (11) has the same optimum with a *sub-problem*,

$$\min_{X,G} \operatorname{tr}(XG^{\dagger}X^{T}),$$

s.t. $X \odot P = M, \operatorname{tr}(G) = 1, G \in \mathbb{S}^{n}_{+}, \mathcal{R}(X^{T}) \subseteq \mathcal{R}(G).$
(90)

It should be emphasized that the feasible domain of the above problem is smaller than problem (11) such that we call it a *sub-problem*.

According to Page 76 and 651 of [46], the sub-problem is convex. As we analyse above, X_* and D_* are also valid solution for the sub-problem since $\mathcal{R}((X_*)^T) \subseteq \mathcal{R}((D_*)^{\dagger})$.

Via reductio, there is no solution which leads to a smaller value of the sub-problem. Combining with the convexity of the sub-problem, (X_*, D_*) is the optimum of it. Suppose that there exists (X_*, D_0) that satisfies the constraints of problem (11) but does not obey $\mathcal{R}((X_*)^T) \subseteq \mathcal{R}(D_0^{\dagger})$. Then, we can set

$$\hat{D}_0 = \left(\frac{((X_*)^T X_*)^{\frac{1}{2}}}{\operatorname{tr}(((X_*)^T X_*)^{\frac{1}{2}}))^{\dagger}},\tag{91}$$

such that $\operatorname{tr}(X_*\hat{D}_0(X_*)^T) \leq \operatorname{tr}(X_*D_0(X_*)^T)$. Since $\operatorname{tr}(X_*D_0(X_*)^T) \leq \operatorname{tr}(X_*\hat{D}_0(X_*)^T)$, $\operatorname{tr}(X_*D_0(X_*)^T) = \operatorname{tr}(X_*\hat{D}_0(X_*)^T)$. In other words, (X_*, \hat{D}_0) is also an optimal solution of problem (11).

Furthermore, the optimal solution of the sub-problem should be the optimal solution of problem (11).

Now, we need to show the following equation,

$$||X_*||_*^2 = \min_{X \odot P = M} ||X||_*^2.$$
(92)

Suppose that $\tilde{X} = \arg \min_{X \odot P = M} ||X||_*^2$. Since $X_* \odot P = M$, $||X_*||_*^2 \ge ||\tilde{X}||_*^2$. Similarly, if we set

$$\tilde{D} = \left(\frac{(\tilde{X}^T \tilde{X})^{\frac{1}{2}}}{\operatorname{tr}((\tilde{X}^T \tilde{X})^{\frac{1}{2}})}\right)^{\dagger},\tag{93}$$

then (\tilde{X}, \tilde{D}) is a solution of the sub-problem. According to Theorem 3, we have $||X_*||_*^2 \leq ||\tilde{X}||_*^2$. Therefore, we have proven $\|\tilde{X}\|_{*}^{2} = \|X_{*}\|_{*}^{2}$, which means $X_{*} = \tilde{X}$.

Hence, the theorem is proved.

A.9 Proof of Theorem 6

Proof. Let $l_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2$, and the problem to solve is

$$\min_{S1=1,S\geq 0} \operatorname{tr}(XLX^{T}) + \|\boldsymbol{\mu}^{T}S\|_{F}^{2}$$

$$\Leftrightarrow \min_{S1=1,S\geq 0} \sum_{i}^{n} \sum_{j}^{n} l_{ij}S_{ij} + \mu_{i}^{2} \sum_{j}^{n} S_{ij}^{2}.$$
(94)

And the problem is equivalent to solve the following nsubproblems individually,

$$\min_{\mathbf{s}^{i}\mathbf{1}=1,\mathbf{s}^{i}\geq 0} \sum_{j}^{n} l_{ij} S_{ij} + \mu_{i}^{2} \sum_{j} S_{ij}^{2}.$$
 (95)

More abstractly, every subproblem is equivalent to

$$\min_{\boldsymbol{\alpha}^T \mathbf{1}_n = 1, \boldsymbol{\alpha} \ge 0} \| \boldsymbol{\alpha} + \frac{f}{2\lambda} \|_2^2.$$
(96)

Similarly, the Lagrangian of the above equation is

$$\mathcal{L}_{\alpha} = \|\boldsymbol{\alpha} + \frac{\boldsymbol{f}}{2\lambda}\|_{2}^{2} + \xi(1 - \sum_{i=1}^{n} \alpha_{i}) - \sum_{i=1}^{n} \beta_{i}\alpha_{i}, \qquad (97)$$

where ξ and β_i is Lagrangian variables. The KKT conditions are given as

$$\begin{cases} \frac{\partial \mathcal{L}_{\alpha}}{\partial \alpha_{i}} = \alpha_{i} + \frac{f_{i}}{2\lambda} - \xi - \beta_{i} = 0\\ \beta_{i}\alpha_{i} = 0\\ \sum_{i=1}^{n} \alpha_{i} = 1, \beta_{i} \ge 0, \alpha_{i} \ge 0. \end{cases}$$
(98)

Then we consider the following cases

$$\begin{cases} \alpha_i = 0 \Rightarrow \xi - \frac{f_i}{2\lambda} = -\beta_i \le 0\\ \alpha_i \ge 0 \Rightarrow \alpha_i = \xi - \frac{f_i}{2\lambda}, \end{cases}$$
(99)

which means

$$\alpha_i = (\xi - \frac{f_i}{2\lambda})_+. \tag{100}$$

Without loss of generality, assume that $f_1 \leq f_2 \leq \cdots \leq f_n$. If $\xi - \frac{f_{k+1}}{2\lambda} \le 0 < \xi - \frac{f_k}{2\lambda}$, then α is *k*-sparse. Due to $\sum_{i=1}^n \alpha_i =$ 1, we have

$$\sum_{i=1}^{n} \alpha_i = k\xi - \sum_{i=1}^{k} \frac{f_i}{2\lambda} = 1,$$
(101)

which means

$$\xi = \frac{1}{2k\lambda} \sum_{i=1}^{k} f_i + \frac{1}{k}.$$
 (102)

Combine with our assumption and we have

$$\frac{1}{2k\lambda} \sum_{i=1}^{k} f_i + \frac{1}{k} - \frac{f_{k+1}}{2\lambda} \le 0 < \frac{1}{2k\lambda} \sum_{i=1}^{k} f_i + \frac{1}{k} - \frac{f_k}{2\lambda} \Rightarrow \frac{f_k}{2\lambda} < \frac{1}{2k\lambda} \sum_{i=1}^{k} f_i + \frac{1}{k} \le \frac{f_{k+1}}{2\lambda} \Rightarrow \frac{kf_k}{2} - \frac{1}{2} \sum_{i=1}^{k} f_i < \lambda \le \frac{kf_{k+1}}{2} - \frac{1}{2} \sum_{i=1}^{k} f_i.$$
(103)

Hence, if λ is set within the above range, then α will be k-sparse. In other words, λ is converted into the amount of neighbors k. In classification tasks, as the amount of kfrequently takes a small proportion, k is set as a large value and named as the number of activation samples.

If we simply set λ as its upper bound, *i.e.*, $\lambda = \frac{kf_{k+1}}{2}$ – $\frac{1}{2}\sum_{i=1}^{k} f_i$, we have

$$\alpha_{i} = \left(\frac{\sum_{i=1}^{k} f_{i} + 2\lambda}{2k\lambda} - \frac{f_{i}}{2\lambda}\right)_{+} = \left(\frac{f_{k+1} - f_{i}}{kf_{k+1} - \sum_{i=1}^{k} f_{i}}\right)_{+}.$$
 (104)

Hence, the theorem is proved.

REFERENCES

- [1] M. Aharon, M. Elad, and A. Bruckstein, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation,' IEEE Transactions on Signal Processing, vol. 54, no. 11, pp. 4311-4322, 2006.
- [2] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," Pacific Journal of optimization, vol. 6, no. 615-640, p. 15, 2010.
- [3] S. Sodagari, A. Khawar, T. C. Clancy, and R. McGwier, "A projection based approach for radar and telecommunication systems coexistence," in 2012 IEEE Global Communications Conference (GLOBECOM), 2012, pp. 5010-5014.
- [4] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," Foundations of Computational mathematics, vol. 9, no. 6, p. 717, 2009.
- [5] E. J. Candès and T. Tao, "The power of convex relaxation: Nearoptimal matrix completion," IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2053–2080, 2010.
- [6] D. Zhang, Y. Hu, J. Ye, X. Li, and X. He, "Matrix completion by truncated nuclear norm regularization," in 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 2192-2199.
- [7] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2000, New Orleans, LA, USA, July 23-28, 2000. ACM, 2000, pp. 417-424.
- [8] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," Big Data Mining and Analytics, vol. 1, no. 4, pp. 308-323, 2018.
- E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, [9]
- pp. 1–37, 2011. [10] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted lowrank tensors via convex optimization," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5249-5257.
- [11] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, "On the applications of robust pca in image and video processing, Proceedings of the IEEE, vol. 106, no. 8, pp. 1427–1457, 2018. [12] T. K. Pong, P. Tseng, S. Ji, and J. Ye, "Trace norm regularization:
- Reformulations, algorithms, and multi-task learning," SIAM J. Optim., vol. 20, no. 6, pp. 3465–3489, 2010.
- [13] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, 2006, pp. 41-48.
- [14] R. Zhang, H. Zhang, and X. Li, "Robust multi-task learning with flexible manifold constraint," *IEEE Transactions on Pattern Analysis* and Machine Intelligence, pp. 1-1, 2020.
- [15] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization, IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 9, pp. 2117-2130, 2012.

- [16] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," International journal of computer vision, vol. 121, no. 2, pp. 183-208, 2017.
- [17] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," Advances in neural information processing systems, vol. 22, pp. 1033-1041, 2009.
- [18] Z. Xu, X. Chang, F. Xu, and H. Zhang, " $l_{1/2}$ regularization: A thresholding representation theory and a fast solver," IEEE Transactions on neural networks and learning systems, vol. 23, no. 7, pp. 1013-1027, 2012.
- [19] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in Proceedings of the IEEE international conference on computer vision, 2013, pp. 217–224.
- [20] F. Nie, Z. Hu, and X. Li, "Matrix completion based on non-convex low-rank approximation," IEEE Transactions on Image Processing, vol. 28, no. 5, pp. 2378–2388, 2018.
- [21] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," Journal of the American statistical Association, vol. 96, no. 456, pp. 1348–1360, 2001.
- [22] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in Advances in neural information processing systems, 2005, pp. 1329-1336.
- [23] N. Srebro and R. R. Salakhutdinov, "Collaborative filtering in a non-uniform world: Learning with the weighted trace norm," in Advances in Neural Information Processing Systems, 2010, pp. 2056-2064
- [24] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust 1 1-norm," in 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2012, pp. 1410-1417.
- [25] J. Fan, L. Ding, Y. Chen, and M. Udell, "Factor group-sparse regularization for efficient low-rank matrix recovery," in Advances in Neural Information Processing Systems, 2019, pp. 5105–5115.
- [26] F. Shang, J. Cheng, Y. Liu, Z.-Q. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust pca: Algorithms and applications," IEEE transactions on pattern analysis and machine intelligence, vol. 40, no. 9, pp. 2066-2080, 2017.
- [27] Z. Zhao, L. Zhang, X. He, and W. Ng, "Expert finding for question answering via graph regularized matrix completion," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 4, pp. 993-1004.2014.
- [28] K. Ji, J. Tan, J. Xu, and Y. Chi, "Learning latent features with pairwise penalties in low-rank matrix completion," IEEE Transactions on Signal Processing, vol. 68, pp. 4210–4225, 2020. [29] N. Rao, H. Yu, P. Ravikumar, and I. S. Dhillon,
- "Collaborative filtering with graph information: Consistency and scalable methods," in Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada. C. Cortes, N. D. Lawrence, D. D. Lee. M. Sugiyama, and R. Garnett, Eds., 2015, pp. 2107-2115. [Online]. Available: https://proceedings.neurips.cc/paper/2015/ hash/f4573fc71c731d5c362f0d7860945b88-Abstract.html
- [30] G. Liu and P. Li, "Low-rank matrix completion in the presence of high coherence," IEEE Transactions on Signal Processing, vol. 64, no. 21, pp. 5623-5633, 2016.
- [31] C.-G. Li and R. Vidal, "A structured sparse plus structured lowrank framework for subspace clustering and completion," IEEE Transactions on Signal Processing, vol. 64, no. 24, pp. 6557-6570, 2016.
- [32] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," Foundations and Trends® in Machine learning, vol. 3, no. 1, pp. 1-122, 2011.
- [33] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization," in Advances in neural information processing systems, 2010, pp. 1813–1821.
- [34] L. Liu, W. Huang, and D.-R. Chen, "Exact minimum rank approximation via schatten p-norm minimization," Journal of Computational and Applied Mathematics, vol. 267, pp. 218-227, 2014.
- [35] X. P. Li, L. Huang, H. C. So, and B. Zhao, "A survey on matrix completion: Perspective of signal processing," arXiv preprint arXiv:1901.10885, 2019.
- [36] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier

method for exact recovery of corrupted low-rank matrices," arXiv preprint arXiv:1009.5055, 2010.

- [37] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in Advances in neural information processing systems, 2011, pp. 612-620.
- [38] F. Nie, H. Huang, and C. Ding, "Low-rank matrix recovery via efficient schatten p-norm minimization," in Twenty-sixth AAAI conference on artificial intelligence, 2012.
- [39] N. Srebro and A. Shraibman, "Rank, trace-norm and maxnorm," in International Conference on Computational Learning Theory. Springer, 2005, pp. 545–560.
- [40] F. R. Chung and F. C. Graham, Spectral graph theory. American Mathematical Soc., 1997, no. 92.
- [41] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1422-1430.
- [42] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.
- [43] J. M. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in Proc International Conference on Computer Vision, 2005, pp. 756-763.
- [44] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," Acm transactions on interactive intelligent systems (tiis), vol. 5, no. 4, pp. 1-19, 2015.
- [45] E. Juárez-Ruiz, R. Cortés-Maldonado, and F. Pérez-Rodríguez, "Relationship between the inverses of a matrix and a submatrix," *Computación y Sistemas*, vol. 20, no. 2, pp. 251–262, 2016. [46] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*.
- Cambridge university press, 2004.

Xuelong Li (M'02-SM'07-F'12) is a Full Professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China.



Hongyuan Zhang received the B.E. degree in software engineering from Xidian University, Xi'an, China in 2019. He is currently pursuing the Ph.D. degree from the School of Computer Science and the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China.



Rui Zhang (M'19) received the Ph.D degree in computer science at Northwestern Polytechnical University, Xi'an, China in 2018. He currently serves as an Associate Professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xi'an, China.