IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 8, 2019, accepted April 28, 2019, date of publication May 8, 2019, date of current version June 3, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915636

Binary Matrix Completion With Nonconvex Regularizers

CHUNSHENG LIU AND HONG SHAN
College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China
Corresponding author: Chunsheng Liu (liuchunsheng17a@nudt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61602491.

ABSTRACT Many practical problems involve the recovery of a binary matrix from partial information, so the binary matrix completion (BMC) technique has increasingly been of interest in machine learning. In particular, we consider a special case of the BMC problems, in which only a subset of positive elements can be observed. In recent years, convex regularization-based methods are the mainstream approaches for this task. However, applications of nonconvex surrogates in standard matrix completion have demonstrated better empirical performance. Accordingly, we propose a novel BMC model with nonconvex regularizers and provide the recovery guarantee for the model. Furthermore, to solve the resultant nonconvex optimization problem, we improve the popular proximal algorithm with acceleration strategies. It can be guaranteed that the convergence rate of the algorithm is on the order of $1/T$, where T is the number of iterations. The extensive experiments conducted on both synthetic and real-world data sets demonstrate the superiority of the proposed approach over other competing methods.

INDEX TERMS Binary matrix completion, link prediction, nonconvex regularizers, topology inference.

I. INTRODUCTION

The matrix completion problem attempts to recover a low-rank or an approximate low-rank matrix by observing only partial elements [1]. In recent years, many strong theoretical analyses have been developed on the matrix completion problem [2]–[7], which has been performed and applied in a wide variety of practical applications, such as background modeling [8], [9], recommender systems [10], sensor localization [11], [12], image and video processing [13], [14], and link prediction [15]. In particular, all these results are based on a potential assumption that the observed entries are continuous-valued. However, in many practical applications, the observations are not only incomplete but also are often highly quantized to a single bit [16]. Therefore, there is a conspicuous gap between those existing approaches and practical situations, which promotes the rapid development of 1-bit matrix completion [16].

Instead of observing a subset of full entries, a more common situation in practice is to observe only the subset of positive elements. Thus, the observations are not only binary but also nonnegative. For instance, consider the link prediction problem in social networks, where only positive relationships, such as ''friendships'', can be observed, while no ''non-friendships'' are observed. The goal here is to recover the whole social network from the observed friendships (positive entries). In the context of binary classification, the problems learned from positive and unlabeled examples are called positive and unlabeled learning (PU learning for short) [17]. Consequently, the unobserved entries are regarded as unlabeled samples, and then PU learning is applied to matrix completion [18].

The existing methods of PU matrix completion [18], [19] are all based on the convex regularizers such as nuclear norm and max-norm. However, many works [9], [13], [14], [20] state that the (convex) nuclear norm might not be a good enough approximation of the rank function. In contrast, better recovery performance can be achieved by nonconvex surrogates [21]–[23]. Accordingly, we attempt to introduce the nonconvex regularizers into PU matrix completion.

In this paper, we propose a novel model of PU matrix completion with nonconvex regularizers and provide the recovery guarantee for the model. To cope with the challenges of the resulting nonconvex optimization problem, we improve the proximal algorithm with two acceleration schemes: i) Instead of full singular value decomposition (SVD), only a few

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaojie Guo.

leading singular values are needed to generate the next iteration. ii) We replace a large matrix by its projection on leading subspace, and then the reduction of matrix size makes the calculation of proximal operator more efficient. Moreover, we show that further acceleration is available by taking advantage of the sparse structure. Subsequently, the resultant algorithm, named "PU matrix completion with nonconvex regularizers (PUMC_N)," is analyzed in detail from the aspects of convergence and time complexity, respectively.

The primary contributions of our work can be summarized as follows:

- *By employing the nonconvex regularization, we propose a novel PU matrix completion model and provide a strong guarantee for matrix recovery; i.e., the error in recovering an $m \times n$ 0-1 matrix is $\mathcal{O}\left(\frac{1}{\delta^2\sqrt{mn}}\right)$, where δ denotes the sampling rate of positive entries.*
- *We develop an accelerated version of the proximal algorithm for solving the resultant nonconvex optimization model. It can be guaranteed that the proposed algorithm has a convergence rate of $\mathcal{O}\left(1/T\right)$, where T denotes the number of iterations.*
- *We implement and analyze the proposed algorithm on both synthetic and real-world data sets. The experimental results demonstrate the superiority of the resultant algorithm to state-of-the-art methods; that is, in general, for the same observation matrix, the proposed method can obtain a more accurate recovery matrix in less time.*

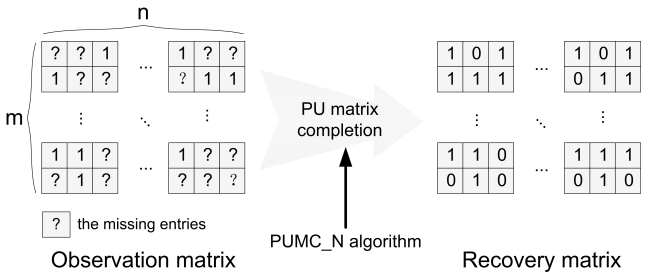


FIGURE 1. The overview of the proposed framework. The left is an observation matrix and the right is the recovery binary matrix, where the PU matrix completion model with nonconvex regularizers is proposed in Section III, and the resulting PUMC_N algorithm is introduced in Section IV.

The paper is organized as follows (Fig. 1). The following section is a brief overview of the related work. In Section III, we propose the model and provide its recovery guarantee. A fast and efficient algorithm is proposed in Section IV, followed by the convergence and time complexity analysis. Experimental results on both synthetic and real-world data sets are presented in Section V. Finally, the conclusion is summarized in Section VI.

Notation: In this paper, vectors and matrices are denoted by lowercase and uppercase boldface, respectively. For a matrix $\mathbf{X}$, $\mathbf{X}^\top$ denotes its transpose, $\mathbf{X}_k = \mathbf{X}(:, 1:k)$ is its leading k columns, $\|\mathbf{X}\|_F = \sqrt{\sum_{i,j} X_{ij}^2}$ is the Frobenius norm of $\mathbf{X}$, and $\|\mathbf{X}\|_* = \sum_i \sigma_i(\mathbf{X})$ is the nuclear norm, where $\sigma_i(\mathbf{X})$ is the i-th largest singular value of $\mathbf{X}$. For a set Ω, $|\Omega|$ is its cardinal number. In addition, we use ∇f for the gradient of a differentiable function f.

II. RELATED WORK

In the last decade, based on the remarkable results of low-rank matrix completion [1], a tremendous amount of work has focused on the problem, which resulted in a burst of progress concerning the matrix completion theory. A strong theoretical basis for matrix completion [2], [3], [5], including the case of approximate low-rank matrices and noisy observations [1], [4], [9], [11], has been established.

However, all these results are based on the underlying assumption that the observed entries are continuous-valued. In practice, many applications, such as the popular *Netflix*[1] and *MovieLens* [16] among recommender systems, have a rating matrix whose entries are discrete and quantized rather than continuous. Consequently, there is a conspicuous gap between standard matrix completion theory and practice, revealing the inadequacy of the corresponding methods in dealing with the above case.

Motivated by the above challenge, 1-bit matrix completion was advocated for the first time in [16] to deal with the binary (1-bit) observations. Theoretical guarantees were provided to show the efficiency of the method. In addition, a suite of experiments on both synthetic and real-world data sets illustrated some practical applications and demonstrated the superiority of 1-bit matrix completion. Then, [24] considered a general nonuniform sampling distribution concerning the 1-bit matrix completion problem followed by corresponding theoretical guarantees. Moreover, the noisy version was studied in [25] under the same sampling scheme as [24]. Instead of the nuclear norm, [25] used the max-norm as a convex relaxation for the rank function. Similarly, [26] addressed the problem of social trust prediction with a 1-bit max-norm constrained formulation. In addition, under constraints on infinity norm and exact rank, the noisy 1-bit matrix completion problem was explored in [27] and [28]. Furthermore, though an analysis on PAC-Bayesian bounds, [29] evaluated the performance of 1-bit matrix completion.

Relative to the settings of 1-bit matrix completion, there is a more common situation in practice. Consider the link prediction problem in social networks: instead of observing a subset of full entries, we can only observe a subset of the positive relationships, which is "one-sided" sampling in [18]. A similar situation also occurs in network topology inference problem [48]. In response to such a case, [18] proposed PU matrix completion. This method introduced the idea of PU (positive and unlabeled) learning [17], [30], i.e., learning only in the presence of positive and unlabeled examples. Motivated by the development of semi-supervised classification [31] in recent years, [19] proposed a modified version of PU matrix completion.

[1] https://netflixprize.com/index.html.

In particular, the PU matrix completion was considered under the constraints on the nuclear norm. As the tightest convex lower bound of the matrix rank function, the nuclear norm is the most popular convex regularizer. Many algorithms based on the nuclear norm, such as the accelerated inexact soft-impute algorithm (AIS-Impute) [32], singular value thresholding (SVT) [33], and inexact augmented Lagrange multipliers (IALM) [34], can solve the corresponding convex optimization problem effectively. While the nuclear norm is applied successfully and makes low-rank optimization easier, numerous attempts have recently been made to regard nonconvex regularizers as the better approximation of the matrix rank. For instance, nonconvex surrogates such as the truncated nuclear norm (TNN) [9], [35], log-sum penalty (LSP) [22], [36], and capped ℓ_1 penalty [21] have been successfully applied in many fields and exhibit better empirical performance than nuclear norm regularizers.

III. PROBLEM FORMULATION

Matrix completion is the problem of recovering the underlying target matrix given its partial information [1]. Following the "basic setting" of [18], let the target matrix $\mathbf{M} \in \{0, 1\}^{m\times n}$ be a binary matrix that consists only of ones and zeros, and $\Omega_1 = \left\{(i,j)|\, M_{ij} = 1\right\}$ denotes the index set of all positive elements in $\mathbf{M}$. Equivalently, the observation matrix is denoted, herein, by $\mathbf{A} \in \{0, 1\}^{m\times n}$, and Ω denotes the index set of observation elements. According to the "one-sided" sampling in [18], only a subset of positive entries of $\mathbf{M}$ can be observed, that is, $\Omega \subseteq \Omega_1$. We suppose that the observation process follows the uniform sampling distribution, which is a popular choice for the majority of works; i.e., Ω is sampled randomly from Ω_1. For the observation matrix $\mathbf{A}$, $A_{ij} = 1$ if $(i,j) \in \Omega$, and $A_{ij} = 0$ otherwise. Here, our goal is to recover the underlying target matrix $\mathbf{M}$ from the observation matrix $\mathbf{A}$.

According to the above description, the relationship between $\mathbf{M}$ and $\mathbf{A}$ can be expressed in the following conditional probability.

$$\begin{aligned} P\left(A_{ij} = 0 \middle| M_{ij} = 1\right) &= 1 - \delta \\ P\left(A_{ij} = 1 \middle| M_{ij} = 0\right) &= 0 \end{aligned} \tag{1}$$

where $\delta = |\Omega| \big/ |\Omega_1|$ denotes the sampling rate. We consider the problem of positive and unlabeled matrix completion (PU matrix completion) with nonconvex regularizers in the following form.

$$\min_{\mathbf{X}\in\{0,1\}^{m\times n}} F(\mathbf{X}) \equiv \ell(\mathbf{X}) + \lambda r_n(\mathbf{X}). \tag{2}$$

where λ is a regularization parameter, ℓ is a smooth loss function, and r_n is a nonconvex regularizer in Table 1. In addition, (2) has the following characteristics.

- *ℓ is differentiable with a β-Lipschitz continuous gradient; that is, it follows $\|\nabla\ell(\mathbf{X}_1) - \nabla\ell(\mathbf{X}_2)\|_F \leq \beta\|\mathbf{X}_1 - \mathbf{X}_2\|_F$, $\beta > 0$. Moreover, ℓ is bounded from below, i.e., $\inf \ell > -\infty$.*

TABLE 1. The functions r and thresholds θ for nonconvex regularizers where $\mu > 0$, $\eta = \frac{\lambda}{\rho}$. For the TNN regularizer, μ is an integer denoting the number of leading singular values that are not penalized.

	$\eta r(\sigma_i(\mathbf{X}))$	θ
TNN	$\begin{cases} 0 & , i \leq \mu \\ \eta\sigma_i(\mathbf{X}) & , i > \mu \end{cases}$	$\max(\sigma_{\mu+1}(\mathbf{X}), \eta)$
capped ℓ_1	$\begin{cases} \eta\sigma_i(\mathbf{X}) & , \sigma_i(\mathbf{X}) \leq \mu \\ \eta\mu & , \sigma_i(\mathbf{X}) > \mu \end{cases}$	$\min\left(\eta, \sqrt{2\mu\eta}\right)$
LSP	$\eta\log(\sigma_i(\mathbf{X})/\mu + 1)$	$\min(\eta/\mu, \mu)$

- *$r_n(\mathbf{X}) = \sum_{i=1}^{m} r(\sigma_i(\mathbf{X}))$ is a nonconvex and nonsmooth function, where r is a nondecreasing concave function and $r(0) = 0$.*
- *r_n can be formulated as the difference of two convex functions [47], i.e., $r_n(\mathbf{X}) = \widehat{r}_n(\mathbf{X}) - \widecheck{r}_n(\mathbf{X})$, where $\widehat{r}_n$ and $\widecheck{r}_n$ are convex. (The corresponding convex functions of the nonconvex regularizers mentioned in Section II are provided in Appendix A.)*

To accurately quantify the error in recovering the underlying binary matrix, we propose to adopt the ω-weighted square loss [37], [38] as the loss function. The ω-weighted square loss is defined as

$$\ell_\omega(x,a) = \omega I_{a=a_1}\ell(x,a_1) + (1-\omega) I_{a=a_2}\ell(x,a_2). \tag{3}$$

where $\ell(x,a) = (x-a)^2$ is the square loss, and $I_{a=a_1}$ and $I_{a=a_2}$ are indicator functions, i.e., $I_{a=a_1}$ is 1 if $a = a_1$ is true and 0 otherwise.

Consequently, (2) can be further formulated as follows.

$$\min_{\mathbf{X},\mathbf{A}\in\{0,1\}^{m\times n}} F(\mathbf{X}) \equiv \lambda r_n(\mathbf{X}) + \sum\nolimits_{i,j} \ell_\omega\left(X_{ij}, A_{ij}\right). \tag{4}$$

where $\mathbf{X}$ and $\mathbf{A}$ are the recovery matrix and observation matrix of the underlying target matrix $\mathbf{M}$, respectively, and $\ell_\omega\left(X_{ij}, A_{ij}\right) = \omega I_{A_{ij}=1}\ell\left(X_{ij}, 1\right) + (1-\omega) I_{A_{ij}=0}\ell\left(X_{ij}, 0\right)$.

Recovery Error of (4): Following the definition of the recovery error in [16], [18], here the recovery error can be formulated as

$$R(\mathbf{X}) = \frac{1}{mn}\|\mathbf{X} - \mathbf{M}\|_F^2. \tag{5}$$

where $\mathbf{M}, \mathbf{X} \in \mathbb{R}^{m\times n}$ are the underlying target matrix and its recovery matrix, respectively.

In [38], the label-dependent loss is defined as $U(x,a) = I_{x=1}I_{a=0} + I_{x=0}I_{a=1}$. Similar to (3), we define the weighted version of the label-dependent loss as

$$U_\omega(x,a) = (1-\omega) I_{x=1}I_{a=0} + \omega I_{x=0}I_{a=1}. \tag{6}$$

Therefore, the corresponding ω-weighted expected error can be written as

$$R_\omega(\mathbf{X}) = E\left[\sum\nolimits_{i,j} U_\omega\left(X_{ij}, A_{ij}\right)\right]. \tag{7}$$

According to the class-conditional random noise model in [39], (1) can be written correspondingly as

$$\begin{aligned} P\left(A_{ij} = 0 \middle| M_{ij} = 1\right) &= 1 - \delta = \rho_{+1} \\ P\left(A_{ij} = 1 \middle| M_{ij} = 0\right) &= 0 = \rho_{-1} \end{aligned} \tag{8}$$

Theorem 1: *For the choices* $\hat{\omega} = \frac{1-\rho_{+1}}{2}$ *and* $\kappa = \frac{1-\rho_{+1}}{2}$, *there exists a constant c that is dependent of* $\mathbf{X} \in \mathbb{R}^{m\times n}$ *such that, for any matrix* $\mathbf{X}$, *we have* $R_{\hat{\omega}}(\mathbf{X}) = \kappa R(\mathbf{X}) + c$.

The above theorem (Theorem 1) is a special case of Theorem 9 in [39]. At this juncture, the linear mapping between the recovery error $R(\mathbf{X})$ and the ω-weighted expected error $R_{\hat{\omega}}(\mathbf{X})$ indicates that minimizing $R(\mathbf{X})$ is equivalent to minimizing $R_{\hat{\omega}}(\mathbf{X})$ on the partial information.

Theorem 2 (Main Result 1): *Let* $\hat{\mathbf{X}} \in \mathbb{R}^{m\times n}$ *be the solution to* (4); *then, with a probability of at least* $1-\alpha$,

$$\frac{1}{mn}\left\|\hat{\mathbf{X}}-\mathbf{M}\right\|_F^2 \le \frac{C}{\delta^2}\left(\sqrt{\frac{\log\left(2/\alpha\right)}{mn}}+b\right). \tag{9}$$

where C is an absolute constant, δ *denotes the sampling rate,* $b = \frac{\sqrt{m}+\sqrt{n}+\sqrt[4]{|\Omega_1|}}{mn}$, *and* Ω_1 *is the index set of all positive elements in* $\mathbf{M}$. The proof can be found in Appendix C.

According to Theorem 2, for example, the error in recovering an $m \times m$ 0-1 matrix is $\mathcal{O}\left(\frac{1}{m\delta^2}\right)$ for the proposed method compared to $\mathcal{O}\left(\frac{1}{\sqrt{m}\delta}\right)$, where δ denotes the sampling rate of positive entries.

IV. ALGORITHM

In this section, we will show that the nonconvex model can be solved much more quickly. First, Subsection A shows the basic algorithm for nonconvex regularizers, followed by acceleration schemes in Subsection B. Subsection C summarizes the whole algorithm and Subsection D analyzes the resultant algorithm from the aspects of convergence and time complexity.

A. BASIC ALGORITHM

Let $L_\omega(\mathbf{X},\mathbf{A}) = \sum_{i,j}\ell_\omega\left(X_{ij},A_{ij}\right)$, and in line with the definition of the Frobenius norm, we have the following derivation.

$$\begin{aligned} &L_\omega(\mathbf{X},\mathbf{A}) \\ &= (1-\omega)\sum_{A_{ij}=0}\left(X_{ij}-A_{ij}\right)^2+\omega\sum_{A_{ij}=1}\left(X_{ij}-A_{ij}\right)^2 \\ &= (1-\omega)\left\|\mathbf{X}-\mathbf{A}\right\|_F^2+(2\omega-1)\left\|\mathcal{P}_\Omega(\mathbf{X}-\mathbf{A})\right\|_F^2 \end{aligned} \tag{10}$$

where $[\mathcal{P}_\Omega(\mathbf{X}-\mathbf{A})]_{ij} = (\mathbf{X}-\mathbf{A})_{ij}$ if $(i,j)\in\Omega$ and 0 otherwise. In recent years, the proximal algorithm [32] has been regarded as an efficient method for solving the optimization problem $\min_{\mathbf{X}} f_1(\mathbf{X}) + f_2(\mathbf{X})$, when f_1 and f_2 are convex. The following theorem shows that the convergence of the proximal algorithm.

Theorem 3 [40]: *Let* f_1, f_2 *be lower semicontinuous, and* f_1 *is differentiable with* β*-Lipschitz continuous gradient. If* f_1+f_2 *is coercive and strictly convex, the solution of the problem takes on uniqueness. For an arbitrary initial matrix* $\mathbf{X}$, $\forall\rho > \beta$, *the iterative sequence generated by the following statement can converge to the unique solution of the problem.*

$$\mathbf{X}_{t+1} = prox_{\frac{\lambda}{\rho}f_2}\left(\mathbf{X}_t - \tfrac{1}{\rho}\nabla f_1(\mathbf{X}_t)\right). \tag{11}$$

where $prox_{\frac{\lambda}{\rho}f_2}(\mathbf{Z}) = \arg\min_{\mathbf{X},\mathbf{Z}\in\mathbb{R}^{m\times n}}\left\{\frac{1}{2}\|\mathbf{X}-\mathbf{Z}\|_F^2+\frac{\lambda}{\rho}f_2(\mathbf{X})\right\}$ *denotes the proximal operator.*

When f_2 is the nuclear norm, the following theorem shows that the proximal operator of the nuclear norm has a closed form solution.

Theorem 4 [33]: *For an arbitrary matrix* $\mathbf{Z}\in\mathbb{R}^{m\times n}$, $\forall\tau>0$, *the proximal operator of the nuclear norm of matrix* $\mathbf{X}$ *is*

$$prox_{\tau\|\mathbf{X}\|_*}(\mathbf{Z}) = \mathbf{U}(\Sigma-\tau\mathbf{I})_+\mathbf{V}^\top. \tag{12}$$

where $\mathbf{I}$ *denotes the identity matrix, SVD* $(\mathbf{Z}) = \mathbf{U}\Sigma\mathbf{V}^\top$, *and* $[\mathbf{M}_+]_{ij} = max\left(M_{ij},0\right)$.

For solving (4), we extend the proximal operator to nonconvex problem, similar to Theorem 3, at t-iteration, it products the iterative sequence as follows.

$$\mathbf{X}_{t+1} = prox_{\frac{\lambda}{\rho}r_n}\left(\mathbf{X}_t - \tfrac{1}{\rho}\nabla L_\omega(\mathbf{X}_t,\mathbf{A})\right). \tag{13}$$

where the learning rate, herein, denoted by ρ is a fixed value, and $\nabla L_\omega(\mathbf{X},\mathbf{A})$ denotes the gradient of the ω-weighted loss function, which can be computed efficiently as

$$\tfrac{1}{2}\nabla L_\omega(\mathbf{X},\mathbf{A}) = (1-\omega)(\mathbf{X}-\mathbf{A})+(2\omega-1)\mathcal{P}_\Omega(\mathbf{X}-\mathbf{A}). \tag{14}$$

Recently, due to the successful application on convex optimization problem, the proximal algorithm has been extended to nonconvex situation [9], [13], [14], [20]. Similar to the nuclear norm (Theorem 4), the generalized singular value thresholding [20] was proposed to handle the nonconvex surrogates.

Theorem 5: Generalized singular value thresholding (GSVT) [20]. *For an arbitrary matrix* $\mathbf{Z}\in\mathbb{R}^{m\times n}$, *let* r_n *be a function that satisfies the characteristics in* (2), *then the proximal operator of* r_n *has the following closed form solution.*

$$prox_{\frac{\lambda}{\rho}r_n}(\mathbf{Z}) = \mathbf{U}diag\left(\hat{s}\right)\mathbf{V}^\top. \tag{15}$$

where $\mathbf{U}\Sigma\mathbf{V}^\top$ *is the SVD of* $\mathbf{Z}$, *and* $\hat{s} = \left\{\hat{s}_i\right\}$ *with*

$$\hat{s}_i \in \arg\min_{s_i\ge 0}\tfrac{1}{2}(s_i-\sigma_i(\mathbf{Z}))^2+\tfrac{\lambda}{\rho}r(s_i). \tag{16}$$

Similar to Theorem 4, the above theorem indicates that the closed-form solutions of the nonconvex regularizers in Table 1 do exist. In addition, we generalize the above procedure as the Basic Algorithm shown in Algorithm 1.

B. ACCELERATION

However, the basic algorithm involves a full SVD (step 3) with time complexity $\mathcal{O}\left(mn^2\right)$. Next, we will use the following two schemes to make the basic algorithm much faster.

S1: The first consideration is to use partial SVD. However, $\hat{s}_i$ in (16) actually becomes zero when the singular value $\sigma_i(\mathbf{Z})$ is not larger than a threshold θ, which is automatic thresholding in [41]. This means that only the leading few singular values, instead of all singular values, are needed to

Algorithm 1 Basic Algorithm

Input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\rho > \beta$, the sampling set Ω, and the regularization parameter λ.

initialize $\mathbf{X}_1 = 0$;
for $t = 1, 2, ..., T$ **do**
  $\mathbf{X}_{ig} = \mathbf{X}_t - \frac{1}{\rho}\nabla L_\omega(\mathbf{X}_t, \mathbf{A})$;
  $[\mathbf{U}, \Sigma, \mathbf{V}] = SVD(\mathbf{X}_{ig})$;
  for $i = 1, 2, ..., m$ **do**
    $\hat{s}_i \in \arg\min_{s_i \geq 0} \frac{1}{2}(s_i - \Sigma_{ii})^2 + \frac{\lambda}{\rho} r(s_i)$;
  end
  $\mathbf{X}_{t+1} = \mathbf{U} diag(\{\hat{s}_i\})\mathbf{V}^\top$;
end

Result: $\mathbf{X}_{T+1}$

compute the proximal operator in Theorem 5. The thresholds θ for the mentioned nonconvex regularizers are shown in Table 1.

S2: The second scheme is to reduce the size of the SVD. The following theorem shows that the proximal operator on a large matrix can be replaced by the counterpart on a smaller one. The proof can be found in Appendix D.

Theorem 6: For an arbitrary matrix $\mathbf{Z} \in \mathbb{R}^{m\times n}$, let k be the number of singular values of $\mathbf{Z}$ that are not less than θ; its rank-k SVD is $\mathbf{U}_k\Sigma_k\mathbf{V}_k^\top$, and $\mathbf{W} \in \mathbb{R}^{m\times k}$ is an orthogonal matrix. If span $(\mathbf{U}_k) \subseteq$ span $(\mathbf{W})$, then the following equation holds.

$$prox_{\frac{\lambda}{\rho}r_n}(\mathbf{Z}) = \mathbf{W} prox_{\frac{\lambda}{\rho}r_n}\left(\mathbf{W}^\top\mathbf{Z}\right). \tag{17}$$

According to Theorem 6, a large matrix is replaced by its projection on leading subspace. How does one obtain such a $\mathbf{W}$ in Theorem 6? Two approaches are available to find the $\mathbf{W}$ exactly in the same time complexity. The first method, the PROPACK package [42], is widely applied to partial SVD. And the second is the power method, which has a good approximation guarantee [43]. Compared with the former method, the latter one can benefit particularly from warm-start, taking full advantage of the iterative nature of the proximal algorithm. Hence, we use the power method to get the $\mathbf{W}$, and the details are shown in Algorithm 2.

Algorithm 2 power Method

Input: Let $\mathbf{Z} \in \mathbb{R}^{m\times n}$, $\mathbf{Y} \in \mathbb{R}^{n\times k}$, and the number of iterations H.

$\mathbf{R}_1 = \mathbf{Z}\mathbf{Y}$;
for $h = 1, 2, ..., H$ **do**
  $\mathbf{W}_h = QR(\mathbf{R}_h)$; //only turning the Q matrix
  $\mathbf{R}_{h+1} = \mathbf{Z}\left(\mathbf{Z}^\top\mathbf{W}_h\right)$;
end

Result: $\mathbf{W}_H$

Let $\mathbf{X}_{ig} = \mathbf{X}_t - \frac{1}{\rho}\nabla L_\omega(\mathbf{X}_t, \mathbf{A})$, through the implementation of the above two acceleration measures, (13) can be rewritten as

$$\mathbf{X}_{t+1} = \mathbf{W}\mathbf{U}_a diag(\hat{s})\mathbf{V}_a^\top. \tag{18}$$

where $\mathbf{U}_a\Sigma_a\mathbf{V}_a^\top$ is the rank-a SVD of $\mathbf{W}^\top\mathbf{X}_{ig}$, a is an integer denoting the number of singular values that are greater than the threshold θ, and $\hat{s}$ can be obtained from (16).

Algorithm 3 Positive and Unlabeled Matrix Completion With Nonconvex Regularizers (PUMC_N)

Input: Let $\lambda_0 > \lambda$, $\rho > \beta$, υ, $\mathbf{A} \in \mathbb{R}^{m\times n}$, and the sampling set Ω.

Initialize $\mathbf{X}_0 = \mathbf{X}_1 = 0$, $\alpha_0 = \alpha_1 = 1$, and $\mathbf{V}_0, \mathbf{V}_1 \in \mathbb{R}^{n\times 1}$ as random Gaussian matrices;
for $t = 1, 2, ..., T$ **do**
  $\mathbf{Z}_t = \mathbf{X}_t + \frac{\alpha_{t-1}-1}{\alpha_t}(\mathbf{X}_t - \mathbf{X}_{t-1})$;
  $\mathbf{Z}_{ig} = \mathbf{Z}_t - \frac{1}{\rho}\nabla L_\omega(\mathbf{Z}_t, \mathbf{A})$;
  $\mathbf{Y}_t = QR([\mathbf{V}_t, \mathbf{V}_{t-1}])$;
  $\mathbf{W} = powermethod(\mathbf{Z}_{ig}, \mathbf{Y}_t)$;
  $[\mathbf{U}, \Sigma, \mathbf{V}] = SVD(\mathbf{W}^\top\mathbf{Z}_{ig})$;
  $\lambda_t = (\lambda_{t-1} - \lambda)\upsilon^t + \lambda$;
  for $i = 1, 2, ..., k$ **do**
    $\hat{s}_i \in \arg\min_{s_i \geq 0} \frac{1}{2}(s_i - \Sigma_{ii})^2 + \frac{\lambda_t}{\rho} r(s_i)$;
  end
  $\mathbf{X}_{t+1} = \mathbf{W}\mathbf{U}_k diag(\{\hat{s}_i\})\mathbf{V}_k^\top$;
  $\mathbf{V}_{t+1} = \mathbf{V}$;
  $\alpha_{t+1} = \frac{1}{2}\left(\sqrt{4\alpha_t^2 + 1} + 1\right)$;
end

Result: $\mathbf{X}_{T+1}$

C. THE WHOLE ALGORITHM

We summarize the whole procedure for solving (4) in Algorithm 3 and name it PUMC_N (Positive and Unlabeled Matrix Completion with Nonconvex regularizers). In step 3, similar to the nmAPG algorithm [44], a linear combination of $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$ is used to accelerate the algorithm. The column spaces of the current iteration ($\mathbf{V}_t$) and previous iteration ($\mathbf{V}_{t-1}$) are used to accomplish the warm start in step 5, as in [32]. Steps 6 and 7 perform **S2**, and in step 8, a continuation strategy is introduced to speed up the algorithm further. Specifically, λ is dynamic and, as the iteration proceeds, gradually decreases from a large value. In addition, steps $9-11$ perform **S1**.

D. ALGORITHM ANALYSIS

1) CONVERGENCE ANALYSIS

First, we present a lemma that provides the basic support for the convergence analysis of the proposed algorithm. The following lemma shows that the objective function F is non-increasing as the iterations proceed.

Lemma 1 [45]: Let $\{\mathbf{X}_t\}$ be the iterative sequence produced by (13)*, for the optimization problem* (2)*, we have $F(\mathbf{X}_{t+1}) \leq F(\mathbf{X}_t) - \frac{\rho-\beta}{2}\|\mathbf{X}_{t+1} - \mathbf{X}_t\|_F^2$, where $\rho > \beta$.*

The following theorem shows that the proposed algorithm generates a bounded iterative sequence. Moreover, the proof can be found in Appendix E.

Theorem 7: Let $\{\mathbf{X}_t\}$ *be the iterative sequence produced by* (13)*, we say* $\{\mathbf{X}_t\}$ *is a bounded iterative sequence, i.e.,* $\sum_{t=1}^{\infty} \|\mathbf{X}_{t+1} - \mathbf{X}_t\|_F^2 < \infty$.

For the convex optimization problem in Theorem 3, the proximal mapping in [45] is denoted by $G_{\frac{1}{\rho}f_2}(\mathbf{X}_t) = prox_{\frac{1}{\rho}f_2}\left(\mathbf{X}_t - \frac{1}{\rho}\nabla f_1(\mathbf{X}_t)\right) - \mathbf{X}_t$. In particular, when f_2 is convex, $\left\|G_{\frac{1}{\rho}f_2}(X_t)\right\|_2^2$ can be used to conduct the convergence analysis. In contrast, if f_2 is nonconvex, it is no longer applicable. Hence, we use $\left\|G_{\frac{1}{\rho}f_2}(\mathbf{X}_t)\right\|_F^2 = \|\mathbf{X}_{t+1} - \mathbf{X}_t\|_F^2$ instead to perform convergence analysis of the proposed algorithm. The convergence of Algorithm 3 is shown in the following theorem, and the proof can be found in Appendix F.

Theorem 8 (Main Result 2): Let $\{\mathbf{X}_t\}$ *be the iterative sequence in Algorithm 3. For the consecutive elements* $\mathbf{X}_t$ *and* $\mathbf{X}_{t+1}$*, we have*

$$\min_{t=1,\cdots,T} \|\mathbf{X}_{t+1} - \mathbf{X}_t\|_F^2 \leq \frac{2}{(\rho-\beta)T}\left(F(\mathbf{X}_1) - \inf F\right). \quad (19)$$

2) TIME COMPLEXITY

Assume that $\mathbf{Y}_t$ in step 5 of Algorithm 3 has k_t columns at the current iteration. Consequently, step 5 takes $\mathcal{O}\left(nk_t^2\right)$ time. Next, step 3 shows that $\mathbf{Z}_t$ is a linear combination of $\mathbf{X}_{t-1}$ and $\mathbf{X}_t$. Let $c_t = (\alpha_{t-1} - 1)/\alpha_t$. By combining step 3 and step 4, we have

$$\mathbf{Z}_{ig} = \{c_1\mathbf{X}_t + c_2\mathbf{X}_{t-1} + c_3\mathbf{A}\} + c_4\mathcal{P}_\Omega(\mathbf{Z}_t - \mathbf{A}). \quad (20)$$

where $c_1 = (1 + c_t)(1 - c_3)$, $c_2 = c_t(c_3 - 1)$, $c_3 = \frac{2(1-\omega)}{\rho}$, and $c_4 = \frac{2(1-2\omega)}{\rho}$. The first three terms involve low-rank matrices, whereas the last term involves a sparsity structure. The combined structure in (20) was studied specifically in [42]. Consider the multiplication of $\mathbf{Z}_{ig}$ and a vector $\mathbf{b} \in \mathbb{R}^n$. For the low-rank part, the multiplication costs $\mathcal{O}((m+n)k_t)$ time, whereas the sparse part cost $\mathcal{O}(\|\Omega\|_1)$ time. Hence, the cost is $\mathcal{O}((m+n)k_t + \|\Omega\|_1)$ per vector multiplication. Step 7 performs a rank-k_t SVD of $\mathbf{W}^\top\mathbf{Z}_{ig}$, and it takes $\mathcal{O}\left((m+n)k_t^2 + \|\Omega\|_1 k_t\right)$ time. In summary, the order of the time complexity at the current iteration is $\mathcal{O}\left((m+n)k_t^2 + \|\Omega\|_1 k_t\right)$,where $\|\Omega\|_1 < mn$ and $k_t < n$. Therefore, the time complexity of PUMC_N is much cheaper than the $\mathcal{O}\left(mn^2\right)$ complexity of GPG in [20] and the $\mathcal{O}(mn)$ complexity of BiasMC in [18].

V. EXPERIMENTS

In this section, we perform experiments on synthetic and real-world data sets and demonstrate the effectiveness of the proposed algorithm in practical applications, including link prediction, topology inference, and recommender system. All experiments are implemented in Matlab on Windows 10 with Intel Xeon CPU (2.8GHz) and 128GB memory.

A. SYNTHETIC DATA

Data Sets: As in [18], [41], we assume the matrix $\mathbf{Q} \in \mathbb{R}^{m\times m}$ is generated by $\mathbf{Q} = \mathbf{M}_1\mathbf{M}_2$, where the elements of $\mathbf{M}_1 \in \mathbb{R}^{m\times k}$ and $\mathbf{M}_2 \in \mathbb{R}^{k\times m}$ are obtained from the Gaussian distribution $\mathcal{N}(0, 1)$. Therefore, the underlying binary matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ can be generated by $M_{ij} = I_{Q_{ij}\geq q}$, without loss of generality, we assume that $q = 0.5$. We fix $k = 5$ and very m in {50, 100, 500, 1000, 2000}.

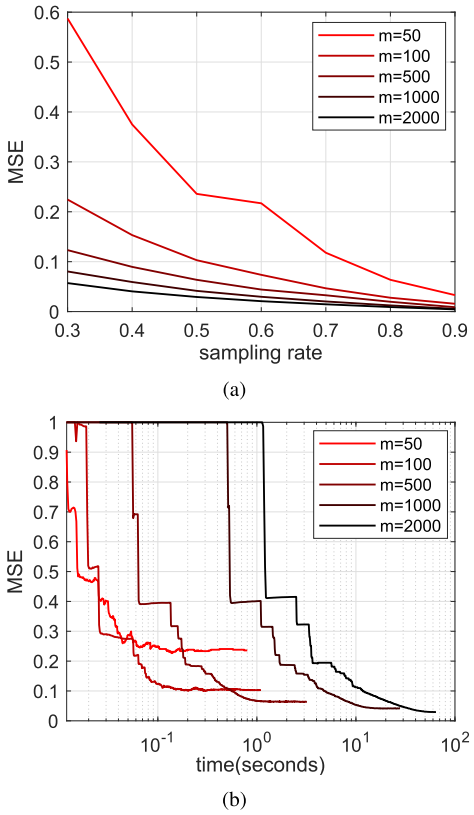


FIGURE 2. Performance analysis of the proposed PUMC_N algorithm on synthetic datasets. (a) MSE vs sampling rate on synthetic datasets. (b) The sampling rate is fixed at 0.5, MSE vs time (in seconds) on synthetic data sets.

For each scenario, the mean square error $MSE = \left\|\mathcal{P}_{\bar{\Omega}}(\mathbf{X} - \mathbf{M})\right\|_F^2 \Big/ \left\|\mathcal{P}_{\bar{\Omega}}(\mathbf{M})\right\|_F^2$ is used for performance evaluation, where $\mathbf{X}$ is the recovery matrix for underlying matrix $\mathbf{M}$, and $\bar{\Omega}$ is the index set of unobserved elements. Each experiment is repeated ten times with the sampling rate (δ) varying from 0.3 to 0.9, and the average results are reported. From Theorem 1, if the sampling rate $\delta = 0.3$ (only 30% 1's in $\mathbf{M}$ are observed), $\omega = 0.15$ is chosen. Results are shown in Fig. 2. Only TNN regularizer (with μ in Table 1 set to 5) is used in synthetic experiments, similar results can be obtained from other nonconvex regularizers in Table 1. Fig. 2(a) shows the testing MSE at different sampling rates. Notice that for an increasing sampling rate we see a monotonous decrease in testing MSE until it is close to zero. This is reasonable since a larger number of observations give rise to more accurate

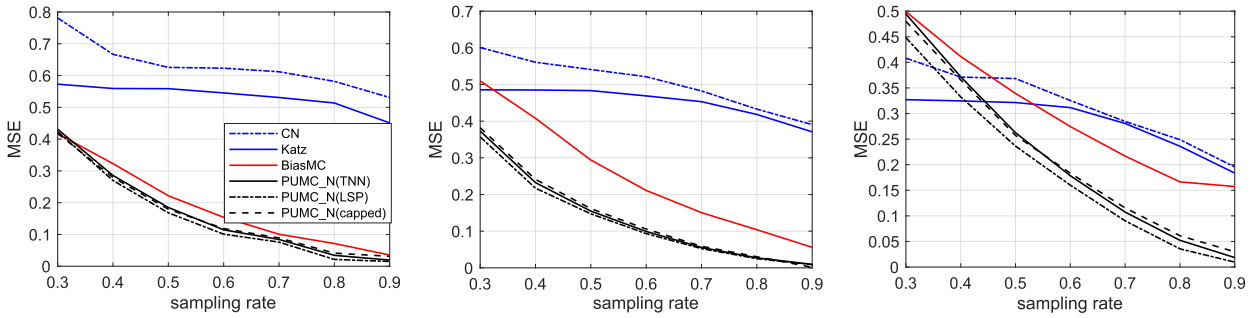


FIGURE 3. Performance comparison of the link prediction methods. (a) Testing MSE vs sampling rate on the *Jazz* data set. (b) Testing MSE vs sampling rate on the *USAir* data set. (c) Testing MSE vs sampling rate on the *PB* data set.

TABLE 2. Link prediction results on three data sets. Testing MSE is rounded to four decimal places and time is in seconds. The best results are highlighted.

		Jazz		*USAir*		*PB*	
		MSE	time	MSE	time	MSE	time
	CN	0.6257	**0.2**	0.5409	**0.2**	0.3683	1.2
	Katz	0.5587	0.3	0.4834	0.4	0.3214	**0.9**
	BiasMC	0.2217	2.3	0.2940	7.1	0.3393	49.5
PUMC_N	TNN	0.1849	1.9	0.1639	2.6	0.2625	10.3
	LSP	**0.1681**	1.8	**0.1385**	3.1	**0.2214**	16.5
	capped ℓ_1	0.1795	1.4	0.1548	1.9	0.2678	8.7

information of the underlying matrix. In addition, there is also a negative correlation between MSE and the matrix size, which is particularly evident at a smaller sampling rate. In particular, if the underlying matrix $\mathbf{M}$ is large enough, it can be recovered accurately at a much small sampling rate (that is small number of observations). In Fig. 2(b), we follow the settings in [41] and fix the sampling rate at 0.5 (from Theorem 1, $\omega = 0.25$). It can be seen that the MSE drops sharply and precipitously at the beginning. Moreover, the larger the matrix, the more time the algorithm takes, and the smaller the MSE.

B. LINK PREDICTION

1) DATA SETS

Consider the link prediction problem in undirected and unweighted social networks. The corresponding adjacency matrix is regarded as the underlying target matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$, where m is the number of nodes. Thus, we suppose that if there is a link between the node i and j, then $M_{ij} = 1$. Otherwise, $M_{ij} = 0$. We perform the experiments of link prediction on three data sets: *Jazz*[2] [15], [48], *USAir*[3] [15], [49], and *PB* [49]. A summary of these three data sets is reported in Table 3.

2) METHODS

We compare three link prediction methods, including i) similarity-based methods [50] Common Neighbors (CN) and Katz (with $\beta = 0.01$), ii) PU matrix completion based method BiasMC [18], iii) the proposed PUMC_N algorithm with the nonconvex regularizers in Table 1, we fix $\mu = 5$ for TNN regularizer, $\mu=\sqrt{\lambda}$ for LSP regularizer, and $\mu= 2\lambda$ for capped ℓ_1.

Afterwards, we also use the mean square error $MSE = \left\|\mathcal{P}_{\bar{\Omega}}(\mathbf{X}-\mathbf{M})\right\|_F / \left\|\mathcal{P}_{\bar{\Omega}}(\mathbf{M})\right\|_F$ for performance evaluation, where $\mathbf{X}$ is the recovery matrix for underlying matrix $\mathbf{M}$, and $\bar{\Omega}$ is the index set of unobserved elements. Each experiment is repeated ten times, and the average results are reported.

Fig. 3 shows the testing MSE under varying sampling rate from 0.3 to 0.9 on the three data sets. As can be seen, the performance improvement is universal for an increasing sampling rate. In general, the performance of matrix completion based methods (BiasMC and PUMC_N) is superior to similarity-based methods. Furthermore, nonconvex regularizers, including TNN, LSP, and capped ℓ_1, lead to a lower testing MSE than other methods. More precisely, PUMC_N clearly outperforms the other three on *Jazz* and *USAir*, and when the sampling rate is not less than 0.5, PUMC_N is superior over others on *PB*.

TABLE 3. Summary of data sets in link prediction and topology inference experiments.

	#nodes	#edges
Jazz	198	2,742
USAir	332	2,126
PB	1,224	16,715
as4	32	161
contact	274	2,124
Wikivote	889	2,914

[2]http://konect.uni-koblenz.de/networks/arenas-jazz.
[3]http://snap.stanford.edu/data/.

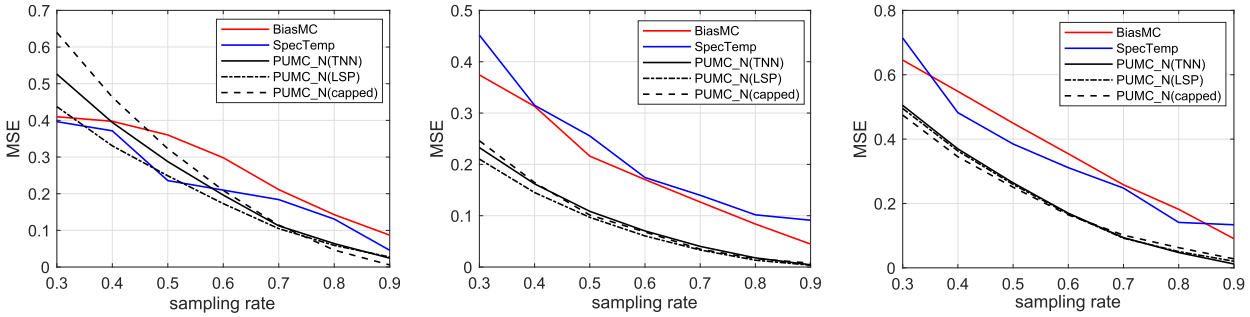


FIGURE 4. Performance comparison of the topology inference methods. (a) Testing MSE vs sampling rate on the *as4* data set. (b) Testing MSE vs sampling rate on the *contact* data set. (c) Testing MSE vs sampling rate on the *Wikivote* data set.

TABLE 4. Topology inference results on three data sets and time is in seconds.

		as4		*contact*		*Wikivote*	
		MSE	time	MSE	time	MSE	time
	BiasMC	0.3602	**0.1**	0.2161	2.7	0.4496	27.9
	SpecTemp	**0.2357**	0.9	0.2554	5.5	0.3849	72.9
PUMC_N	TNN	0.2876	**0.1**	0.1086	1.6	0.2641	9.2
	LSP	0.2489	**0.1**	**0.0969**	1.2	0.2592	7.1
	capped ℓ_1	0.3226	**0.1**	0.1013	**0.9**	**0.2505**	**5.4**

In addition, we follow the settings in [41], 50% of the observations are used for training and the rest for testing. Table 2 shows the testing MSE and time of each method. It can be seen that PUMC_N leads to the lowest testing MSE. In particular, the testing MSE of PUMC_N is about three times smaller than similarity-based methods. However, it is not the fastest solver, especially if the matrix size is large. Moreover, the MSE vs time of the proposed method on these three data sets is provided in Appendix B.

C. TOPOLOGY INFERENCE

1) DATA SETS

We follow the setup in link prediction, and our goal here is recovering the complete adjacency matrix from the incomplete observations. Another three data sets (Table 3) are used for the topology inference experiments, including i) *as4*[4] [46], which describes different types of interactions among the students, ii) *contact*[5] [15], [51], which represents contacts between people measured by carried wireless devices, iii) *Wikivote*[6] [52], which contains the Wikipedia voting data from its inception till January 2008.

2) METHODS

We compare the performance of the proposed PUMC_N algorithm with BiasMC method [18] as well as recent GSP-based method SpecTemp [46]. It is worth noting that the sampling rate in SpecTemp is the percentage of eigenvectors (spectral templates) available. Each experiment is repeated ten times with the sampling rate varying from 0.3 to 0.9, and the average results are reported.

Results are shown in Fig. 4. As can be seen from Fig. 4(a), PUMC_N is not superior to other topology inference methods when the underlying matrix is small, especially at a low sampling rate. More precisely, testing MSEs of PUMC_N are larger than the counterpart of SpecTemp. However, it also can be seen from Fig. 4 that for an increasing matrix size, the superiority of PUMC_N to others is obvious.

Similarly, Table 4 shows the testing MSE and time of each method when the sampling rate $\delta = 0.5$. It can be seen that the testing MSE of PUMC_N is about two times smaller than competitors on *contact* and *Wikivote*. And it is worth noting that PUMC_N is the fastest algorithm among the three methods. Moreover, the MSE vs time on these three data sets is provided in Appendix B.

TABLE 5. Summary of Recommender system data sets.

	#users	#items	#ratings
MovieLens-100K	943	1,682	100,000
FlimTrust	1,508	2,071	35,497
Douban	3,000	3,000	136,891

D. RECOMMENDER SYSTEM

1) DATA SETS

In this practical setting, the popular data sets (Table 5), including *MovieLens (100K)*[7] [16], *FlimTrust*[8] [53], and

[4] http://vladowiki.fmf.uni-lj.si/doku.phpid=pajek:data:pajek:students.
[5] http://konect.uni-koblenz.de/networks/contact.
[6] http://networkrepository.com/soc-wiki-Vote.php.
[7] http://vladowiki.fmf.uni-lj.si/doku.phpid=pajek:data:pajek:students.
[8] https://www.librec.net/datasets.html.

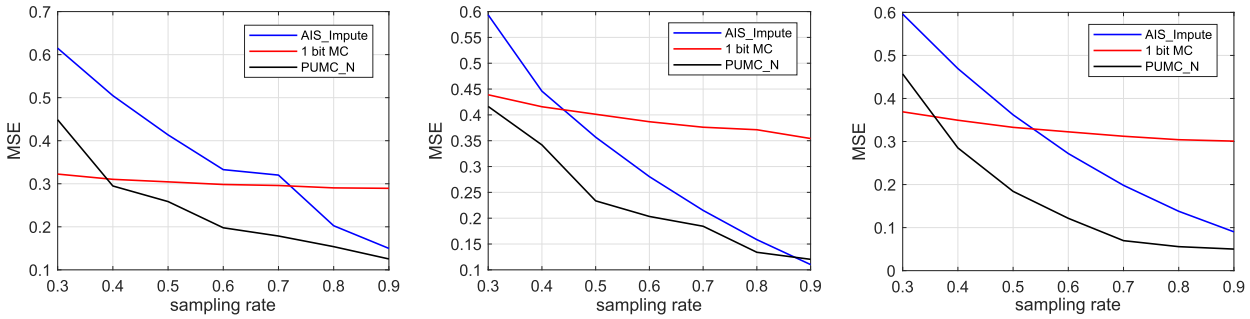


FIGURE 5. Performance comparison of the recommender system methods. (a) Testing MSE vs sampling rate on the *MovieLens* data set. (b) Testing MSE vs sampling rate on the *FlimTrust* data set. (c) Testing MSE vs sampling rate on the *Douban* data set.

TABLE 6. Recommender system results on three data sets and time is in seconds.

	MovieLens-100K		*FlimTrust*		*Douban*	
	MSE	time	MSE	time	MSE	time
AIS-Impute	0.5759	19.6	0.5454	33.9	0.6017	132.4
1-bit matrix completion	0.3044	164.1	0.4012	597.5	0.3329	2298.4
PUMC_N (LSP)	**0.2586**	**12.4**	**0.2336**	**27.2**	**0.1842**	**92.7**

Douban[9] [54], are used to evaluate the performance of our algorithm. We follow the setup in [16], [19] and convert these ratings in each data set to binary observations by comparing each rating to the average value (which is ∼3, considering three data sets together) of whole data sets.

2) METHODS

We compare with the nuclear norm based algorithm AIS-Impute [32], as well as 1-bit matrix completion in [19]. In particular, the AIS-Impute can be considered as an accelerated and inexact version of the proximal algorithm, and the 1-bit matrix completion is constrained by infinity norm and nuclear norm.

Results are shown in Fig. 5. Each point in the figure is the average across ten replicate experiments. Moreover, Table 6 shows the performance of the mentioned methods on three data sets. From the above experiments, LSP regularizer usually has better or comparable performance than the other two regularizers, thus we only use LSP regularizer here. It can be seen that in the three recommended algorithms, as long as the sampling rate is not less than 0.4, PUMC_N will result in the lowest MSE in the least time. In addition, when the sampling rate is low, the performance of PUMC_N needs to be improved. Moreover, the MSE vs time on the above three data sets is provided in Appendix B.

VI. CONCLUSION

In this paper, we addressed the problem of binary matrix completion with nonconvex regularizers, where the observations consist only of positive entries. We proposed a novel PU matrix completion model (4) for tackling the task based on the commonly used nonconvex regularizers and the ω-weighted loss. In particular, the error bound for the model was derived to show that the underlying matrix $\mathbf{M} \in \{0, 1\}^{m\times n}$ can be recovered accurately. Accordingly, we improved the proximal algorithm with two main acceleration strategies in nonconvex settings for solving (4), and the convergence can also be guaranteed. The experiments on both synthetic and real-world data sets have verified the effectiveness of the proposed approach and validated the superiority over the state-of-the-art methods.

There still remain several directions for further work. From the experimental results, it can be seen that there is still room for further performance improvements at a low sampling rate. Additionally, as in [24], [25], a general nonuniform sampling distribution will be considered. In addition, to further accelerate the proposed algorithm and apply it to massive data sets, we will focus on its distributed version.

APPENDIX

A. THE CORRESPONDING CONVEX FUNCTIONS

Table 7 presents the corresponding convex functions of nonconvex regularizers mentioned in Table 1, where $\widehat{r}_n(\mathbf{X}) = \sum_{i=1}^{m} \widehat{r}(\mathbf{X})$ and $\breve{r}_n(\mathbf{X}) = \sum_{i=1}^{m} \breve{r}(\mathbf{X})$. In Table 7, it can be seen that the nonconvex functions r can be represented by two convex functions $\widehat{r}$ and $\breve{r}$, which satisfy the third characteristic of (2).

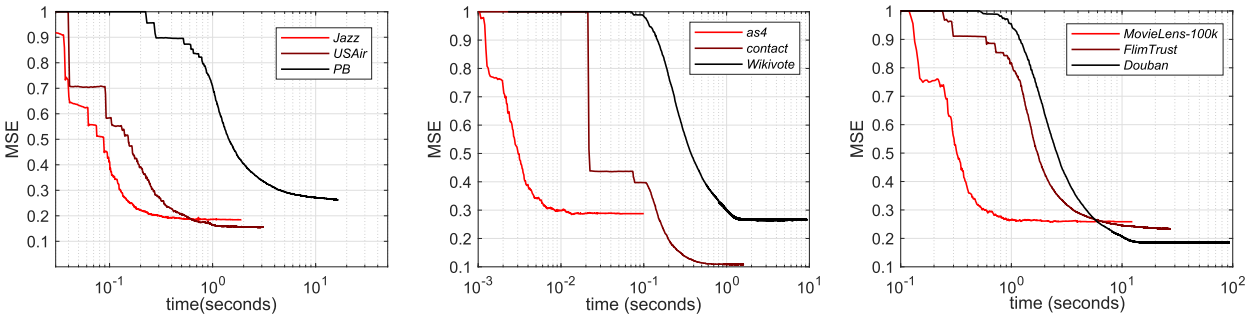
B. TESTING MSE VS TIME ON REAL-WORLD DATA SETS

Fig. 6 shows the MSE vs time of the proposed PUMC_N algorithm on real-world data sets. We follow the settings in [41] and fix the sampling rate at 0.5 (from Theorem 1, $\omega = 0.25$). Similar to Fig. 2(b), the testing MSE drops sharply

[9] https://github.com/fmonti/mgcnn.

TABLE 7. The corresponding convex functions $\widehat{r}$ and $\breve{r}$ of the nonconvex regularizers mentioned in Section II, where $\mu > 0$. For the TNN regularizer, μ is the number of leading singular values that are not penalized, $\mathbf{A} = \mathbf{U}_\mu$, $\mathbf{B} = \mathbf{V}_\mu$, where $\mathbf{U}\Sigma\mathbf{V}^\top$ is the SVD of $\mathbf{X}$.

	$r(\sigma_i(\mathbf{X}))$	$\widehat{r}(\sigma_i(\mathbf{X}))$	$\breve{r}(\sigma_i(\mathbf{X}))$
TNN	$\begin{cases} 0 & , i \le \mu \\ \sigma_i(\mathbf{X}) & , i > \mu \end{cases}$	$\sigma_i(\mathbf{X})$	$\left[diag\left(\mathbf{AXB}^\top\right)\right]_i$
capped ℓ_1	$\begin{cases} \sigma_i(\mathbf{X}) & , \sigma_i(\mathbf{X}) \le \mu \\ \mu & , \sigma_i(\mathbf{X}) > \mu \end{cases}$	$\sigma_i(\mathbf{X})$	$[\sigma_i(\mathbf{X}) - \mu]_+$
LSP	$\log(\sigma_i(\mathbf{X})/\mu + 1)$	$\sigma_i(\mathbf{X})$	$(\sigma_i(\mathbf{X}) - \log(\sigma_i(\mathbf{X})/\mu + 1))$



(a) (b) (c)

FIGURE 6. Performance comparison. (a) Testing MSE vs time on the link prediction data sets. (b) Testing MSE vs time on the topology inference data sets. (c) Testing MSE vs time on the recommender system data sets.

and precipitously at the beginning. Moreover, the larger the matrix, the more time the PUMC_N algorithm takes.

C. PROOF OF THEOREM 2

On the one hand, we find that $\min_{\mathbf{X}} R(\mathbf{X}) = 0$, when. From Theorem 1, we have

$$\begin{aligned} & R_{\hat{\omega}}(\mathbf{X}) - \min_{\mathbf{X}} R_{\hat{\omega}}(\mathbf{X}) \\ &= (\kappa R(\mathbf{X}) + c) - \min_{\mathbf{X}} (\kappa R(\mathbf{X}) + c) \\ &= \kappa \left(R(\mathbf{X}) - \min_{\mathbf{X}} R(\mathbf{X}) \right) \\ &= \kappa R(\mathbf{X}) \end{aligned} \tag{A1}$$

On the other hand, similar to the Theorem 1 in [18], we have the upper bound of ω-weighted expected error.

$$\begin{aligned} & R_{\hat{\omega}}(\mathbf{X}) - \min_{\mathbf{X}} R_{\hat{\omega}}(\mathbf{X}) \\ & \le \frac{C_1}{\delta} \left(\sqrt{\frac{\log(2/\alpha)}{mn}} + \frac{\sqrt{m} + \sqrt{n} + \sqrt[4]{|\Omega_1|}}{mn} \right). \end{aligned} \tag{A2}$$

Consequently, combining (A1) and (A2), we have

$$\begin{aligned} & \frac{1}{mn} \left\| \hat{\mathbf{X}} - \mathbf{M} \right\|_F^2 \\ & \le \frac{2}{\delta} \left[\frac{C_1}{\delta} \left(\sqrt{\frac{\log(2/\alpha)}{mn}} + \frac{\sqrt{m} + \sqrt{n} + \sqrt[4]{|\Omega_1|}}{mn} \right) \right] \\ & \le \frac{C}{\delta^2} \left(\sqrt{\frac{\log(2/\alpha)}{mn}} + \frac{\sqrt{m} + \sqrt{n} + \sqrt[4]{|\Omega_1|}}{mn} \right). \end{aligned} \tag{A3}$$

where $\hat{\mathbf{X}}$ is the solution of (4), and $C = 2C_1$ is absolute constant. This completes the proof of Theorem 2.

D. PROOF OF THEOREM 6

According to the conditions in Theorem 6, we have

$$\mathbf{W}\mathbf{W}^\top = \mathbf{W}^\top\mathbf{W} = \mathbf{I},\ span(\mathbf{U}_k) \subseteq span(\mathbf{W}). \tag{A4}$$

And then let $\mathbf{W} = \left[\mathbf{W}_\parallel; \mathbf{W}_\perp\right]$, where $\mathbf{W}_\perp\mathbf{U}_k = 0$, and $span(\mathbf{U}_k) = span\left(\mathbf{W}_\parallel\right)$, and we must have

$$\mathbf{U}_k\mathbf{U}_k^\top = \mathbf{W}_\parallel\mathbf{W}_\parallel^\top. \tag{A5}$$

Hence, $\mathbf{W}^\top\mathbf{Z} = \left[\mathbf{W}_\parallel; \mathbf{W}_\perp\right]^\top\mathbf{Z}$ and its rank-k SVD is $\left[\mathbf{W}_\parallel; 0\right]^\top\mathbf{U}_k\Sigma_k\mathbf{V}_k^\top$.

According to Theorem 5, (15) can be rewritten as

$$prox_{\frac{\lambda}{\rho}r_n}\left(\mathbf{W}^\top\mathbf{Z}\right) = \begin{bmatrix} \mathbf{W}_\parallel^\top \\ 0 \end{bmatrix} \mathbf{U}_k\hat{\Sigma}_k\mathbf{V}_k^\top. \tag{A6}$$

where $\hat{\Sigma}_k$ is the set of solution in (16). And then,

$$\begin{aligned} \mathbf{W}prox_{\frac{\lambda}{\rho}r_n}\left(\mathbf{W}^\top\mathbf{Z}\right) &= \left[\mathbf{W}_\parallel; \mathbf{W}_\perp\right] \begin{bmatrix} \mathbf{W}_\parallel^\top \\ 0 \end{bmatrix} \mathbf{U}_k\hat{\Sigma}_k\mathbf{V}_k^\top \\ &= \mathbf{W}_\parallel\mathbf{W}_\parallel^\top\mathbf{U}_k\hat{\Sigma}_k\mathbf{V}_k^\top \end{aligned} \tag{A7}$$

Combining (A5) and (A7), we have

$$\begin{aligned} \mathbf{W}prox_{\frac{\lambda}{\rho}r_n}\left(\mathbf{W}^\top\mathbf{Z}\right) &= \mathbf{W}_\parallel\mathbf{W}_\parallel^\top\mathbf{U}_k\hat{\Sigma}_k\mathbf{V}_k^\top \\ &= \mathbf{U}_k\left(\mathbf{U}_k^\top\mathbf{U}_k\right)\hat{\Sigma}_k\mathbf{V}_k^\top \\ &= \mathbf{U}_k\hat{\Sigma}_k\mathbf{V}_k^\top \\ &= prox_{r_n}(\mathbf{Z}) \end{aligned} \tag{A8}$$

This completes the proof the Theorem 6.

E. PROOF OF THEOREM 7

From Lemma 1, we have

$$F(\mathbf{X}_{t+1}) \leq F(\mathbf{X}_t) - \frac{\rho-\beta}{2}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2. \quad (A9)$$

Let $t \in [1, T]$, and sum the formula (2) of every iteration, we have

$$F(\mathbf{X}_{T+1}) \leq F(\mathbf{X}_1) - \frac{\rho-\beta}{2}\sum_{t=1}^{T}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2. \quad (A10)$$

That is,

$$\sum_{t=1}^{T}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2 \leq \frac{2}{\rho-\beta}\left(F(\mathbf{X}_1)-F(\mathbf{X}_{T+1})\right). \quad (A11)$$

Let $T \to +\infty$, and from the characteristics in (2), $F(\mathbf{X})$ is bounded from below, we have

$$F(\mathbf{X}_1) - F(\mathbf{X}_{T+1}) < \infty. \quad (A12)$$

That is,

$$\sum_{t=1}^{T}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2 \leq \frac{2}{\rho-\beta}\left(F(\mathbf{X}_1)-F(\mathbf{X}_{T+1})\right) < \infty. \quad (A13)$$

This completes the proof the Theorem 7. And we must have $\lim_{t\to\infty}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2 = 0$. Hence, the iteration sequence has limit point.

F. PROOF OF THEOREM 8

From the proof of Theorem 7 (Appendix E), we have

$$\begin{aligned}\min_{t=1,2,\cdots,T}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2 &\leq \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2 \\ &\leq \frac{2}{T(\rho-\beta)}\left(F(\mathbf{X}_1)-F(\mathbf{X}_{T+1})\right). \quad (A14)\end{aligned}$$

And then,

$$\begin{aligned}\min_{t=1,2,\cdots,T}\|\mathbf{X}_{t+1}-\mathbf{X}_t\|_F^2 &\leq \frac{2}{T(\rho-\beta)}\left(F(\mathbf{X}_1)-F(\mathbf{X}_{T+1})\right) \\ &\leq \frac{2}{T(\rho-\beta)}\left(F(\mathbf{X}_1)-\inf F\right). \quad (A15)\end{aligned}$$

This completes the proof the Theorem 8.

REFERENCES

[1] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.
[2] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.
[3] B. Recht, "A simpler approach to matrix completion," *J. Mach. Learn. Res.*, vol. 12, pp. 3413–3430, Jan. 2011.
[4] A. Rohde and A. Tsybakov, "Estimation of high-dimensional low-rank matrices," *Ann. Statist.*, vol. 39, no. 2, pp. 887–930, 2011.
[5] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.
[6] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *J. Mach. Learn. Res.*, vol. 11, pp. 2057–2078, Mar. 2010.
[7] T. T. Cai and W. Zhou, "Matrix completion via max-norm constrained optimization," *Electron. J. Stat.*, vol. 10, no. 1, pp. 1493–1525, 2016.
[8] Q. Sun, S. Xiang, and J. Ye, "Robust principal component analysis via capped norms," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Chicago, IL, USA, 2013, pp. 311–319.
[9] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, and I. S. Kweon, "Partial sum minimization of singular values in robust PCA: Algorithm and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 744–758, Apr. 2016.
[10] Z. Wang, M. J. Lai, Z.-S. Lu, W. Fan, H. Davulcu, and J.-P. Ye, "Orthogonal rank-one matrix pursuit for low rank matrix completion," *SIAM J. Sci. Comput.*, vol. 37, no. 1, pp. A488–A514, Jan. 2015.
[11] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2409–2419, Mar. 2018.
[12] C. Liu, H. Shan, and B. Wang, "Wireless sensor network localization via matrix completion based on bregman divergence," *Sensors*, vol. 18, no. 9, pp. 2974–2991, Sep. 2018.
[13] C. Lu, J. Tang, S. Yan, and Z. Lin, "Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 829–839, Feb. 2016.
[14] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 183–208, Jan. 2017.
[15] R. Pech, D. Hao, L. Pan, H. Cheng, and T. Zhou, "Link prediction via matrix completion," *Europhys. Lett.*, vol. 117, no. 3, p. 38002, Mar. 2017.
[16] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters, "1-bit matrix completion," *Inf. Inference*, vol. 3, no. 3, pp. 189–223, 2014.
[17] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Las Vegas, NV, USA, 2008, pp. 213–220.
[18] C. J. Hsieh, N. Natarajan, and I. S. Dhillon, "PU learning for matrix completion," *J. Mach. Learn. Res.*, vol. 37, pp. 2445–2453, Jul. 2015.
[19] M. Hayashi, T. Sakai, and M. Sugiyama. (2018). "Binary matrix completion using unobserved entries." [Online]. Available: https://arxiv.org/abs/1803.04663
[20] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin, "Generalized singular value thresholding," in *Proc. 29th AAAI Conf. Artif. Intell.*, Austin TX, USA, 2015, pp. 1805–1811.
[21] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, 2008.
[22] T. Zhang, "Analysis of multi-stage convex relaxation for sparse regularization," *J. Mach. Learn. Res.*, vol. 11, pp. 1081–1107, Mar. 2010.
[23] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.
[24] O. Klopp, J. Lafond, É. Moulines, and J. Salmon, "Adaptive multinomial matrix completion," *Electron. J. Statist.*, vol. 9, no. 2, pp. 2950–2975, 2015.
[25] T. Cai and W.-X. Zhou, "A max-norm constrained minimization approach to 1-bit matrix completion," *J. Mach. Learn. Res.*, vol. 14, pp. 3619–3647, Dec. 2013.
[26] J. Wang, J. Shen, and H. Xu. (Apr. 2015). "Social trust prediction via max-norm constrained 1-bit matrix completion." [Online]. Available: https://arxiv.org/abs/1504.06394
[27] S. Bhaskar and A. Javanmard, "1-bit matrix completion under exact low-rank constraint," in *Proc. 49th Annu. Conf. Inf. Sci. Syst.*, Baltimore, MD, USA, 2015, pp. 1–6.
[28] R. Ni and Q. Gu, "Optimal statistical and computational rates for one bit matrix completion," *J. Mach. Learn. Res.*, vol. 51, pp. 426–434, May 2016.
[29] V. Cottet and P. Alquier, "1-bit matrix completion: PAC-Bayesian analysis of a variational approximation," *Mach. Learn.*, vol. 107, no. 3, pp. 579–603, Mar. 2018.
[30] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 1675–1685.
[31] T. Sakai, M. C. du Plessis, G. Niu, and M. Sugiyama, "Semi-supervised classification based on classification from positive and unlabeled data," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 2998–3006.

[32] Q. Yao and J. T. Kwok, "Accelerated inexact soft-impute for fast large-scale matrix completion," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 4002–4008.
[33] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.
[34] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Math. Program.*, vol. 9, pp. 1–19, Sep. 2010.
[35] X. Su, Y. Wang, X. Kang, and R. Tao, "Nonconvex truncated nuclear norm minimization based on adaptive bisection method," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[36] J. Yao, D. Meng, Q. Zhao, W. Cao, and Z. Xu, "Nonconvex-sparsity and nonlocal-smoothness-based blind hyperspectral unmixing," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2991–3006, Jun. 2019.
[37] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic, "One-class matrix completion with low-density factorizations," in *Proc. IEEE Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 1055–1060.
[38] C. Scott, "Calibrated asymmetric surrogate losses," *Electron. J. Statist.*, vol. 6, pp. 958–992, May 2012.
[39] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari, "Learning with Noisy Labels," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 1196–1204.
[40] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.
[41] Q. Yao, J. T. Kwok, T.-F. Wang, and T.-Y. Liu, "Large-scale low-rank matrix learning with nonconvex regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[42] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, no. 3, pp. 2287–2322, Aug. 2019.
[43] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.
[44] H. Li and Z. Lin, "Accelerated proximal gradient methods for nonconvex programming," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 379–387.
[45] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods," *Math. Program.*, vol. 137, nos. 1–2, pp. 91–129, Feb. 2013.
[46] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.
[47] P. Gong, C. Zhang, Z. Lu, J. Huang, and J. Ye, "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," *J. Mach. Learn. Res.*, vol. 28, no. 2, pp. 37–45, Mar. 2013.
[48] P. M. Gleiser and L. Danon, "Community structure in jazz," *Adv. Complex Syst.*, vol. 6, no. 4, pp. 565–573, Dec. 2003.
[49] M. Gao, L. Chen, B. Li, and W. Liu, "A link prediction algorithm based on low-rank matrix completion," *Appl. Intell.*, vol. 48, no. 12, pp. 4531–4550, Dec. 2018.
[50] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 58, no. 7, pp. 1019–1031, 2007.
[51] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, and J. Scott, "Impact of human mobility on opportunistic forwarding algorithms," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 606–620, Jun. 2007.
[52] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *Proc. 28th Int. Conf. Hum. Factors Comput. Syst.*, Atlanta, GA, USA, 2010, pp. 1361–1370.
[53] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel evidence-based Bayesian similarity measure for recommender systems," *ACM Trans. Web*, vol. 10, no. 2, May 2016, Art. no. 8.
[54] F. Monti, M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 3697–3707.

CHUNSHENG LIU received the B.S. and M.S. degrees from Electronic Engineering Institute, Hefei, China, in 2014 and 2016, respectively. He is currently pursuing the Ph.D. degree with the National University of Defense Technology, Hefei. His main research interests include wireless network situation awareness and machine learning.

HONG SHAN received the B.S. and M.S. degrees from the University of Electronic Science and Technology, Chengdu, China, in 1985 and 1988, respectively, and the Ph.D. degree from the College of Communication Engineering, in 1997. He is currently a Professor with the National University of Defense Technology, Hefei, China. His research interests include wireless networks, information systems, and the optimization of protocols for wireless networks.