LOW-RANK MATRICES RECOVERY VIA ENTROPY FUNCTION

Dung N. Tran[†], *Shuai Huang* [†], *Sang Peter Chin*[†♯], *Trac D. Tran*[†]

[†]Department of ECE, JHU, 3400 N. Charles St., Baltimore, MD 21218
[♯]Dept. of CS, BU, 111 Cummington Mall, Boston, MA 02215

ABSTRACT

The low-rank matrix recovery problem consists of reconstructing an unknown low-rank matrix from a few linear measurements, possibly corrupted by noise. One of the most popular method in low-rank matrix recovery is based on nuclear-norm minimization, which seeks to simultaneously estimate the most significant singular values of the target low-rank matrix by adding a penalizing term on its nuclear norm. In this paper, we introduce a new method that requires substantially fewer measurements needed for exact matrix recovery compared to nuclear norm minimization. The proposed optimization program utilizes a *sparsity promoting* regularization in the form of the entropy function of the singular values. Numerical experiments on synthetic and real data demonstrates that the proposed method outperforms stage-of-the-art nuclear norm minimization algorithms.

Index Terms— low-rank matrix recovery, matrix completion, entropy, iteratively reweighted nuclear norm minimization.

1. INTRODUCTION

In a low-rank matrix recovery problem, an unknown matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ with $\text{rank}(\boldsymbol{X}) = r \ll \min\{n_1, n_2\}$ is measured via a linear mapping $\mathcal{A} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^m$, resulting in an observation vector $\boldsymbol{y} = \mathcal{A}(\boldsymbol{X})$. The goal is to recover the low-rank matrix $\boldsymbol{X}$ based on the measurements $(\mathcal{A}, \boldsymbol{y})$. This problem has found numerous applications in various fields such as face recognition [1][2], recommender systems [3], computational biology [4], and linear system identification/ control [5].

A natural way to recover a low-rank matrix $\boldsymbol{X}$ from its linear measurements is to solve the *linearly constrained rank minimization* problem

$$\min_{\boldsymbol{X}} \ \text{rank}(\boldsymbol{X}) \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{y}. \tag{1}$$

However, this problem is known to be NP-hard. Instead of solving (1) directly, one is often interested in a tractable method which minimizes a surrogate of the rank function. Let $\boldsymbol{\sigma}(\boldsymbol{X}) = (\sigma_1(\boldsymbol{X}), ..., \sigma_n(\boldsymbol{X}))$ be the *vector of singular values* of $\boldsymbol{X}$, where $\sigma_1(\boldsymbol{X}) \geq ... \geq \sigma_n(\boldsymbol{X}) \geq 0$ are its singular values, and $n = \min\{n_1, n_2\}$, then $\text{rank}(\boldsymbol{X}) = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_0$. Here, $\|\boldsymbol{\sigma}(\boldsymbol{X})\|_0$ counts the number of nonzero elements of $\boldsymbol{\sigma}(\boldsymbol{X})$. Therefore, a natural relaxation of (1) is to replace the rank function by a sparsity promoting function of the singular value vector. Most of previous works have focused on methods that penalize $\|\boldsymbol{\sigma}(\boldsymbol{X})\|_1$ which is defined as the *sum of the singular values* of $\boldsymbol{X}$. Specifically, a recent heuristic introduced in [6] minimizes this *convex surrogate* of the rank function over the linear constraints, resulting in the *nuclear norm minimization (NNM)* problem

$$\min_{\boldsymbol{X}} \ \|\boldsymbol{X}\|_* \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{y}, \tag{2}$$

where $\|\boldsymbol{X}\|_* = \sum_i \sigma_i(\boldsymbol{X}) = \|\boldsymbol{\sigma}(\boldsymbol{X})\|_1$ is the *nuclear norm* of $\boldsymbol{X}$. Candès et. al. [7] has shown that under certain incoherence conditions on the singular values of the target matrix, solving this convex optimization problem results to a near optimal low-rank solution. However, this assumption may be violated in many practical applications, leading to the suboptimality of the solution of NNM.

In this paper, we propose an alternative approximation to the rank function than nuclear norm. In particular, we introduce the *entropy function* $h : \mathbb{R}^n \to \mathbb{R}_+$ which is defined as

$$h(\boldsymbol{x}) = -\sum_i \frac{|x_i|}{\|\boldsymbol{x}\|_1} \log \frac{|x_i|}{\|\boldsymbol{x}\|_1}, \tag{3}$$

where $\boldsymbol{x} \in \mathbb{R}^n$. We adopt the convention that $0 \log 0 = 0$ and $h(\boldsymbol{0}) = 0$. To recover a low-rank matrix $\boldsymbol{X}$ from its linear measurements, we solve the following *ENtropy Minimization (ENM)* problem

$$\min_{\boldsymbol{X}} \ h(\boldsymbol{\sigma}(\boldsymbol{X})) \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{y}. \tag{4}$$

In the next section, we argue that minimizing the entropy function leads to the sparsity of the singular value vector of the solution. This implies that solving (5) results in a low-rank solution. In Section 3, we show that this nonconvex optimization problem can be solved efficiently by an *iteratively reweighted nuclear norm minimization (IRNN)* procedure. Finally, numerical results on synthetic and real image data are presented in Section 4 demonstrating that the

This work is partially supported by NSF-DMS-1222567 (NSF), FA9550-12-1-0136 (AFOSR), NSF-CCF-1117545, NSF-CCF-1422995 and NSF-EECS-1443936.

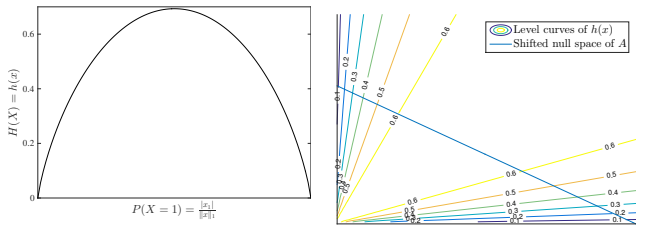
proposed method outperforms state-of-the-art algorithms for solving the nuclear norm minimization problem.

2. ENTROPY MINIMIZATION AND LOW-RANK MATRIX RECOVERY

We show in this section that solving (5) produces a low-rank solution by first arguing that the entropy function promotes sparsity.

Given a nonzero vector $\boldsymbol{x} \in \mathbb{R}^n$, let X be a discrete random variable with possible values $\{1, ..., n\}$. Define $P(X = i) = \frac{|x_i|}{\|\boldsymbol{x}\|_1}$, then $\{\frac{|x_1|}{\|\boldsymbol{x}\|_1}, ..., \frac{|x_n|}{\|\boldsymbol{x}\|_1}\}$ is the distribution of X and $H(X) = h(\boldsymbol{x})$. Here $H(X)$ is the Shannon entropy of the random variable X. Information theory "implies" that the entropy of this random variable is maximized when its distribution is uniform. On the other hand, making the distribution of $\boldsymbol{X}$ skewing towards a few of its values significantly decreases the entropy. This is equivalent to making $\boldsymbol{x}$ more sparse.

To illustrate this point, consider a nonzero vector $\boldsymbol{x} \in \mathbb{R}^2$, and define the binary random variable X as above. The shape of $H(X)$ is plotted in Figure 1(a). This is in fact the well-known binary entropy function. It can be seen that $h(\boldsymbol{x}) = H(X)$ attains its maximum when $x_1 = x_2$ whereas its minima occur when $\boldsymbol{x}$ is 1 sparse, i.e., either x_1 or x_2 is zero.



(a) (b)

Fig. 1. (a) Binary entropy function. (b) Illustration of Lemma 1: minimum entropy occurs at 1-sparse solutions.

We now turn to our problem of interest. Given the linear measurements $(\mathcal{A}, \boldsymbol{y})$ of a low-rank matrix $\boldsymbol{X}$, to recover $\boldsymbol{X}$, we propose to solve

$$\min_{\boldsymbol{X}} \ h(\boldsymbol{\sigma}(\boldsymbol{X})) \quad \text{s.t.} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{y}. \tag{5}$$

For the sake of illustration, consider the case that $\boldsymbol{X}$ is a nonnegative diagonal matrix, and let $\boldsymbol{x} = \text{diag}(\boldsymbol{X})$ so that $\boldsymbol{x} = \boldsymbol{\sigma}(\boldsymbol{X})$. We assume that $\boldsymbol{x}$ is sparse. Note that (5) now reduces to a Compressed Sensing problem. We thus consider the following equivalent problem

$$\min_{\boldsymbol{x}} \ h(\boldsymbol{x}) \quad \text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{6}$$

for some sensing matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m < n$ and measurement vector $\boldsymbol{b} \in \mathbb{R}^m$. The sparsity promoting property of the entropy function leads to sparse minimizers of (6), as shown in the following lemma.

Lemma 1. *Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, $m < n$, be a sampling matrix. Assume that the optimal solution $\boldsymbol{x}^*$ of (6) is unique, then $\|\boldsymbol{x}^*\|_0 \leq m$.*

The implication of this lemma is that the entropy function has a tendency to prefer sparse solutions. As a consequence, solving (5) produces a low-rank solution whose singular value vector is sparse. This lemma can be proved by way of contradiction. In particular, one can assume that $\boldsymbol{x}^*$ has more than m nonzero elements. Consequently, there is some nontrivial vector $\boldsymbol{h}$ in the null space of $\boldsymbol{A}$ that is supported on the support of $\boldsymbol{x}^*$. Then one can construct a sparser solution with smaller entropy by carefully nullifying some elements of $\boldsymbol{x}^*$. This procedure is illustrated in Figure 1(b).

3. ENTROPY-MINIMIZATION ALGORITHMS

In practice, measurements are often concatenated by noise. We thus solve the following robust variant of (5)

$$\min_{\boldsymbol{X}} \lambda h(\boldsymbol{\sigma}(\boldsymbol{X})) + f(\boldsymbol{X}; \mathcal{A}, \boldsymbol{y}), \tag{7}$$

for some loss function $f : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}_+$ with Lipchitz continuous gradient. We will show later in Section 4 that solving (7) leads to a better estimate of the target low-rank matrix compared to NNM. The trade-off is that this optimization is nonconvex. To solve (7), we use a *linearization technique* which replaces both the two terms of the objective function by their affine approximations. Our procedure is similar to that in [8]. However, it is important to note that (7) is not the same as the nonconvex nonsmooth low-rank minimization problems solved in [8] since $h(\cdot)$ is not separable in its parameters.

In a simplified setting, we denote $\boldsymbol{\sigma} = \boldsymbol{\sigma}(\boldsymbol{X})$, $\boldsymbol{\sigma}^t = \boldsymbol{\sigma}(\boldsymbol{X}^t)$, and $f(\boldsymbol{X}) = f(\boldsymbol{X}; \mathcal{A}, \boldsymbol{y})$. As $h(\boldsymbol{\sigma}) \approx h(\boldsymbol{\sigma}^t) + \nabla h(\boldsymbol{\sigma}^t)^T(\boldsymbol{\sigma} - \boldsymbol{\sigma}^t)$, we thus can obtain the solution at iteration $t+1$ based on the solution at iteration t by solving the relaxed problem

$$\begin{aligned} \boldsymbol{X}^{t+1} &= \operatorname*{argmin}_{\boldsymbol{X}} \lambda \nabla h(\boldsymbol{\sigma}^t)^T \boldsymbol{\sigma} + f(\boldsymbol{X}) \\ &= \operatorname*{argmin}_{\boldsymbol{X}} \lambda \sum_i w_i^t \sigma_i + f(\boldsymbol{X}), \end{aligned} \tag{8}$$

where $w_i^t = \frac{\partial h(\boldsymbol{\sigma})}{\partial \sigma_i}|_{\boldsymbol{\sigma}=\boldsymbol{\sigma}^t}$. The following lemma gives the form of the weights used in this weighted nuclear norm minimization problem.

Lemma 2. *Let h be the entropy function defined in (3), and let $\boldsymbol{\sigma}$ be a positive vector, then*

$$\frac{\partial h(\boldsymbol{\sigma})}{\partial \sigma_i} = -\frac{\log \sigma_i}{\|\boldsymbol{\sigma}\|_1} + \frac{\sum_j \sigma_j \log \sigma_j}{\|\boldsymbol{\sigma}\|_1^2}. \tag{9}$$

As a consequence,

$$w_i^t = -\frac{\log \sigma_i^t}{\|\boldsymbol{\sigma}^t\|_1} + \frac{\sum_j \sigma_j^t \log \sigma_j^t}{\|\boldsymbol{\sigma}^t\|_1^2}, \tag{10}$$

for $\sigma_i^t > 0$. In case $\sigma_i^t = 0$, we let $w_i^t = +\infty$. As stated in [8], the above weighted nuclear norm minimization is much more challenging to solve than the weighted ℓ_1 norm minimization [9] as the weighted nuclear norm is nonconvex. To overcome this difficulty, we next linearize the loss function $f(\boldsymbol{X})$ and add a proximal term:

$$f(\boldsymbol{X}) \approx f(\boldsymbol{X}^t) + \nabla f(\boldsymbol{X}^t)^T(\boldsymbol{X} - \boldsymbol{X}^t) + \frac{\rho}{2}\|\boldsymbol{X} - \boldsymbol{X}^t\|_F^2, \tag{11}$$

where $\rho > L_f$ and L_f is the Lipschitz constant of ∇f. Therefore, we can update $\boldsymbol{X}^{t+1}$ by solving the relaxed problem

$$\begin{aligned}\boldsymbol{X}^{t+1} &= \underset{\boldsymbol{X}}{\operatorname{argmin}}\ \lambda\sum_i w_i^t\sigma_i + f(\boldsymbol{X}^t) \\ &\quad + \nabla f(\boldsymbol{X}^t)^T(\boldsymbol{X} - \boldsymbol{X}^t) + \frac{\rho}{2}\|\boldsymbol{X} - \boldsymbol{X}^t\|_F^2 \\ &= \underset{\boldsymbol{X}}{\operatorname{argmin}}\ \lambda\sum_i w_i^t\sigma_i + \frac{\rho}{2}\left\|\boldsymbol{X} - \left(\boldsymbol{X}^t - \frac{1}{\rho}\nabla f(\boldsymbol{X}^t)\right)\right\|_F^2.\end{aligned} \tag{12}$$

Although this problem is nonconvex, it has a closed form solution due to the following property of the weights w_i^t's.

Lemma 3. *If $\sigma_1^t \geq \sigma_2^t \geq ... \geq \sigma_n^t \geq 0$, then $0 \leq w_1^t \leq w_2^t \leq ... \leq w_n^t$.*

We can now obtain the closed form solution of (12) based on this property of the weights.

Lemma 4. *[8][10] Let $\lambda > 0$, $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$, and $0 \leq w_1 \leq w_2 \leq ... \leq w_n$, where $n = \min\{n_1, n_2\}$. Let $\boldsymbol{X}^*$ be the optimal solution of the minimization problem*

$$\min_{\boldsymbol{X}} \lambda\sum_i w_i\sigma_i(\boldsymbol{X}) + \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{Z}\|_F^2, \tag{13}$$

then

$$\boldsymbol{X}^* = \boldsymbol{U}\mathcal{D}_{\lambda\boldsymbol{w}}(\boldsymbol{\Sigma})\boldsymbol{V}^T, \tag{14}$$

where $\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ and $\mathcal{D}_{\lambda\boldsymbol{w}}(\boldsymbol{\Sigma}) = diag\{(\sigma_i - \lambda w_i)_+\}$ is the singular value shrinkage operator [11].

The main steps of the algorithm is summarized in Algorithm 1.

Algorithm 1 Entropy-Minimization

input: measurements $(\mathcal{A}, \boldsymbol{y})$, $\lambda > 0$, and $\rho > L_f$.
initialization: $\boldsymbol{X}^0$.
while not converged **do**
Update the weights:

$$w_i^t = -\frac{\log\sigma_i^t}{\|\boldsymbol{\sigma}^t\|_1} + \frac{\sum_j \sigma_j^t \log\sigma_j^t}{\|\boldsymbol{\sigma}^t\|_1^2},\ i = 1, ..., n \tag{15}$$

Update the estimate:

$$\boldsymbol{X}^{t+1} = \boldsymbol{U}\mathcal{D}_{(\lambda/\rho)\boldsymbol{w}}(\boldsymbol{\Sigma})\boldsymbol{V}^T, \tag{16}$$

where $\boldsymbol{X}^t - \frac{1}{\rho}\nabla f(\boldsymbol{X}^t) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$.
end while
output: Estimated solution $\hat{\boldsymbol{X}}$.

Discussion: Our proposed algorithm for solving the ENM problem is in fact an iteratively reweighted nuclear norm minimization algorithm. It is therefore an improvement of NNM that it more democratically penalizes nonzero singular values. Specifically, Lemma 3 implies that the insignificant singular values at each iteration have larger weights during the subsequent iterations, which would eventually vanish after the algorithm terminates. As a result, the obtained solutions are often have lower ranks comparing to NNM. Finally, in practice, ENM tends to encourage the singular values of X to have a Laplacian distribution.

4. NUMERICAL RESULTS

This section shows various experiments on both synthetic data and real image data to illustrate the effectiveness of the proposed algorithm. We perform experiments for the matrix completion problem

$$\min_{\boldsymbol{X}} \operatorname{rank}(\boldsymbol{X}) \quad \text{s.t.} \quad X_{ij} = M_{ij}, \quad (i,j) \in \Omega \tag{17}$$

where $\boldsymbol{X}$ is the target matrix and Ω is the index set of the observed entries. This problem is a special case of the linearly constrained low-rank matrix recovery problem (1) where $\mathcal{A}$ is a random subsampling operator. In all experiments, we choose the Frobenius norm as the loss function and initialize $\boldsymbol{X}^0$ by Singular Value Thresholding (SVT) [11].

4.1. Synthetic data

In this subsection, we compare the exact recovery ability of our ENM algorithm with that of SVT [11] and Augmented Lagrange Multiplier (ALM) [12] algorithms on synthetic data. These algorithms solves the NNM problem for matrix

completion:

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_* \quad \text{s.t.} \quad X_{ij} = M_{ij}, \quad (i,j) \in \Omega \qquad (18)$$

We used the implementations of these algorithms provided on the authors websites [1].

We fixed the size of the target matrix as $n_1 = 100$ and $n_2 = 100$, and the number of measurements as $m = 0.5 n_1 n_2$. Exact recovery ability of various algorithms is benchmarked against against various numbers of rank r of the target matrix. For each r, the experiment was repeated 100 times. Each time, we first sample two matrices $\boldsymbol{M}_1 \in \mathbb{R}^{n_1 \times r}$ and $\boldsymbol{M}_2 \in \mathbb{R}^{n_2 \times r}$ with i.i.d. Gaussian entries, and set $\boldsymbol{M} = \boldsymbol{M}_1 . \boldsymbol{M}_2^T$. We then sample a subset Ω of m entries uniformly at random. For both SVT and ALM, we used the default parameters in the publicly released code. To evaluate the obtained solutions, we use the Relative Error defined as $\frac{\|\hat{\boldsymbol{X}} - \boldsymbol{M}\|_F^2}{\|\boldsymbol{M}\|_F^2}$, where $\hat{\boldsymbol{X}}$ is the recovered solution employed to evaluate recovery performance. In our experiments, a target matrix is successfully recovered if the Relative Error is less than 10^{-3}. The probability of exact recovery is plotted in Figure 2. It can be seen that our algorithm outperforms SVT and ALM in the sense that it requires significantly less samples to recovery a low-rank matrix comparing to other popular algorithms.

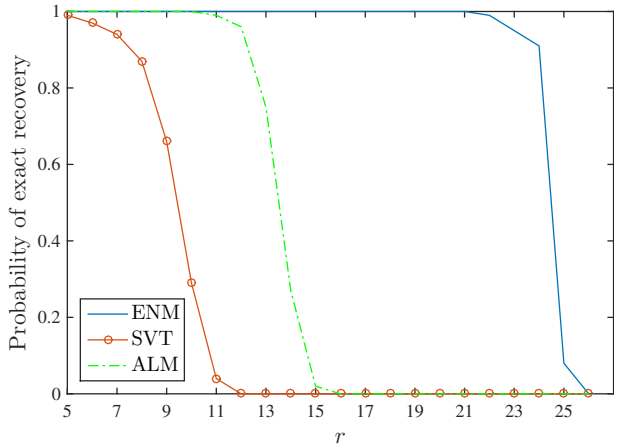


Fig. 2. Probability of exact recovery on synthetic data.

4.2. Image recovery

We now validate the performance of the proposed algorithm on real image recovery problem. The chosen image is the MIT logo which is of size 38×73 and approximately of rank 5 with 5 dominant singular values. Figure 3 shows the gray scale MIT logo image and its singular values. We compare our algorithm with the Accelerated Proximal Gradient with Line search (APGL) algorithm [13] which solves the following NNM problem for matrix completion

$$\min_{\boldsymbol{X}} \lambda \|\boldsymbol{X}\|_* + \frac{1}{2} \|\mathcal{P}_\Omega(\boldsymbol{X} - \boldsymbol{M})\|_F^2, \qquad (19)$$

[1] http://www.perception.csl.illinois.edu/matrix-rank/sample_code.html

where $\mathcal{P}_\Omega$ is a linear operator such that the (i,j)th component of $\mathcal{P}_\Omega(X)$ is equal to X_{ij} if $(i,j) \in \Omega$ and zero otherwise. In our experiments, only a subset of entries of the target image, chosen uniformly at random, is observed. We varied the number of observed elements. For each number of random samples, the experiment is repeated 1000 times. We use the Relative Error defined in the previous subsection to evaluate the performance of the algorithms. The Relative Error curves are shown in Figure 4. We can see that the ENM algorithm performs much better than APGL.

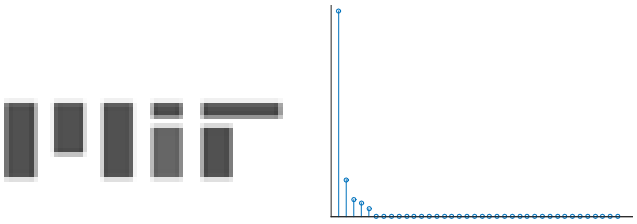


Fig. 3. MIT logo image and its singular values.

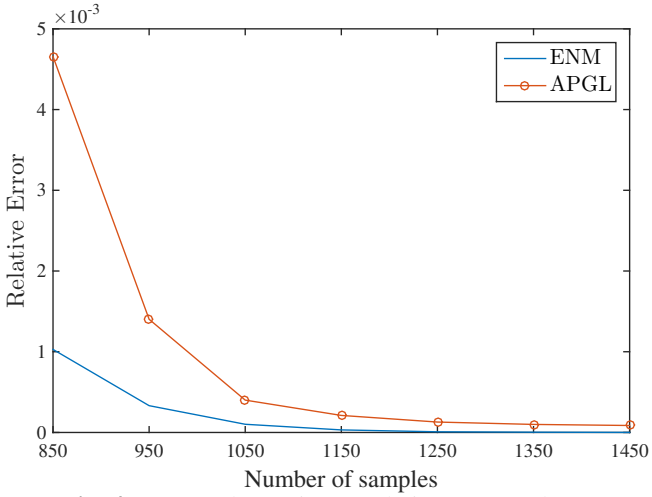


Fig. 4. Low-rank matrix completion on MIT logo.

5. CONCLUSIONS

In this paper, we propose the Entropy-Minimization program for solving low-rank matrix recovery problems. The proposed method minimizes the entropy function of the singular values of the target matrix. We show that minimizers of this program are low-rank and provide a better approximation to original matrices compared to NNM. Moreover, our proposed Entropy-Minimization algorithm can be solved efficiently by an iterative reweighted nuclear norm minimization algorithm. Numerical experiments on both synthetic and real image data demonstrated the superior of the proposed method to state-of-the-art algorithms for solving NNM.

6. REFERENCES

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, May 2011.

[2] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 218–233, 2003.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42(8), pp. 30–37, 2009.

[4] Eric C. Chi, Hua Zhou, Gary K Chen, Diego Ortega Del Vecchyo, and Kenneth Lange, "Genotype imputation via matrix completion," *Genome Research*, vol. 23(3), pp. 509–518, 2013.

[5] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, pp. 471–501, 2010.

[6] M. Fazel, "Matrix rank minimization with applications," *PhD thesis, Stanford University*, 2002.

[7] E. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, 2010.

[8] Canyi Lu, Jinhui Tang, Shuicheng Yan, and Zhouchen Lin, "Generalized nonconvex nonsmooth low-rank minimization," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4130–4137, June 2014.

[9] E. J. Candès, M. B. Wakin, and S. Boyd, "Enhancing sparsity by reweighted l1 minimization," *Journal of Fourier Analysis and Applications, special issue on sparsity*, vol. 14, no. 5, pp. 877–905, December 2008.

[10] K. Chen, H. Dong, and K. Chan, "Reduced rank regression via adaptive nuclear norm penalization," *Biometrika*, 2013.

[11] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, March 2009.

[12] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of a corrupted low-rank matrices," *UIUC Technical Report UILU-ENG-09-2215, Tech. Rep.*, 2009.

[13] K. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of Optimization*, 2010.