A Fast Automatic Low-rank Determination Algorithm for Noisy Matrix Completion

Tatsuya Yokota * [1] and Andrzej Cichocki * †
* RIKEN Brain Science Institute, Japan
E-mail: yokota@brain.riken.jp
† System Research Institute, Polish Academy of Science (PAN), Poland

***Abstract*—Rank estimation is an important factor for low-rank based matrix completion, and most works devoted to this problem have considered the minimization of nuclear norm instead of matrix rank. However, when nuclear norm minimization shifts to 'regularization' due to noise, it is difficult to estimate original matrix rank, precisely. In present paper, we propose a new fast algorithm to precisely estimate matrix rank and perform completion without using nuclear norm. In our extensive experiments, the proposed algorithm significantly outperformed nuclear-norm based method for accuracy, especially and Incremental OptSpace regarding computational time. Our model selection scheme has many promising extensions for constrained matrix factorizations and tensor decompositions, and these extensions could be useful for wide range of practical applications.**

I. Introduction

Matrix completion is an estimation procedure for missing values of incomplete matrix by using only available elements and structural properties. Particularly, the research of low-rank matrix completion has been well progressed from theoretical studies to applications in recent years [4], [3], [2], [12], [10]. Main objective of low-rank matrix completion is to minimize the rank of a matrix which completely fits to the observed matrix for all available elements. However, since it is the NP-hard non-convex optimization problem, nuclear-norm minimization is employed for the optimization:

$$\text{minimize } ||\boldsymbol{X}||_*, \text{ s.t. } P_\Omega(\boldsymbol{X}) = P_\Omega(\boldsymbol{M}), \tag{1}$$

where $\boldsymbol{M} \in \mathbb{R}^{I\times J}$ is an observed matrix, $\boldsymbol{X} \in \mathbb{R}^{I\times J}$ is an output estimated matrix, Ω is a set of indices of available elements, a projection operator $P_\Omega(\cdot)$ is defined by $[P_\Omega(\boldsymbol{X})]_{ij} := \begin{cases} \boldsymbol{X}_{ij} & (i,j)\in\Omega \\ 0 & otherwise \end{cases}$, the equality constraint means $\boldsymbol{X}_{ij} = \boldsymbol{M}_{ij}$ for $(i,j)\in\Omega$, $||\boldsymbol{X}||_* = \sum_{i=1}^{\min(I,J)} \sigma_i(\boldsymbol{X})$ denotes the nuclear norm, and $\sigma_i(\boldsymbol{X})$ denotes the ith largest singular value of matrix $\boldsymbol{X}$. The optimization problem (1) is convex [13] and many efficient algorithms have been proposed as Singular Value Thresholding (SVT) [2], Fixed Point Continuation (FPC) [12], and Inexact Augmented Lagrange Multiplier (IALM) [11].

To apply the low-rank matrix completion for real applications, it is necessary to consider noisy observations as $\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{Z}$, where $\boldsymbol{M}_0$ is an unknown ground-truth low-rank matrix and $\boldsymbol{Z}$ is a zero-mean Gaussian noise matrix that the standard deviation is σ. Candes and Plan [3] proposed a method to solve this challenging problem by the following optimization problem

$$\text{minimize } ||\boldsymbol{X}||_*, \text{ s.t. } ||P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{M})||_F \leq \delta, \tag{2}$$

where $\delta > 0$ is a thresholding parameter assuming $||P_\Omega(\boldsymbol{Z})||_F \leq \delta$, and then δ can be estimated with high probability from $\delta^2 \leq (|\Omega| + \sqrt{8|\Omega|})\sigma^2$. The authors [3] proposed to use the Lagrangian version

$$\text{minimize } \mu||\boldsymbol{X}||_* + \frac{1}{2}||P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{M})||_F^2, \tag{3}$$

with appropriate value of parameter μ by using FPC algorithm [12]. In this method, they tune the value of μ solving (3) so that $||P_\Omega(\boldsymbol{X}^*(\mu)) - P_\Omega(\boldsymbol{M})||_F = \delta$, where $\boldsymbol{X}^*$ denotes the optimization result of (3).

The key point of noisy matrix completion is to estimate exact rank of matrix $\boldsymbol{M}_0$ from noisy entries $\boldsymbol{M}_{ij}$ $(i,j)\in\Omega$. One possible solution is the minimization of nuclear norm, however we observed that nuclear norm minimization estimated always larger rank than the true rank for noisy data. This is because the nuclear norm minimization shifts to regularization in noisy case which may decrease the flexibility of model, and then the larger rank matrix $\boldsymbol{X}$ is required for best fitting. Thus, in nuclear norm minimization, we expect that if estimated matrix rank $\boldsymbol{X}$ is correct, then estimation error is not exactly minimized, and vice versa. Keshavan et. al [9], [8] proposed a criterion without using nuclear norm to estimate matrix rank by

$$\hat{R} = \underset{i}{\text{argmin}} \frac{\sigma_{i+1}(P_\Omega(\boldsymbol{M})) + \sigma_1(P_\Omega(\boldsymbol{M}))\sqrt{i/E}}{i}, \tag{4}$$

where $E = |\Omega|/\sqrt{IJ}$, and implemented an algorithm for matrix completion which is known as OptSpace. Since instability of the criterion (4), the algorithm sometimes fails to estimate a true matrix rank (e.g., too low/high), they also proposed an alternative algorithm of the Incremental OptSpace [9], [8] for ill-conditioned cases. Incremental OptSpace gradually increases R one by one from rank-1 matrix until getting an appropriate fitting, and it works quite well to estimate matrix rank precisely even in ill-conditioned cases, however, high computational complexity of the algorithm, especially for large-scale problems is the main obstacle for its practical applications.

[1]Thanks to JSPS KAKENHI (Grant Number 15K16067) for funding

In this paper, we reveal a disadvantage of nuclear norm minimization and propose a new greedy algorithm for precise matrix rank estimation for noisy low-rank matrix completion by searching optimal rank. The concept of the proposed algorithm, named the greedy low-rank matrix completion (GLMC), is closely related to the Incremental OptSpace, however our algorithm is much simpler and faster, and it works more efficiently even for relatively large matrices.

The rest of this paper is organized as follows. In Section II, we propose a new GLMC method, and its accelerated version. In Section III, we demonstrate experiments for comparison. Discussions are provided in Section IV. Finally, concluding remarks are described in Section V.

II. Proposed method

In our method, we minimize number of components R of linear matrix factorization model $\boldsymbol{U\Sigma V}^T$ instead of matrix rank:

$$\underset{R}{\text{minimize}}\ R, \text{ s.t. } ||P_\Omega(\boldsymbol{U\Sigma V}^T) - P_\Omega(\boldsymbol{M})||_F \leq \delta, \quad (5)$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{R\times R}$ is a diagonal matrix consisting of $[\lambda_1, \lambda_2, ..., \lambda_R]$, $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_R] \in \mathbb{R}^{I\times R}$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_R] \in \mathbb{R}^{J\times R}$ are left and right orthogonal factor matrices, respectively. Actually, R and matrix rank of $\boldsymbol{U\Sigma V}^T$ are equivalent by assuming $\lambda_r > 0,\ \forall r \in \{1, 2, ..., R\}$ and $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I} \in \mathbb{R}^{R\times R}$, where $\boldsymbol{I}$ is an identity matrix. Let $E(R) = \min_{\boldsymbol{U},\boldsymbol{\Sigma},\boldsymbol{V}} ||P_\Omega(\boldsymbol{U\Sigma V}^T) - P_\Omega(\boldsymbol{M})||_F$ and R^* be optimal rank with δ, we have

$$E(1) \geq E(2) \geq \cdots \geq E(R^* - 1) \geq \delta \geq E(R^*). \quad (6)$$

Thus it is not difficult to obtain enough small R^* by gradually increasing $R \leftarrow R + 1$ while we solve the following sub-optimization problem:

$$[\boldsymbol{U}^*, \boldsymbol{\Sigma}^*, \boldsymbol{V}^*] = \underset{\boldsymbol{U},\boldsymbol{\Sigma},\boldsymbol{V}}{\text{argmin}}\, ||P_\Omega(\boldsymbol{U\Sigma V}^T) - P_\Omega(\boldsymbol{M})||_F. \quad (7)$$

This sub-optimization problem is a special case of the manifold optimization step in OptSpace [8] which solves (7) by using a gradient based optimization scheme. In our paper, we propose a new simpler algorithm to solve (7), named as the Greedy Fixed-rank Matrix Completion (GFMC), in Section II-A. By using GFMC we propose a simple and fast algorithm to solve (5) in Algorithm 1. We call the proposed algorithm as Greedy Low-rank Matrix Completion (GLMC).

Algorithm 1 Greedy Low-rank Matrix Completion (GLMC) algorithm

Input: $\boldsymbol{M}$, Ω, δ, and ϵ (tolerance for GFMC)
Initialize: $R \leftarrow 0$;
repeat
  $R \leftarrow R + 1$;
  $\boldsymbol{X} \leftarrow \text{GFMC}(\boldsymbol{M}, \Omega, R, \epsilon)$;
until $||P_\Omega(\boldsymbol{X}) - P_\Omega(\boldsymbol{M})||_F \leq \delta$
Output: $\boldsymbol{X}$

Algorithm 2 Sub-optimization: Greedy Fixed-rank Matrix Completion (GFMC)

Input: $\boldsymbol{M} \in \mathbb{R}^{I\times J}$, Ω, R, ϵ (tolerance)
Initialize: generate $\boldsymbol{U} \in \mathbb{R}^{I\times R}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{R\times R}$, $\boldsymbol{V} \in \mathbb{R}^{J\times R}$, randomly.
$\boldsymbol{Y}_{ij} \leftarrow \begin{cases} \boldsymbol{M}_{ij} & (i,j) \in \Omega \\ (\boldsymbol{U\Sigma V}^T)_{ij} & otherwise \end{cases}$;
repeat
  $E_1 \leftarrow ||\boldsymbol{U\Sigma V}^T - \boldsymbol{Y}||_F$;
  $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow \text{tSVD}(\boldsymbol{Y}, R)$;
  $\boldsymbol{Y}_{ij} \leftarrow (\boldsymbol{U\Sigma V}^T)_{ij}$ for $(i,j) \notin \Omega$;
  $E_2 \leftarrow ||\boldsymbol{U\Sigma V}^T - \boldsymbol{Y}||_F$;
until $|E_1 - E_2| \leq \epsilon$
Output: $\boldsymbol{X} = \boldsymbol{U\Sigma V}^T$

Algorithm 3 Accelerated GLMC algorithm

Input: $\boldsymbol{M}$, Ω, δ, ν (typically 0.01), and ϵ
Initialize: $R \leftarrow 1$, and generate $\boldsymbol{U} \in \mathbb{R}^{I\times R}$, $\boldsymbol{\Sigma} \in \mathbb{R}^{R\times R}$, $\boldsymbol{V} \in \mathbb{R}^{J\times R}$, randomly.
$\boldsymbol{Y}_{ij} \leftarrow \begin{cases} \boldsymbol{M}_{ij} & (i,j) \in \Omega \\ (\boldsymbol{U\Sigma V}^T)_{ij} & otherwise \end{cases}$;
repeat
  $E_1 \leftarrow ||\boldsymbol{U\Sigma V}^T - \boldsymbol{Y}||_F$;
  $[\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}] \leftarrow \text{tSVD}(\boldsymbol{Y}, R)$;
  $\boldsymbol{Y}_{ij} \leftarrow (\boldsymbol{U\Sigma V}^T)_{ij}$ for $(i,j) \notin \Omega$;
  $E_2 \leftarrow ||\boldsymbol{U\Sigma V}^T - \boldsymbol{Y}||_F$;
  if $E_2 > \delta$ and $\frac{|E_1 - E_2|}{|E_2 - \delta|} < \nu$ **then** $R \leftarrow R + 1$;
until $E_2 \leq \delta$ and $|E_1 - E_2| < \epsilon$
Output: $\boldsymbol{X} = \boldsymbol{U\Sigma V}^T$

A. Sub-optimization problem

In this section, we explain the GFMC algorithm. Problem (7) can be transformed to:

$$\text{minimize}\ \ ||\boldsymbol{U\Sigma V}^T - \boldsymbol{Y}||_F^2, \quad \text{s.t. } \boldsymbol{Y}_{ij} = \begin{cases} \boldsymbol{M}_{ij} & (i,j) \in \Omega \\ (\boldsymbol{U\Sigma V}^T)_{ij} & otherwise \end{cases}. \quad (8)$$

It can be solved by the iterations of the following update rules until converge: $[\boldsymbol{U}^k, \boldsymbol{\Sigma}^k, \boldsymbol{V}^k]$ are obtained as the largest R singular values and singular vectors of matrix $\boldsymbol{Y}^k$ by the truncated SVD (tSVD) [1], and $\boldsymbol{Y}_{ij}^{k+1} \leftarrow (\boldsymbol{U}^k\boldsymbol{\Sigma}^k(\boldsymbol{V}^k)^T)_{ij}\ (i,j) \notin \Omega$. The GFMC is summarized in Algorithm 2.

B. Accelerated algorithm

If current value of R is too small to fit the matrix factorization model sufficiently well to the given incomplete matrix during the iteration process in Algorithm 2, then we can stop the algorithm for current R and increase $R \leftarrow R+1$, and run the algorithm again for a new increased R. In this procedure, we switch in early stage of iterations to increased R if the following condition is met:

$$\frac{|E_1 - E_2|}{|E_2 - \delta|} < \nu, \quad (9)$$

where $\nu > 0$ is a stopping threshold (typically, $\nu = 0.01$). Note that the condition (9) means that when the convergence speed becomes relatively slow, we stop the iteration procedure for current R. By incorporating this simple stopping/switching criterion, we finally developed an accelerated algorithm which is summarized in Algorithm 3. When we set ν sufficiently small, Algorithm 3 provides the same result as Algorithm 1, however, computational time is not reduced well for too small ν. Conversely, for relatively larger ν, the algorithm is accceralated very well, however accuracy of the rank estimation is reduced.

III. Experiments

A. Nuclear-norm minimization vs rank minimization for synthetic noisy matrix completion

Next, we investigated the estimation accuracy of matrix rank and missing values for various value of δ. In both methods of FPC [3], [12] and the GLMC, δ is a common key parameter to constrain the errors between noisy observations and estimated values. Obviously, small error δ requires a highly flexible model, then estimated matrix rank becomes relatively large, and vice versa.

In this experiment, we generated R-rank matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ by multiplication of $\boldsymbol{A} \in \mathbb{R}^{N\times R}$ and $\boldsymbol{B} \in \mathbb{R}^{R\times N}$ plus noisy matrix $\boldsymbol{Z} \in \mathbb{R}^{N\times N}$, where $\boldsymbol{A}_{ir} \sim N(0,1)$, $\boldsymbol{B}_{rj} \sim N(0,1)$, and $\boldsymbol{Z}_{ij} \sim N(0,\sigma^2)$. Parameters were set as $N = 100$, $R = 5$, $\sigma = 0.2$. We randomly removed 50% entries from $\boldsymbol{M}$, and applied FPC and GLMC algorithms with various values of δ. Figure 1 shows the result of estimated matrix rank and root mean squared error (RMSE) between the estimator and true low-rank matrix $\boldsymbol{AB}$. The minimum errors were 1.47e-1 by FPC with $\delta = 10.8$ and 9.55e-2 by GLMC with $\delta \in [14.4, 50.5]$. Correct rank was estimated by FPC with $\delta \in [21.6, 108.2]$, and by GLMC with $\delta \in [14.4, 50.5]$. The problem with the FPC is that the δ which guarantee the minimum error is not overlapped with the range of δ performing the correct matrix rank. On the other hand, the best range of δ for GLMC is very consistent for both the RMSE and rank estimation.

Next, we investigate the performance for various settings of true ranks R, missing rates, and noise levels σ. We applied the IALM, FPC-δ, FPC-μ, and the GLMC. Threshold δ for FPC-δ and GLMC was set based on the noise level parameter σ and number of observations by $\delta = \sqrt{(|\Omega| + \sqrt{8|\Omega|})\sigma^2}$, μ for FPC-μ was set by $\mu = (\sqrt{I} + \sqrt{J})\sqrt{|\Omega|/(IJ)}\sigma$, where both optimal values of δ and μ were taken from [3]. Figure 2 shows the result of average $\pm$ standard deviation of estimated matrix rank and RMSE for various problem settings: matrix ranks, missing rates, and noise levels. We can see the significant improvements of rank estimation and RMSE performances by using GLMC algorithm in comparison with the nuclear-norm based completion scheme. Rank estimation by GLMC was almost always correct, and RMSE were reduced 30-40% in comparison to the FPC algorithm. These results indicate the

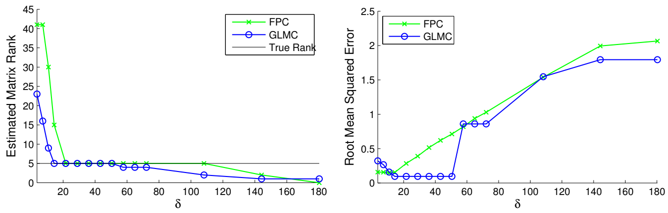
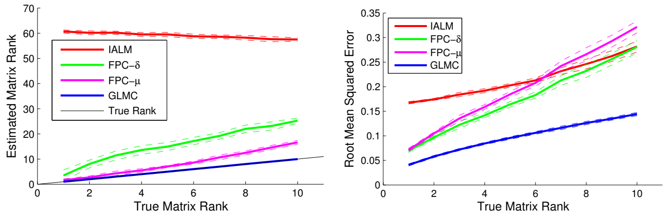
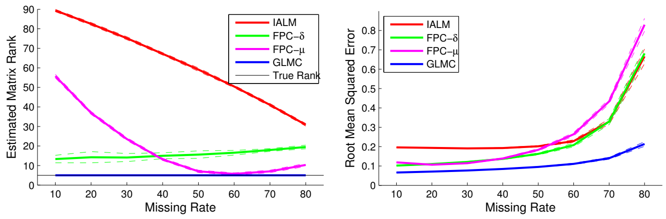


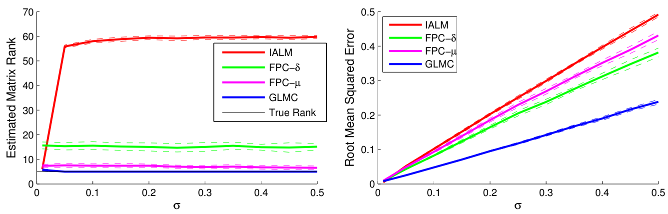
Fig. 1. Estimated rank and root mean squared error for various values of δ.



(a) Various matrix ranks



(b) Various missing rates



(c) Various noise levels σ

Fig. 2. Performance of IALM [11], FPC [3], [12], and GLMC (proposed) algorithms for various settings.

disadvantage of nuclear-norm minimization (regularization) for noisy data.

B. Computational time comparison with OptSpace

As mentioned previously, the proposed method is closely related to a special case of OptSpace and Incremental OptSpace. Since our optimization problem is equivalent to a special case of the OptSpace, we compared only computational times. In our experiments, we generated R-rank matrix $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ in the same way to Section III-A. We randomly removed 50% entries from $\boldsymbol{M}$, set $\sigma = 0.2$, and applied GFMC, Accelerated GLMC, OptSpace [8] and Incremental OptSpace [9] algorithms. We used implementations of OptSpace in C and MATLAB which are distributed in online, and applied our stopping criteria into OptSpace for comparison. Table I shows the computational times of these algorithms in C and MATLAB implementations. Our accelerated GLMC implemented only in MATLAB was generally faster than

TABLE I
COMPARISON OF COMPUTATIONAL TIMES FOR GFMC, ACCELERATED GLMC (AGLMC), AND OPTSPACE (OS), INCREMENTAL OPTSPACE (IOS) ALGORITHMS: -c AND -m DENOTE IMPLEMENTING LANGUAGES OF C AND MATLAB.

[sec]		Completion with True Rank			Completion with Rank Estimation		
N	R	GFMC-m	OS-c	OS-m	AGLMC-m	IOS-c	IOS-m
200	5	0.73	**0.26**	0.47	**0.67**	1.06	2.41
200	10	0.66	**0.58**	1.56	**1.47**	5.76	17.47
200	20	**1.85**	1.96	16.26	**4.04**	33.14	151.53
400	5	0.99	**0.75**	1.43	**1.87**	2.84	8.68
400	10	**1.60**	2.10	4.52	**3.97**	11.20	45.30
400	20	2.81	**2.68**	31.04	**9.09**	41.69	492.64
1000	5	5.05	**3.27**	—	**10.42**	11.79	—
1000	10	7.57	**6.40**	—	**22.37**	41.90	—
1000	20	**12.36**	15.35	—	**49.62**	161.60	—
1000	50	**35.10**	48.62	—	**171.85**	808.73	—
1000	100	**116.60**	197.71	—	**515.36**	4239.42	—

Incremental OptSpace for both implementations.

IV. DISCUSSION

A. Advantage of the proposed algorithm

In this paper, we discussed the methods of automatic rank determination for noisy low-rank incomplete matrix, and revealed some problems of nuclear-norm based methods and Incremental OptSpace. For noisy data, nuclear-norm based method tunes a trade-off parameter μ by solving FPC algorithm iteratively, and Incremental OptSpace tunes R by solving OptSpace with fixed rank R iteratively. Both algorithms are essentially similar from this aspect. The problem with FPC and OptSpace algorithms is that they use highly redundant procedures many times, so their computational costs are high. On the other hand, the proposed AGLMC algorithm can reduce such redundant procedures to stop the algorithm with current R and go to next step in early stage which is an essential advantage of the AGLMC algorithm.

In comparison with the computational complexity of the algorithms, nuclear-norm based methods compute full-rank SVD in each iteration, on the other hand GFMC algorithm computes tSVD with R in each iteration. Computational complexity of tSVD is $\mathcal{O}(N^2R)$ for $R \leq \sqrt{N}$ according to [1]. Thus, GFMC is generally faster than FPC. In contrast, computational complexity of OptSpace is $\mathcal{O}(|\Omega|R\log N)$ according to [8]. In our experiment, computational times of GFMC in MATLAB and OptSpace in C were not so different.

B. Extension methods for promising practical applications

Rank estimation or model selection is a very important for practical applications such as blind source separation [7], [6] when the number of latent sources is unknown. Our greedy rank estimation scheme is very useful for such a objective, however our model may not be always appropriate for practical applications because of the orthogonal constraints for $\boldsymbol{U}$ and $\boldsymbol{V}$, and then often the extension methods for sparse, smooth, or nonnegative factor matrices are necessary. Thus it is a very interesting extension to replace the GFMC by penalized matrix decomposition (PMD)[14], nonnegative matrix factorization (NMF) [6], or other constrained matrix factorization methods.

To apply tensor data, matrix decomposition model can be extended to tensor decomposition model. For example the our AGLMC can be easily applied to PARAFAC decomposition model [5]. The estimation of the optimal number of components for many existing algorithms of PARAFAC decomposition or its constrained versions [6], [15] by using our greedy rank estimation scheme is applicable for many practical applications.

V. CONCLUSIONS

In this paper, we proposed a new method for matrix-rank estimation from noisy observations via greedy low-rank matrix completion. The proposed method minimizes the matrix rank directly without using nuclear-norm minimization. Our greedy low-rank matrix completion algorithm was able to estimate exact matrix rank, and performed faster matrix completions than Incremental OptSpace in experiments. Its extension to the constrained matrix/tensor completions are promising for real applications.

REFERENCES

[1] M. Brand. Fast low-rank modifications of the thin singular value decomposition. *Linear Algebra and its Applications*, 415(1):20–30, 2006.

[2] J. F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[3] E. J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[4] E. J. Candes and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[5] J. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319, 1970.

[6] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari. *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation.* John Wiley & Sons, 2009.

[7] A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis*, volume 46. John Wiley & Sons, 2004.

[8] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[9] R. H. Keshavan and S. Oh. A gradient descent algorithm on the Grassman manifold for matrix completion. *arXiv preprint arXiv:0910.5260*, 2009.

[10] W. Li, L. Zhao, Z. Lin, D. Xu, and D. Lu. Non-local image inpainting using low-rank matrix completion. *Computer Graphics Forum*, 2014.

[11] 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.

[12] S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1-2):321–353, 2011.

[13] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[14] D. M. Witten, R. Tibshirani, and T. Hastie. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10:515–534, 2009.

[15] T. Yokota, R. Zdunek, A. Cichocki, and Y. Yamashita. Smooth nonnegative matrix and tensor factorizations for robust multi-way data analysis. *Signal Processing*, 113:234–249, 2015.