entropy

MDPI

Article

Correntropy Based Matrix Completion

Yuning Yang [1], Yunlong Feng [2] and Johan A. K. Suykens [3,*]

[1] College of Mathematics and Information Science, Guangxi University, Nanning 530004, China; yyang@gxu.edu.cn
[2] Department of Mathematics and Statistics, The State University of New York at Albany, Albany, NY 12222, USA; ylfeng@albany.edu
[3] Department of Electrical Engineering, ESAT-STADIUS, KU Leuven, Kasteelpark Arenberg 10, Leuven B-3001, Belgium
* Correspondence: johan.suykens@esat.kuleuven.be; Tel: +32-1632-1802

Received: 24 December 2017; Accepted: 22 February 2018; Published: 6 March 2018

Abstract: This paper studies the matrix completion problems when the entries are contaminated by non-Gaussian noise or outliers. The proposed approach employs a nonconvex loss function induced by the maximum correntropy criterion. With the help of this loss function, we develop a rank constrained, as well as a nuclear norm regularized model, which is resistant to non-Gaussian noise and outliers. However, its non-convexity also leads to certain difficulties. To tackle this problem, we use the simple iterative soft and hard thresholding strategies. We show that when extending to the general affine rank minimization problems, under proper conditions, certain recoverability results can be obtained for the proposed algorithms. Numerical experiments indicate the improved performance of our proposed approach.

Keywords: robust matrix completion; hard/soft iterative thresholding; non-Gaussian noise; outliers; linear convergence

1. Introduction

Arising from a variety of applications such as online recommendation systems [1,2], image inpainting [3,4] and video denoising [5], the matrix completion problem has drawn tremendous and continuous attention over recent years [6–12]. The matrix completion aims at recovering a low rank matrix from partial observations of its entries [7]. The problem can be mathematically formulated as:

$$\min_{X\in\mathbb{R}^{m\times n}} \ \operatorname{rank}(X) \ \ \text{s.t.} \ \ X_{ij} = B_{ij}, \ \ (i,j) \in \Omega, \tag{1}$$

where $X,\ B \in \mathbb{R}^{m\times n}$ and Ω is an index set. Due to the nonconvexity of the rank function $\operatorname{rank}(\cdot)$, solving this minimization problem is NP-hard in general. To obtain a tractable convex relaxation, the nuclear norm heuristic was proposed [7]. Incorporated with the least squares loss, the nuclear norm regularization was proposed to solve (1) when the observed entries are contaminated by Gaussian noise [13–16]. In real-world applications, datasets might be contaminated by non-Gaussian noise or sparse gross errors, which can appear in both explanatory and response variables. However, it has been well understood that the least squares loss cannot be resistant to non-Gaussian noise or outliers.

To address this problem, some efforts have been made in the literature. Ref. [17] proposed a robust approach by using the least absolute deviation loss. Huber's criterion was adopted in [18] to introduce robustness into matrix completion. Ref. [19] proposed to use an L_p $(0 < p \leq 1)$ loss to enhance the robustness. However, as explained later, the approaches mentioned above cannot be robust to impulsive errors. In this study, we propose to use the correntropy-induced loss function in matrix completion problems when pursuing robustness.

Correntropy, which serves as a similarity measurement between two random variables, was proposed in [20] within the information-theoretic learning framework developed in [21]. It is shown that in prediction problems, error correntropy is closely related to the error entropy [21]. The correntropy and the induced error criterion have been drawing a great deal of attention in the signal processing and machine learning community. Given two scalar random variables U, V, the correntropy $\mathcal{V}_\sigma$ between U and V is defined as $\mathcal{V}_\sigma(U,V) = \mathbb{E}\mathcal{K}_\sigma(U,V)$ with $\mathcal{K}_\sigma$ a Gaussian kernel given by $\mathcal{K}_\sigma(u,v) = \exp\left\{-(u-v)^2/\sigma^2\right\}$, the scale parameter $\sigma > 0$ and (u,v) a realization of (U,V). It is noticed in [20] that the correntropy $\mathcal{V}_\sigma(U,V)$ can induce a new metric between U and V.

In this study, by employing the correntropy-induced losses, we propose a nonconvex relaxation approach to robust matrix completion. Specifically, we develop two models: one with a rank constraint and the other with a nuclear norm regularization term. To solve them, we propose to use simple, but efficient algorithms. Experiments on synthetic, as well as real data are implemented and show that our methods are effective even for heavily-contaminated datasets. We make the following contributions in this paper:

- In Section 3, we propose a nonconvex relaxation strategy for the robust matrix completion problem, where the robustness benefits from using a robust loss. Based on this loss, a rank constraint, as well as a nuclear norm penalized model is proposed. We also extend the proposed models to deal with the affine rank minimization problem, which includes the matrix completion as a special case.
- In Section 4, we propose to use simple, but effective algorithms to solve the proposed models, which are based on gradient descent and employ the hard/soft shrinkage operators. By verifying the Lipschitz continuity, the convergence of the algorithms can be proven. When extended to affine rank minimization problems, under proper conditions, certain recoverability results are obtained. These results give understandings of this loss function in an algorithmic sense, which is in accordance with and extends our previous work [22].

This paper is organized as follows: In Section 2, we review some existing (robust) matrix completion approaches. In Section 3, we propose our nonconvex relaxation approach. Two algorithms are proposed in Section 4 to solve the proposed models. Theoretical results will be presented in Section 4.1. Experimental results are reported in Section 5. We end this paper in Section 6 with concluding remarks.

2. Related Work and Discussions

In matrix completion, solving the optimization problem in Model (1) is NP-hard, and a usual remedy is to consider the following nuclear norm convex relaxation:

$$\min_{X\in\mathbb{R}^{m\times n}} \|X\|_* \quad \text{s.t. } X_{i,j} = B_{i,j}, \ (i,j) \in \Omega. \tag{2}$$

Theoretically, it has been demonstrated in [7,8] that under proper assumptions, with an overwhelming probability, one can reconstruct the original matrix. Situations of the matrix completion with noisy entries have been also considered; see, e.g., [6,9]. In the noisy setting, the corresponding observed matrix turns out to be:

$$B_\Omega = X_\Omega + E, \tag{3}$$

where B_Ω denotes the projection of B onto Ω, and E refers to the noise. The following two models are frequently adopted to deal with the noisy case:

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|X_\Omega - B_\Omega\|_F^2 \quad \text{s.t. } \operatorname{rank}(X) \le R,$$

and its convex relaxed and regularized heuristic:

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|X_\Omega - B_\Omega\|_F^2 + \lambda\|X\|_*,$$

where $\lambda > 0$ is a regularization parameter. Similar theoretical reconstruction results have been also derived in the noiseless case under technical assumptions. Along this line, various approaches have been proposed [14–16,23,24]. Among others, Refs. [10,25] interpreted the matrix completion problem as a specific case of the trace regression problem endowed with an entry-wise least squares loss, $\|\cdot\|_F^2$. In the above-mentioned settings, the noise term E is usually assumed to be Gaussian or sub-Gaussian to ensure the good generalization ability, which certainly excludes the heavily-tailed noise and/or outliers.

Existing Robust Matrix Completion Approaches

It has been well understood that the least squares estimator cannot deal with non-Gaussian noise or outliers. To alleviate this limitation, some efforts have been made.

In a seminal work, Ref. [17] proposed a robust matrix completion approach, in which the model takes the following form:

$$\min_{X,E\in\mathbb{R}^{m\times n}} \|E\|_1 + \lambda\|X\|_* \quad \text{s.t. } X_\Omega + E = B_\Omega. \tag{4}$$

The above model can be further formulated as:

$$\min_{X\in\mathbb{R}^{m\times n}} \|X_\Omega - B_\Omega\|_1 + \lambda\|X\|_*,$$

where $\lambda > 0$ is a regularization parameter. The robustness of the model (4) results from using the least absolute deviation loss (LAD). This model was later applied to the column-wise robust matrix completion problem in [26].

By further decomposing E into $E = E_1 + E_2$, where E_1 refers to the noise and E_2 stands for the outliers, Ref. [18] proposed the following robust reconstruction model:

$$\min_{X,E_2\in\mathbb{R}^{m\times n}} \|X_\Omega - B_\Omega - E_2\|_F^2 + \lambda\|X\|_* + \gamma\|E_2\|_1,$$

where $\lambda, \gamma > 0$ are regularization parameters. They further showed that the above estimator is equivalent to the one obtained by using Huber's criterion when evaluating the data-fitting risk. We also note that [19] adopted an L_p $(0 < p \leq 1)$ loss to enhance the robustness.

3. The Proposed Approach

3.1. Our Proposed Nonconvex Relaxation Approach

As stated previously, matrix completion models based on the least squares loss cannot perform well with non-Gaussian noise and/or outliers. Accordingly, robustness can be pursued by using a robust loss as mentioned earlier. Associated with a nuclear norm penalization term, they are essentially regularized M-estimator. However, note that the LAD loss and the L_p loss penalize the small residuals strongly and hence cannot lead to accurate prediction for unobserved entries from the trace regression viewpoint. Moreover, robust statistics reminds us that models based on the above three mentioned loss functions cannot be robust to impulsive errors [27,28]. These limitations encourage us to employ more robust surrogate loss functions to address this problem. In this paper, we present a nonconvex relaxation approach to deal with the matrix completion problem with entries heavily contaminated by noise and/or outliers.

In our study, we propose the robust matrix completion model based on a robust and nonconvex loss, which is defined by:

$$\rho_\sigma(t) = \sigma^2(1 - \exp(-t^2/\sigma^2)),$$

with $\sigma > 0$ a scale parameter. To give an intuitive impression, plots of loss functions mentioned above are given in Figure 1. As mentioned above, this loss function is induced by the correntropy, which measures the similarity between two random variables [20,21] and has found many successful applications [29–31]. Recently, it was shown in [22] that regression with the correntropy-induced losses regresses towards the conditional mean function with a diverging scale parameter σ when the sample size goes to infinity. It was also shown in [32] that when the noise variable admits a unique global mode, regression with the correntropy-induced losses regresses towards the conditional mode. As argued in [22,32], learning with correntropy-induced losses can be resistant to non-Gaussian noise and outliers, while ensuring good prediction accuracy simultaneously with properly chosen σ.

Associated with the ρ_σ loss, our rank-constraint robust matrix completion problem is formulated as:

$$\min_{X \in \mathbb{R}^{m \times n}} \quad \ell_\sigma(X) \quad \text{s.t.} \quad \text{rank}(X) \leq R, \tag{5}$$

where the data-fitting risk $\ell_\sigma(X)$ is given by:

$$\ell_\sigma(X) = \frac{1}{2} \sum_{(i,j) \in \Omega} \rho_\sigma \left(X_{ij} - B_{ij} \right) = \frac{\sigma^2}{2} \sum_{(i,j) \in \Omega} \left(1 - \exp\left(-(X_{ij} - B_{ij})^2 / \sigma^2 \right) \right).$$

The nuclear norm heuristic model takes the following form:

$$\min_{X \in \mathbb{R}^{m \times n}} \ \ell_\sigma(X) + \lambda \|X\|_*, \tag{6}$$

where $\lambda > 0$ is a regularization parameter.

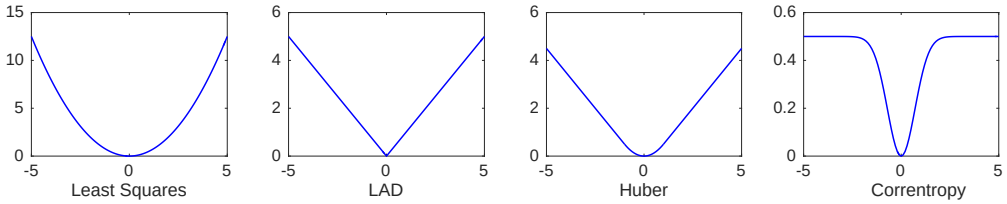

Figure 1. Different losses: least squares, absolute deviation loss (LAD), Huber's loss and ρ_σ (Welsch loss).

3.2. Affine Rank Minimization Problem

In this part, we will show that our robust matrix completion approach can be extended to deal with the robust affine rank minimization problems.

It is known that the matrix completion problem (1) is a special case of the following affine rank minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \quad \text{rank}(X) \quad \text{s.t.} \quad \mathcal{A}(X) = b, \tag{7}$$

where $b \in \mathbb{R}^p$ is given, and $\mathcal{A} : \mathbb{R}^{m \times n} \to \mathbb{R}^p$ is a linear operator defined by:

$$\mathcal{A}(\cdot) := \left[\langle A^1, \cdot \rangle, \ \langle A^2, \cdot \rangle, \ldots, \ \langle A^p, \cdot \rangle \right]^T,$$

where $A^i \in \mathbb{R}^{m \times n}$ for each i. Introduced and studied in [33], this problem has drawn much attention in recent years [14–16,23]. Note that (7) can be reduced to the matrix completion problem (1) if

we set $p = |\Omega|$ (the cardinality of Ω), and let $A^{(i-1)n+j} = e_i(m)e_j(n)^T$ for each $(i,j) \in \Omega$, where $e_i(m), i = 1,\ldots,m$ and $e_j(n), j = 1,\ldots,n$ are the canonical basis vector of $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively.

In fact, (5) and (6) can be naturally extended to handle cases with noise and outliers of (7). Denote the risk as follows:

$$\tilde{\ell}_\sigma(X) = \frac{\sigma^2}{2}\sum_{i=1}^{p}\left(1-\exp\left(-\left(\left\langle A^i, X\right\rangle - b_i\right)^2/\sigma^2\right)\right).$$

The rank constrained model can be formulated as:

$$\min_{X\in\mathbb{R}^{m\times n}} \tilde{\ell}_\sigma(X) \text{ s.t. } \operatorname{rank}(X) \le R, \tag{8}$$

and the nuclear norm regularized heuristic takes the form:

$$\min_{X\in\mathbb{R}^{m\times n}} \tilde{\ell}_\sigma(X) + \lambda\|X\|_*. \tag{9}$$

Referring to computational considerations presented below, we will focus on the more general optimization problems (8) and (9), which can be directly applied to (5) and (6).

4. Algorithms and Analysis

We consider using gradient descent-based algorithms to solve the proposed models. It is usually admitted that gradient descent is not very efficient. However, in our experiments, we find that gradient descent is still efficient, and comparable with some state-of-the-art methods. On the other hand, we present recoverability and convergence rate results for gradient descent applied to the proposed models. Such results and analysis may help us better understand the models and such a nonconvex loss function from the algorithmic aspects.

We first consider gradient descent with hard thresholding for solving (8). The derivation is standard. Denote $S_R := \{X \in \mathbb{R}^{m\times n} \mid \operatorname{rank}(X) \le R\}$. By the differentiability of ℓ_σ, when Y is sufficiently close to X, ℓ_σ can be approximated by:

$$\ell_\sigma(X) \approx \ell_\sigma(Y) + \langle\nabla\ell_\sigma(Y), X-Y\rangle + \frac{\alpha}{2}\|X-Y\|_F^2.$$

Here, $\alpha > 0$ is a parameter, and $\nabla\ell_\sigma(Y)$, the gradient of ℓ_σ at Y, is equal to:

$$\sum_{i=1}^{p}\exp\left(-(\langle A^i, Y\rangle - b_i)^2/\sigma^2\right)(\langle A^i, Y\rangle - b_i)A^i. \tag{10}$$

Now, the iterates can be generated as follows:

$$\begin{aligned} X^{(k+1)} &= \arg\min_{X\in S_R} \ell_\sigma(X^{(k)}) + \left\langle\nabla\ell_\sigma(X^{(k)}), X-X^{(k)}\right\rangle \\ &\quad + \frac{\alpha}{2}\|X-X^{(k)}\|_F^2 \\ &= \arg\min_{X\in S_R}\|X-Y^{(k+1)}\|_F^2 \end{aligned} \tag{11}$$

with:

$$Y^{(k+1)} = X^{(k)} - \alpha^{-1}\nabla\ell_\sigma(X^{(k)}). \tag{12}$$

We simply write (11) as $X^{(k+1)} = \mathcal{P}_{S_R}(Y^{(k+1)})$, where $\mathcal{P}_{S_R}$ denotes the hard thresholding operator, i.e., the best rank-R approximation to $Y^{(k+1)}$. The algorithm is presented in Algorithm 1.

Algorithm 1 Gradient descent iterative hard thresholding for (8).

Input: linear operator $\mathcal{A}: \mathbb{R}^{m\times n} \to \mathbb{R}^p$, initial guess $X^{(0)} \in \mathbb{R}^{m\times n}$, prescribed rank $R \geq 1$, $\sigma > 0$
Output: the recovered matrix $X^{(k+1)}$
while a certain stopping criterion is not satisfied **do**
1: Choose a fixed step-size $\alpha^{-1} > 0$.
2: Compute the gradient descent step (12)

$$Y^{(k+1)} = X^{(k)} - \alpha^{-1}\nabla \ell_\sigma(X^{(k)}).$$

3: Perform the hard thresholding operator to obtain

$$X^{(k+1)} = \mathcal{P}_{S_R}\left(Y^{(k+1)}\right),$$

and set $k := k+1$.
end while

The algorithm starts from an initial guess $X^{(0)}$ and continues until some stopping criterion is satisfied, e.g., $\|X^{(k+1)} - X^{(k)}\|_F \leq \epsilon$, where ϵ is a certain given positive number. Indeed, such a stopping criterion makes sense, as Proposition A3 shows that $\|X^{(k)} - X^{(k+1)}\|_F \to 0$. To ensure the convergence, the step-size should satisfy $\alpha > L := \|\mathcal{A}\|_2^2$, where $\|\mathcal{A}\|_2$ denotes the spectral norm of $\mathcal{A}$. For matrix completion, the spectral norm is smaller than one, and thus, we can set $\alpha > 1$. In Appendix A, we have shown the Lipschitz continuity of $\nabla \ell_\sigma(\cdot)$, which is necessary for the convergence of the algorithm. α can also be self-adaptive by using a certain line-search rule. Algorithm 2 is the line-search version of Algorithm 1.

Algorithm 2 Line-search version of Algorithm 1.

Input: linear operator $\mathcal{A}: \mathbb{R}^{m\times n} \to \mathbb{R}^p$, initial guess $X^{(0)} \in \mathbb{R}^{m\times n}$, prescribed rank $R \geq 1$, $\sigma > 0$, $\alpha^{(0)} > 0$, $\delta \in (0,1)$, $\eta > 1$
Output: the recovered matrix $X^{(k+1)}$
while a certain stopping criterion is not satisfied **do**
1: $\alpha^{(k+1)} = \alpha^{(k)}$
repeat
2: $X^{(k+1)} = \mathcal{P}_{S_R}\left(X^{(k)} - \frac{1}{\alpha^{(k+1)}}\nabla \ell_\sigma(X^{(k)})\right)$
3: $\alpha^{(k+1)} := \alpha^{(k+1)}\eta$
until $\ell_\sigma(X^{(k+1)}) \leq \ell_\sigma(X^{(k)}) - \frac{\delta\alpha^{(k+1)}}{2}\|X^{(k+1)} - X^{(k)}\|_F^2$
4: $\alpha^{(k+1)} := \alpha^{(k+1)}/\eta$,
and set $k := k+1$.
end while

Solving (9) is similar, with only the hard thresholding $\mathcal{P}_R$ replaced by the soft thresholding $\mathcal{S}_\tau$, which can be derived as follows. Denote $Y^{(k+1)} = U\mathrm{diag}\left(\{\sigma_i\}_{1\leq i\leq r}\right)V^T$ as the SVD of $Y^{(k+1)}$. Then, $S_{\lambda/\alpha}$ is the matrix soft thresholding operator [13,16] defined as $S_{\lambda/\alpha}(Y^{(k+1)}) = U\mathrm{diag}\left(\max\{\sigma_i - \lambda/\alpha, 0\}\right)V^T$. Gradient descent-based soft thresholding is summarized in Algorithm 3.

Algorithm 3 Gradient descent iterative soft thresholding for (9).

Input: linear operator $\mathcal{A}: \mathbb{R}^{m\times n} \to \mathbb{R}^p$, initial guess $X^{(0)} \in \mathbb{R}^{m\times n}$, parameter $\lambda > 0$, $\sigma > 0$
Output: the recovered matrix $X^{(k+1)}$
while a certain stopping criterion is not satisfied **do**
1: Choose a fixed step-size $\alpha^{-1} > 0$, or choose it via the line-search rule.
2: Compute

$$Y^{(k+1)} = X^{(k)} - \alpha^{-1}\nabla \ell_\sigma(X^{(k)}).$$

3: Perform the soft thresholding operator to obtain

$$X^{(k+1)} = S_{\lambda/\alpha}(Y^{(k+1)}),$$

and set $k := k+1$.
end while

4.1. Convergence

With the Lipschitz continuity of $\nabla \ell_\sigma$ presented in Appendix A, it is a standard routine to show the convergence of Algorithms 1 and 3, i.e., let $\{X^{(k)}\}$ be a sequence generated by Algorithm 1 or 3. Then, every limit point of the sequence is a critical point of the problem. In fact, the results can be enhanced to the statement that "the entire sequence converges to a critical point", namely one can prove that $\lim_{k\to\infty} X^{(k)} = X^*$ where X^* is a critical point. This can be achieved by verifying the so-called Kurdyka–Łojasiewicz (KL) property [34] of the problems (8) and (9). As this is not the main concern of this paper, we omit the verification here.

4.2. Recoverability and Linear Convergence Rate

For affine rank minimization problems, the convergence rate results have been obtained in the literature; see, e.g., [23,24]. However, all the existing results are obtained for algorithms that solve the optimization problems incorporating the least squares loss. In this part, we are concerned with the recoverability and convergence rate of Algorithm 1. These results give the understanding of this loss function from the algorithmic aspect, which is in accordance with and extends our previous work [22].

It has been known that the convergence rate analysis requires the matrix RIPcondition [33]. In our context, instead of using the matrix RIP, we adopt the concept of the matrix scalable restricted isometry property (SRIP) [24].

Definition 1 (SRIP [24]). *For any $X \in S_r$, there exist constants $\nu_r, \mu_r > 0$ such that:*

$$\nu_r \|X\|_F \le \|\mathcal{A}(X)\|_F \le \mu_r \|X\|_F.$$

Due to the scalability of ν_r, μ_r on the operator $\mathcal{A}$, SRIP is a generalization of the RIP [33] as commented in [24]. We point out that the results of Algorithm 1 for the affine rank minimization problem (8) rely on the SRIP condition. However, in the matrix completion problem (5), this condition cannot be met, since ν_r in this case is zero. Consequently, the results provided below cannot be applied directly to the matrix completion problem (5). However, similar results might be established for (5), if some refined RIP conditions are assumed to hold for the operator $\mathcal{A}$ in the situation of matrix completion [23]. To obtain the convergence rate results, besides the SRIP condition, we also need to make some assumptions.

Assumption 1.

1. *At the $(k+1)$-th iteration of Algorithm 1, the parameter σ^{k+1} in the loss function ℓ_σ is chosen as:*

$$\sigma^{k+1} = \max\left\{ \frac{\|\mathcal{A}(X^{(k)}) - b\|_F}{\sqrt{2(1-\beta)}}, \hat{\sigma} \right\},$$

   where $\beta \in [0.988, 1)$, and $\hat{\sigma}$ is a positive constant.
2. *The spectral norm of A is upper bounded as $\|A\|_2^2 \le \frac{6}{5}\nu_{2R}^2$.*

Based on Assumption 1, the following results for Algorithm 1 can be derived.

Theorem 1. *Assume that $\mathcal{A}(X^*) + \epsilon = b$, where X^* is the matrix to be recovered and $\operatorname{rank}(X^*) = R$. Assume that Assumption 1 holds. Let $\{X^{(k)}\}$ be generated by Algorithm 1, with the step-size $\alpha = \|A\|_2^2$. Then*

1. *at iteration $(k+1)$, Algorithm 1 will recover a matrix X^{k+1} satisfying:*

$$\|X^{(k+1)} - X^*\|_F \le q_1^{k+1} \|X^{(0)} - X^*\|_F + \frac{2}{1-q_1} \frac{\|\epsilon\|_F}{\|A\|_2},$$

   where $q_1 \in (0.8165, 0.9082)$ depending on β.

2. *If there is no noise or outliers, i.e., $\mathcal{A}(X^*) = b$, then the algorithm converges linearly in the least squares and ℓ_σ sense, respectively, i.e.,*

$$\begin{aligned} \|\mathcal{A}(X^{(k+1)}) - b\|^2 &\leq q_2 \|\mathcal{A}(X^{(k)}) - b\|^2, \text{ and} \\ \tilde{\ell}_{\sigma^{k+1}}(X^{(k+1)}) &\leq q_3\, \tilde{\ell}_{\sigma^k}(X^{(k)}), \end{aligned}$$

where $q_2 \in (0.8, 0.9898)$ and $q_3 \in (0.2, 0.262)$, depending on the choice of β.

The proof of Theorem 1 relies on the following lemmas, which reveal certain properties of the loss function $\tilde{\ell}_\sigma$.

Lemma 1. *For any $\sigma > 0$ and $t \in \mathbb{R}$, it holds:*

$$\frac{\sigma^2}{2}\left(1 - \exp\left(\frac{-t^2}{\sigma^2}\right)\right) \leq \frac{t^2}{2}.$$

Proof. For any $\sigma > 0$, let $f(t) := \frac{t^2}{2} - \frac{\sigma^2}{2}(1 - \exp(\frac{-t^2}{\sigma^2}))$. Since $f(t)$ is even, we need to only consider $t \geq 0$. Note that $f^{'}(t) = t - t\exp(\frac{-t^2}{\sigma^2})$, which is nonnegative when $t \geq 0$. Therefore, $f(t)$ is a nondecreasing function on $[0, +\infty)$. On the other hand, $f^{'}(0) = 0$ and $f(t) = 0$. Thus, the minimum of $f(t)$ is $f(0) = 0$. As a result, $f(t) \geq 0$. This completes the proof. □

Lemma 2. *Assuming that $\beta \in [0, 1)$, and $0 < \delta \leq 2(1 - \beta)$, it holds:*

$$g(\delta) := 1 - \exp(-\delta) - \beta\delta \geq 0.$$

Proof. Since $\delta > 0$, it is not hard to check that $1 - \exp(-\delta) \geq \delta - \frac{1}{2}\delta^2$. From the range of δ, it follows $\delta - \frac{1}{2}\delta^2 \geq \beta\delta$. This completes the proof. □

Lemma 3. *Given a fixed $t \in \mathbb{R}$, for $\sigma > 0$, $h(\sigma) := \sigma^2(1 - \exp(-t^2/\sigma^2))$ is nondecreasing with respect to σ.*

Proof. It is not hard to check that h' is nonnegative on $\sigma > 0$. □

Proof of Theorem 1. By the fact that X^* is rank-R and $X^{(k+1)}$ is the best rank-R approximation to $Y^{(k+1)}$, we have:

$$\begin{aligned} & \|X^{(k+1)} - X^*\|_F \\ \leq\ & \|X^{(k+1)} - Y^{(k+1)}\|_F + \|Y^{(k+1)} - X^*\|_F \\ \leq\ & 2\|Y^{(k+1)} - X^*\|_F \\ =\ & 2\|X^{(k)} - X^* - \frac{1}{\alpha}\nabla \ell_{\sigma^{k+1}}(X^{(k)})\|_F. \end{aligned}$$

Since:

$$\begin{aligned} \text{vec}\left(\nabla \ell_{\sigma^{k+1}}(X^{(k)})\right) &= A^T\Lambda\left(A\text{vec}(X^{(k)}) - b\right) \\ &= A^T\Lambda\left(A\text{vec}(X^{(k)} - X^*) - \epsilon\right), \end{aligned}$$

we know that:

$$\begin{aligned}
& \left\| X^{(k)} - X^* - \frac{1}{\alpha}\nabla \ell_{\sigma^{k+1}}(X^{(k)}) \right\|_F \\
= & \left\| \mathrm{vec}(X^{(k)} - X^*) - \frac{1}{\alpha} A^T \Lambda \left(A\mathrm{vec}(X^{(k)} - X^*) - \epsilon \right) \right\|_F \\
\leq & \left\| \mathrm{vec}(X^{(k)} - X^*) - \frac{1}{\alpha} A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\|_F \\
& + \frac{1}{\alpha} \left\| A^T \Lambda \epsilon \right\|_F \\
\leq & \left\| \mathrm{vec}(X^{(k)} - X^*) - \frac{1}{\alpha} A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\|_F \\
& + \frac{\|\epsilon\|_F}{\|A\|_2},
\end{aligned}$$

where the last inequality follows from:

$$\|A^T \Lambda \epsilon\|_F \leq \|A\|_2 \|\Lambda\|_2 \|\epsilon\|_F \leq \|A\|_2 \|\epsilon\|_F$$

and the choice of the step-size α. It remains to estimate $\|\mathrm{vec}(X^{(k)} - X^*) - \frac{1}{\alpha} A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*)\|_F$. We first see that:

$$\begin{aligned}
& \left\| \mathrm{vec}(X^{(k)} - X^*) - \frac{1}{\alpha} A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\|_F^2 \\
= & -\frac{2}{\alpha} \left\langle \mathrm{vec}(X^{(k)} - X^*), A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\rangle \\
& + \frac{1}{\alpha^2} \left\| A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\|_F^2 + \left\| X^{(k)} - X^* \right\|_F^2
\end{aligned} \tag{13}$$

To verify our first assertion, it remains to bound the first two terms by means of $\|X^{(k)} - X^*\|_F^2$. We consider the first term. Denoting $y_i^k = \langle A^i, X^{(k)} - X^* \rangle$, we know that:

$$\begin{aligned}
& \left\langle \mathrm{vec}(X^{(k)} - X^*), A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\rangle \\
= & \left\langle A\mathrm{vec}(X^{(k)} - X^*), \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\rangle \\
= & \sum_{i=1}^{p} \exp\left(- \left(\frac{\langle A^i, X^{(k)} \rangle - b_i}{\sigma^{k+1}} \right)^2 \right) \left(y_i^k \right)^2 .
\end{aligned}$$

The choice of σ^{k+1} tells us that:

$$\exp\left(- \left(\frac{\langle A^i, X^{(k)} \rangle - b_i}{\sigma^{k+1}} \right)^2 \right) \geq \exp\left(-2(1-\beta) \right),$$

and consequently:

$$\begin{aligned}
& -\frac{2}{\alpha} \left\langle \mathrm{vec}(X^{(k)} - X^*), A^T \Lambda A \mathrm{vec}(X^{(k)} - X^*) \right\rangle \\
& \leq -\frac{2}{\alpha} \exp\left(-2(1-\beta) \right) \|A\mathrm{vec}(X^{(k)} - X^*)\|_F^2 .
\end{aligned} \tag{14}$$

Then, by the fact that $\|\Lambda\|_2^2 \leq 1$ and the choice of the step-size α, we observe that the second term of (13) can be upper bounded by:

$$\frac{1}{\alpha^2}\|A^T\Lambda A\text{vec}(X^{(k)} - X^*)\|_F^2 \leq \frac{1}{\alpha}\|A\text{vec}(X^{(k)} - X^*)\|_F^2. \tag{15}$$

Combining (14) and (15) and denoting $\gamma = 1 - 2\exp(-2(1-\beta))$, we come to the following conclusion:

$$\begin{aligned} & \left\|\text{vec}(X^{(k)} - X^*) - \frac{1}{\alpha}A^T\Lambda A\text{vec}(X^{(k)} - X^*)\right\|_F^2 \\ \leq & \left\|X^{(k)} - X^*\right\|_F^2 + \frac{\gamma}{\alpha}\left\|A\text{vec}(X^{(k)} - X^*)\right\|_F^2 \\ \leq & \left\|X^{(k)} - X^*\right\|_F^2 + \frac{\gamma\nu_{2R}^2}{\alpha}\left\|X^{(k)} - X^*\right\|_F^2, \end{aligned}$$

where the last inequality follows from the SRIP condition and the fact that $\gamma < 0$ by the range of β. As a result, we get the following estimation:

$$\begin{aligned} \|X^{(k+1)} - X^*\|_F &\leq 2\|X^{(k)} - X^* - \frac{1}{\alpha}\nabla\ell_{\sigma^{k+1}}(X^{(k)})\|_F \\ &\leq 2\sqrt{1 + \frac{\gamma\nu_{2R}^2}{\alpha}}\|X^{(k)} - X^*\|_F + 2\frac{\|\epsilon\|_F}{\|A\|_2} \\ &\leq 2\sqrt{1 + \frac{5\gamma}{6}}\|X^{(k)} - X^*\|_F + 2\frac{\|\epsilon\|_F}{\|A\|_2} \end{aligned} \tag{16}$$

where the last inequality follows from the assumption $\alpha = \|A\|_2^2 \leq 6/5\nu_{2R}^2$. Denote $q_1 = 2\sqrt{1 + \frac{5\gamma}{6}}$. The range of β tells us that $q_1 \in (0.8165, 0.9082)$. Iterating (16), we obtain:

$$\|X^{(k+1)} - X^*\|_F \leq q_1^{k+1}\|X^{(0)} - X^*\|_F + \frac{2}{1 - q_1}\frac{\|\epsilon\|_F}{\|A\|_2}.$$

Therefore, The first assertion concerning the recoverability is proven.

Suppose there is no noise or outliers, i.e., we have $\mathcal{A}(X^*) = b$. In this case, it follows from (16) that:

$$\|X^{(k+1)} - X^*\|_F \leq q_1\|X^{(k)} - X^*\|_F,$$

and then, the SRIP condition tells us that:

$$\begin{aligned} \|\mathcal{A}(X^k) - b\|_F^2 &\leq \mu_{2R}^2\|X^{k+1} - X^*\|_F^2 \\ &\leq \mu_{2R}^2 q_1^2\|X^{(k)} - X^*\|_F^2 \\ &\leq \left(\frac{\mu_{2R}}{\nu_{2R}}\right)^2 q_1^2\|\mathcal{A}(X^k) - b\|_F^2 \\ &\leq \frac{6}{5}q_1^2\|\mathcal{A}(X^k) - b\|_F^2, \end{aligned}$$

where the last inequality comes from the inequality chain $\mu_{2R}^2 \leq \|A\|_2^2 \leq 6/5\nu_{2R}^2$. Denote $q_2 = 6q_1^2/5$. Then, $q_2 \in (0.8, 0.9898)$. Therefore, the algorithm converges linearly to X^* in the least squares sense.

We now proceed to show the linear convergence in the $\tilde{\ell}_\sigma$ sense. Following from the inequality $\|X^{(k+1)} - Y^{(k+1)}\|_F^2 \leq \|X^* - Y^{(k+1)}\|_F^2$, we obtain:

$$\begin{aligned} & \frac{\alpha}{2}\|X^{(k+1)} - X^{(k)}\|_F^2 \\ & + \left\langle \nabla \ell_{\sigma^{k+1}}(X^{(k)}), X^{(k+1)} - X^{(k)} \right\rangle \\ \leq \; & \frac{\alpha}{2}\|X^{(k)} - X^*\|_F^2 + \left\langle \nabla \ell_{\sigma^{k+1}}(X^{(k)}), X^* - X^{(k)} \right\rangle. \end{aligned}$$

Combining with Inequality (A1), we see that $\tilde{\ell}_{\sigma^{k+1}}(X^{(k+1)})$ can be upper bounded by:

$$\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}) + \frac{\alpha}{2}\|X^{(k)} - X^*\|_F^2 + \left\langle \nabla \tilde{\ell}_{\sigma^{k+1}}(X^{(k)}), X^* - X^{(k)} \right\rangle. \tag{17}$$

We need to upper bound $\left\langle \nabla \tilde{\ell}_{\sigma^{k+1}}(X^{(k)}), X^* - X^{(k)} \right\rangle$ and $\frac{\alpha}{2}\|X^{(k)} - X^*\|_F^2$ in terms of $\tilde{\ell}_{\sigma^{k+1}}(X^{(k)})$. We first consider the second term. Under the SRIP condition, we have:

$$\begin{aligned} \|X^{(k)} - X^*\|_F^2 & \leq \frac{1}{\nu_{2R}^2}\|\mathcal{A}(X^{(k)} - X^*)\|_F^2 \\ & = \frac{1}{\nu_{2R}^2}\|\mathcal{A}(X^{(k)}) - b\|_F^2. \end{aligned}$$

By setting $\delta = \left(\frac{y_i^k}{\sigma^{k+1}}\right)^2$, we get $\delta \leq 2(1-\beta)$. Lemma 2 tells us that:

$$\beta (y_i^k)^2 \leq (\sigma^{k+1})^2 \left(1 - \exp\left(-(y_i^k/\sigma^{k+1})^2\right)\right).$$

Summing the above inequalities over i from 1 to p, we have:

$$\begin{aligned} & \beta\|\mathcal{A}(X^{(k)}) - b\|_F^2 \\ \leq \; & (\sigma^{k+1})^2 \sum_{i=1}^{p} \left(1 - \exp\left(-(y_i^k/\sigma^{k+1})^2\right)\right) \\ = \; & 2\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}). \end{aligned}$$

Therefore, $\frac{\alpha}{2}\|X^{(k)} - X^*\|_F^2$ can be bounded as follows:

$$\begin{aligned} \tfrac{\alpha}{2}\|X^{(k)} - X^*\|_F^2 & \leq \tfrac{\alpha}{2\nu_{2R}^2}\|\mathcal{A}(X^{(k)} - X^*)\|^2 \\ & \leq \tfrac{\alpha}{\beta\nu_{2R}^2}\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}). \end{aligned} \tag{18}$$

We proceed to bound $\left\langle \nabla \tilde{\ell}_{\sigma^{k+1}}(X^{(k)}), X^* - X^{(k)} \right\rangle$. It follows from (14) and Lemma 1 that:

$$\begin{aligned} & \left\langle \nabla \tilde{\ell}_{\sigma^{k+1}}(X^{(k)}), X^* - X^{(k)} \right\rangle \\ \leq \; & -\exp(-2(1-\beta))\|\mathcal{A}(X^{(k)}) - b\|^2 \\ \leq \; & -2\exp(-2(1-\beta))\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}). \end{aligned} \tag{19}$$

Combining (17)–(19) together, we get:

$$\begin{aligned} & \tilde{\ell}_{\sigma^{k+1}}(X^{(k+1)}) \\ \leq & \left(1+\frac{\alpha}{\beta\nu_{2R}^2}-2\exp(-2(1-\beta))\right)\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}) \\ \leq & \left(1+\frac{6}{5\beta}-2\exp(-2(1-\beta))\right)\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}), \end{aligned}$$

where the last inequality follows from $\alpha \leq \frac{6}{5}\nu_{2R}^2$.

By Lemma 3, the function $\sigma^2(1-\exp(-t^2/\sigma^2))$ is nondecreasing with respect to $\sigma > 0$. This in connection with the fact that:

$$\begin{aligned} \sigma^{k+1} &= \max\left\{\frac{\|\mathcal{A}(X^{(k)})-b\|_F}{\sqrt{2(1-\beta)}},\sigma\right\} \\ \leq \quad \sigma^{k} &= \max\left\{\frac{\|\mathcal{A}(X^{(k-1)})-b\|_F}{\sqrt{2(1-\beta)}},\sigma\right\} \end{aligned}$$

yields $\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}) \leq \tilde{\ell}_{\sigma^{k}}(X^{(k)})$. Let $q_3 = 1+\frac{6}{5\beta}-2\exp(-2(1-\beta))$, and consequently, $q_3 \in (0.2, 0.2620)$. We thus have:

$$\tilde{\ell}_{\sigma^{k+1}}(X^{(k+1)}) \leq q_3\tilde{\ell}_{\sigma^{k+1}}(X^{(k)}) \leq q_3\,\tilde{\ell}_{\sigma^{k}}(X^{(k)}).$$

The proof is now completed. □

The above results show that it is possible that Algorithm 1 will find $\mathcal{X}^*$ if the magnitude of the noise is not too large. Moreover, the results also imply that the algorithm is safe when there is no noise.

5. Numerical Experiments

This section presents numerical experiments to illustrate the effectiveness of our methods. Empirical comparisons with other methods are implemented on synthetic and real data contaminated by outliers or non-Gaussian noise.

The following 4 algorithms are implemented. RMC-ℓ_σ-IHTand RMC-ℓ_σ-ISTare denoted as Algorithms 1 and 3 incorporated with the line-search rule, respectively. The approach proposed in [16] is denoted as MC-ℓ_2-IST, which is an iterative soft thresholding algorithm based on the least squares loss. The robust approach based on the LAD loss proposed in [17] is denoted by RMC-ℓ_1-ADM. Empirically, the σ value of ℓ_σ is set to be 0.5; the tuned parameter λ of RMC-ℓ_σ-IST and MC-ℓ_2-IST is set to $\lambda = \frac{\min\{m,n\}}{10\sqrt{\max\{m,n\}}}$, while for RMC-$\ell_1$-ADM, $\lambda = 1/\sqrt{\max\{m,n\}}$, as suggested in [17]. All the numerical computations are conducted on an Intel i7-3770 CPU desktop computer with 16 GB of RAM. The supporting software is MATLAB R2013a. Some notations used frequently in this section are introduced first in Table 1. Bold number in the tables of this section means that it is the best among the competitors.

Table 1. Notations used in the experiments.

Notations	Descriptions
ρ_r	the ratio of the rank to the dimensionality of a matrix
ρ_o	the ratio of outliers to the number of entries of a matrix
ρ_m	the level of missing entries
s_n	the factor of scale of noise

5.1. Evaluation on Synthetic Data

The synthetic datasets are generated in the following way:

1. Generating a low rank matrix: We first generate an $m \times n$ matrix with i.i.d. Gaussian entries $\sim N(0,1)$, where $m = n = 1000$. Then, a $\lfloor \rho_r m \rfloor$-rank matrix M is obtained from the above matrix by rank truncation, where ρ_r varies from 0.04–0.4.
2. Adding outliers: We create a zero matrix $E \in \mathbb{R}^{m \times n}$ and uniformly randomly sample $\rho_o m^2$ entries, where ρ_o varies from 0–0.6. These entries are randomly drawn from the chi-square distribution, with four degrees of freedom. Multiplied by 10, the matrix E is used as the sparse error matrix.
3. Missing entries: $\rho_m m^2$ of the entries are randomly missing, with ρ_m varying between $\{0, 10\%, 20\%, 30\%\}$. Finally, the observed matrix is denoted as $B = P_\Omega(M + E)$.

RMC-ℓ_σ-IHT (Algorithm 1), RMC-ℓ_σ-IST (Algorithm 3) and RMC-ℓ_1-ADM [17] are implemented respectively on the matrix completion problem with the datasets generated above. For these three algorithms, the same initial guess with the all-zero matrix $X^0 = 0$ is applied. The stopping criterion is $\|X^{(k+1)} - X^{(k)}\|_F \leq 10^{-3}$, or restrictions on the number of iterations, which is set to be 500. For each tuple (ρ_m, ρ_r, ρ_o), we repeat 10 runs. The algorithm is regarded as successful if the relative error of the result $\hat{X}$ satisfies $\|\hat{X} - M\|_F / \|M\|_F \leq 10^{-1}$.

Experimental results of RMC-ℓ_σ-IHT (top), RMC-ℓ_σ-IST (middle) and RMC-ℓ_1-ADM (bottom) are reported in Figure 2, which are given in terms of phase transition diagrams. In Figure 2, the white zones denote perfect recovery in all the experiments, while the black ones denote failure for all the experiments. In each diagram, the x-axis represents the ratio of rank, i.e., we let $\rho_r = \frac{\text{rank}}{m} \in [0.04, 0.4]$, and the y-axis represents the level of outliers, i.e., we let $\rho_o = \frac{\sharp\text{outliers}}{m^2} \in [0, 0.6]$. The level of missing entries ρ_m varies from left to right in each row. As shown in Figure 2, our approach outperforms RMC-ℓ_1-ADM when ρ_o and ρ_r increase. We also observe that RMC-ℓ_σ-IHT performs better than RMC-ℓ_σ-IST when the level of outliers increases, while RMC-ℓ_σ-IST outperforms RMC-ℓ_σ-IHT when the ratio of missing entries increases.

Comparison of the computational time and the relative error are also reported in Table 2. In this experiment, the level of missing entries $\rho_m = \{20\%, 30\%\}$, the ratio of rank $\rho_r = 0.1$ and the level of outliers ρ_o varies between $\{0.1, 0.15, 0.2, 0.25, 0.3\}$. For each ρ_o, we randomly generate 20 instances and then average the results. In the table, "time" denotes the CPU time, with the unit being second, and "rel.err" represents the relative error introduced in the previous paragraph. The results also demonstrate the improved performance of our methods in most of the cases on CPU time and relative error, especially for RMC-ℓ_σ-IHT.

5.2. Image Inpainting and Denoising

One typical application of matrix completion is the image inpainting problem [4]. The datasets and the experiment are conducted as follows:

1. We first choose five gray images, named "Baboon", "Camera Man", "Lake", "Lena" and "Pepper" (the size of each image is 512×512), each of which is stored in a matrix M.
2. The outliers matrix E is added to each M, where E is generated in the same way as the previous experiment, and the level of outliers ρ_o varies among $\{0.3, 0.4, 0.5, 0.6, 0.7\}$.
3. The ratio of the missing entries is set to 30%. RMC-ℓ_σ-IST, RMC-ℓ_1-ADM and MC-ℓ_2-IST, are tested in this experiment. In addition, we also test the Cauchy loss-based model $\min_X \ell_c(X) + \lambda\|X\|_*$, which is denoted as RMC-ℓ_c-IST, where:

$$\ell_c := \frac{c^2}{2} \sum_{(i,j)\in\Omega} \ln\left(1 + \left(X_{ij} - B_{ij}\right)^2 / c^2\right),$$

where $c > 0$ is a parameter controlling the robustness. Empirically, we set $c = 0.15$. Other parameters are set to the same as those of RMC-ℓ_σ-IST. The above model is also solved

by soft thresholding similar to Algorithm 3. Note that Cauchy loss has a similar shape as that of Welsch loss and also enjoys the redescending property; such a loss function is also frequently used in the robust statistics literature. The initial guess is $X^0 = 0$. The stopping criterion is $\|X^{(k+1)} - X^{(k)}\|_F \leq 10^{-2}$, or the iterations exceed 500.

Detailed comparison results in terms of the relative error and CPU time are listed in Table 3, from which one can see the efficiency of our method. Indeed, experimental results show that our method can be terminated within 80 iterations. According to the relative error in Table 3, our method performs the best in almost all cases, followed by RMC-ℓ_c-IST. This is not surprising because the Cauchy loss-based model enjoys similar properties as the proposed model. We also observe that the RMC-ℓ_1-ADM algorithm cannot deal with situations when images are heavily contaminated by outliers. This illustrates the robustness of our method.

Table 2. Comparison of RMC-ℓ_σ-IHT(Algorithm 1), RMC-ℓ_σ-IST(Algorithm 3) and RMC-ℓ_1-ADM [17] on CPU time and the relative error on synthetic data. $\rho_m = 0.3$, $\rho_r = 0.1$. rel.err, relative error.

ρ_m	ρ_o	**RMC-ℓ_σ-IHT Algorithm 1**		**RMC-ℓ_σ-IST Algorithm 3**		**RMC-ℓ_1-ADM [17]**	
		Time	**rel.err**	**Time**	**rel.err**	**Time**	**rel.err**
0.2	0.1	15.43	3.80×10^{-03}	20.53	4.55×10^{-02}	19.24	$\mathbf{2.58} \times 10^{-06}$
	0.15	15.31	4.40×10^{-03}	21.26	4.96×10^{-02}	18.32	$\mathbf{2.33} \times 10^{-06}$
	0.2	16.93	5.40×10^{-03}	22.95	5.53×10^{-02}	48.97	$\mathbf{2.82} \times 10^{-04}$
	0.25	19.04	$\mathbf{5.80} \times 10^{-03}$	26.41	6.23×10^{-02}	243.80	1.07×10^{-01}
	0.3	27.10	$\mathbf{7.00} \times 10^{-03}$	29.47	7.01×10^{-02}	137.99	3.16×10^{-01}
	0.35	26.35	$\mathbf{8.00} \times 10^{-03}$	36.03	8.10×10^{-02}	99.26	4.86×10^{-01}
	0.4	23.91	$\mathbf{1.03} \times 10^{-02}$	37.41	9.41×10^{-02}	79.85	6.38×10^{-01}
	0.45	29.64	$\mathbf{1.24} \times 10^{-02}$	45.68	1.10×10^{-01}	67.45	7.77×10^{-01}
	0.5	40.41	$\mathbf{1.69} \times 10^{-02}$	61.39	1.37×10^{-01}	60.08	9.52×10^{-01}
	0.55	60.28	$\mathbf{2.45} \times 10^{-02}$	103.87	1.80×10^{-01}	68.52	$1.39 \times 10^{+00}$
	0.6	102.19	$\mathbf{3.69} \times 10^{-02}$	154.04	2.65×10^{-01}	144.37	$2.86 \times 10^{+00}$
0.3	0.1	16.38	5.20×10^{-03}	24.14	5.66×10^{-02}	24.81	$\mathbf{2.86} \times 10^{-06}$
	0.15	20.14	$\mathbf{5.00} \times 10^{-03}$	23.85	6.41×10^{-02}	110.67	8.30×10^{-03}
	0.2	22.83	$\mathbf{6.00} \times 10^{-03}$	25.92	7.00×10^{-02}	117.91	1.15×10^{-01}
	0.25	20.71	$\mathbf{7.00} \times 10^{-03}$	28.93	7.97×10^{-02}	118.10	3.08×10^{-01}
	0.3	20.77	$\mathbf{8.80} \times 10^{-03}$	32.99	9.21×10^{-02}	89.56	4.68×10^{-01}
	0.35	21.28	$\mathbf{8.20} \times 10^{-03}$	33.72	9.09×10^{-02}	88.73	4.66×10^{-01}
	0.4	27.64	$\mathbf{1.15} \times 10^{-02}$	41.53	1.05×10^{-01}	75.07	5.98×10^{-01}
	0.45	32.38	$\mathbf{1.40} \times 10^{-02}$	48.45	1.23×10^{-01}	71.14	7.13×10^{-01}
	0.5	44.53	$\mathbf{1.68} \times 10^{-02}$	84.67	1.50×10^{-01}	73.63	8.02×10^{-01}
	0.55	62.23	$\mathbf{2.26} \times 10^{-02}$	125.48	1.95×10^{-01}	78.34	8.84×10^{-01}
	0.6	92.14	$\mathbf{3.26} \times 10^{-02}$	241.35	2.78×10^{-01}	74.09	$1.07 \times 10^{+00}$

To better illustrate the robustness of our method empirically, we also attach images recovered by the three methods in Figure 3. For the sake of saving space, we merely list the recovery results for the case $\rho_o = 0.6$ with 30% missing entries. In Figure 3, the first column represents five original images, namely, "Baboon", "Camera Man", "Lake", "Lena" and "Pepper". Images in the second column are contaminated images with 60% outliers and 30% missing entries. Recovered results of each image are report in the remaining columns respectively by using RMC-ℓ_σ-IST, RMC-ℓ_1-ADM, MC-ℓ_2-IST and RMC-ℓ_c-IST. One can observe that the images recovered by our method retain most of the important information, followed by RMC-ℓ_c-IST.

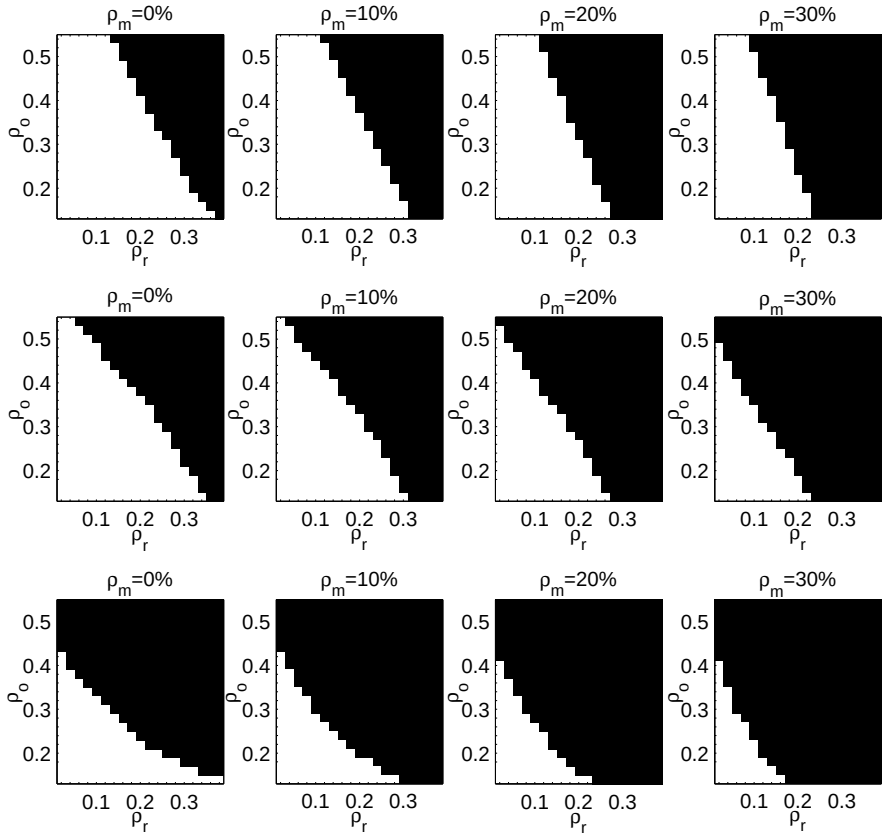

Figure 2. Phase transition diagrams of RMC-ℓ_σ-IHT (Algorithm 1), RMC-ℓ_σ-IST (Algorithm 3) and RMC-ℓ_1-ADM [17]. The first row: RMC-ℓ_σ-IHT; the second row: RMC-ℓ_σ-IST; the last row: RMC-ℓ_1-ADM. x-axis: $\rho_r \in [0.04, 0.4]$; y-axis: $\rho_o \in [0, 0.6]$. From the first column to the last column, ρ_m varies from 0–30%.

Our next experiment is designed to show the effectiveness of our method in dealing with the non-Gaussian noise. We assume that the entries of the noise matrix E are i.i.d drawn from Student's t distribution, with three degrees of freedom. We then scale E by a factor s_n, and we denote the corresponding $E := s_n \cdot E$. The noise scale factor s_n varies in $\{0.01, 0.05, 0.1\}$, and ρ_m varies in $\{0.1, 0.3, 0.5\}$. The results are shown in Table 4, where the image "Building" is used. We list the recovered images in Figure 4 with the case $s_n = 0.05$. From the table and the recovered images, we can see that our method also performs well when the image is only contaminated by non-Gaussian noise.

Table 3. Experimental results of RMC-ℓ_σ-IST (Algorithm 3), RMC-ℓ_1-ADM [17] and MC-ℓ_2-IST [16] on different images with $\rho_r = 0.1$, $\rho_m = 0.3$ and ρ_o varying from 0.3 to 0.7.

ρ_o	Images	Baboon		Camera Man		Lake		Lena		Pepper	
	Method	**Time**	**rel.err**	**Time**	**rel.err**	**Time**	**rel.err**	**Time**	**rel.err**	**Time**	**rel.err**
0.3	RMC-ℓ_σ-IST (Algorithm 3)	**3.17**	1.46×10^{-02}	**3.55**	$\mathbf{1.74}\times10^{-02}$	**3.79**	$\mathbf{1.61}\times10^{-02}$	**4.36**	$\mathbf{2.05}\times10^{-02}$	**3.80**	$\mathbf{1.10}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	32.22	2.86×10^{-02}	35.87	4.36×10^{-02}	26.74	4.57×10^{-02}	20.67	3.98×10^{-02}	33.08	2.46×10^{-02}
	MC-ℓ_2-IST [16]	68.33	$4.35\times10^{+00}$	72.44	$4.44\times10^{+00}$	68.39	$4.14\times10^{+00}$	68.68	$4.22\times10^{+00}$	68.38	$3.07\times10^{+00}$
	RMC-ℓ_c-IST	5.19	$\mathbf{1.38}\times10^{-02}$	5.60	1.83×10^{-02}	5.24	1.70×10^{-02}	4.73	2.46×10^{-02}	4.36	1.61×10^{-02}
0.4	RMC-ℓ_σ-IST (Algorithm 3)	**3.76**	1.73×10^{-02}	**3.94**	$\mathbf{2.15}\times10^{-02}$	**4.69**	$\mathbf{1.96}\times10^{-02}$	**4.58**	$\mathbf{2.41}\times10^{-02}$	**4.91**	$\mathbf{1.42}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	30.93	3.51×10^{-02}	36.76	5.16×10^{-02}	26.67	5.48×10^{-02}	22.41	4.76×10^{-02}	32.18	3.28×10^{-02}
	MC-ℓ_2-IST [16]	68.51	$5.07\times10^{+00}$	68.94	$5.08\times10^{+00}$	68.09	$4.74\times10^{+00}$	68.84	$4.88\times10^{+00}$	68.68	$3.54\times10^{+00}$
	RMC-ℓ_c-IST	4.88	$\mathbf{1.70}\times10^{-02}$	5.73	2.37×10^{-02}	5.34	2.21×10^{-02}	5.39	2.89×10^{-02}	5.56	1.87×10^{-02}
0.5	RMC-ℓ_σ-IST (Algorithm 3)	**4.01**	$\mathbf{2.13}\times10^{-02}$	**4.44**	$\mathbf{2.61}\times10^{-02}$	**5.29**	$\mathbf{2.40}\times10^{-02}$	**5.27**	$\mathbf{2.76}\times10^{-02}$	**6.77**	$\mathbf{1.63}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	24.95	4.91×10^{-02}	27.69	6.57×10^{-02}	22.75	6.92×10^{-02}	20.74	6.71×10^{-02}	26.86	3.98×10^{-02}
	MC-ℓ_2-IST [16]	68.30	$5.56\times10^{+00}$	69.64	$5.62\times10^{+00}$	68.71	$5.37\times10^{+00}$	68.56	$5.44\times10^{+00}$	68.71	$3.91\times10^{+00}$
	RMC-ℓ_c-IST	6.63	2.18×10^{-02}	6.94	2.95×10^{-02}	5.84	2.90×10^{-02}	6.10	3.32×10^{-02}	6.94	2.15×10^{-02}
0.6	RMC-ℓ_σ-IST (Algorithm 3)	**4.98**	$\mathbf{2.65}\times10^{-02}$	6.36	$\mathbf{3.37}\times10^{-02}$	**7.96**	$\mathbf{3.11}\times10^{-02}$	**5.75**	$\mathbf{3.49}\times10^{-02}$	**9.52**	$\mathbf{2.20}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	15.55	1.41×10^{-01}	15.21	1.61×10^{-01}	15.23	1.48×10^{-01}	15.56	1.38×10^{-01}	15.95	9.71×10^{-02}
	MC-ℓ_2-IST [16]	68.22	$6.06\times10^{+00}$	69.93	$6.17\times10^{+00}$	68.73	$5.77\times10^{+00}$	68.34	$5.88\times10^{+00}$	68.51	$4.23\times10^{+00}$
	RMC-ℓ_c-IST	7.93	2.70×10^{-02}	**6.08**	4.51×10^{-02}	8.19	3.22×10^{-02}	7.87	3.81×10^{-02}	10.36	2.85×10^{-02}
0.7	RMC-ℓ_σ-IST (Algorithm 3)	**8.74**	$\mathbf{3.59}\times10^{-02}$	**11.37**	$\mathbf{4.41}\times10^{-02}$	**11.75**	$\mathbf{4.21}\times10^{-02}$	**9.59**	$\mathbf{4.16}\times10^{-02}$	**19.95**	$\mathbf{2.69}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	44.31	$1.90\times10^{+00}$	44.63	$1.96\times10^{+00}$	45.16	$1.81\times10^{+00}$	43.49	$1.85\times10^{+00}$	43.88	$1.37\times10^{+00}$
	MC-ℓ_2-IST [16]	68.54	$6.52\times10^{+00}$	68.75	$6.59\times10^{+00}$	69.06	$6.18\times10^{+00}$	68.41	$6.22\times10^{+00}$	68.62	$4.52\times10^{+00}$
	RMC-ℓ_c-IST	13.12	$\mathbf{3.59}\times10^{-02}$	23.03	5.03×10^{-02}	15.19	4.36×10^{-02}	22.95	4.68×10^{-02}	14.78	3.86×10^{-02}

Figure 3. Comparison of RMC-ℓ_σ-IST, RMC-ℓ_1-ADM and MC-ℓ_2-IST on different images with 60% outliers and 30% missing entries. (**a**) The original low rank images; (**b**) images with 30% missing entries and contaminated by 70% outliers; (**c**) images recovered by RMC-ℓ_σ-IST (Algorithm 3); (**d**) images recovered by RMC-ℓ_1-ADM [17]; (**e**) images recovered by MC-ℓ_2-IST [16]; (**f**) images recovered by RMC-ℓ_c-IST.

Figure 4. Recovery results of RMC-ℓ_σ-IST (third), RMC-ℓ_1-ADM (fourth) and MC-ℓ_2-IST (fifth) on the image "Building" contaminated by non-Gaussian noise with $s_n = 0.05$ and 30% missing entries.

Table 4. Experimental results on the image "Building", contaminated by non-Gaussian noise with varying ρ_m and the noise scale.

s_n	ρ_m	RMC-ℓ_σ-IST Algorithm 3		RMC-ℓ_1-ADM [17]		MC-ℓ_2-IST [16]	
		Time	**rel.err**	**Time**	**rel.err**	**Time**	**rel.err**
0.01	0.1	0.91	$\mathbf{6.70}\times10^{-03}$	2.57	1.76×10^{-02}	0.59	6.70×10^{-03}
	0.3	0.90	$\mathbf{9.60}\times10^{-03}$	2.40	2.32×10^{-02}	0.85	9.60×10^{-03}
	0.5	1.05	$\mathbf{1.44}\times10^{-02}$	2.77	3.24×10^{-02}	1.29	1.44×10^{-02}
0.05	0.1	1.24	$\mathbf{1.58}\times10^{-02}$	1.17	2.16×10^{-02}	0.82	1.91×10^{-02}
	0.3	1.11	$\mathbf{2.03}\times10^{-02}$	1.37	2.70×10^{-02}	1.64	3.63×10^{-02}
	0.5	1.32	$\mathbf{2.49}\times10^{-02}$	2.22	3.61×10^{-02}	1.94	2.88×10^{-02}
0.1	0.1	2.34	3.31×10^{-02}	1.08	$\mathbf{3.04}\times10^{-02}$	1.35	5.72×10^{-02}
	0.3	3.30	$\mathbf{3.40}\times10^{-02}$	1.44	3.78×10^{-02}	2.32	4.28×10^{-02}
	0.5	3.70	$\mathbf{4.66}\times10^{-02}$	2.42	5.53×10^{-02}	3.98	1.55×10^{-01}

5.3. Background Subtraction

Background subtraction, also known as foreground detection, is one of the major tasks in computer vision, which aims at detecting changes in image or video sequences and finds application in video surveillance, human motion analysis and human-machine interaction from static cameras [35].

Given a sequence of images, one can cast them into a matrix B by vectorizing each image and then stacking row by row. In many cases, it is reasonable to assume that the background varies little. Consequently, the background forms a low rank matrix M, while the foreground activity is spatially localized and can be seen as the error matrix E. Correspondingly, the image sequence matrix B can be expressed as the sum of a low rank background matrix M and a sparse error matrix E, which represents the activity in the scene.

In practice, it is reasonable to assume that some entries of the image sequence are missing and the images are contaminated by noise or outliers. Therefore, the foreground object detection problem can be formulated as a robust matrix completion problem. Ref. [36] proposed to use the LAD-loss-based matrix completion approach to separate M and E. The data of this experiment were downloaded from http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html.

Our experiment in this scenario is implemented as follows:

1. We choose the sequence named "Restaurant" for our experiment, which consists of 3057 color images. Each image of "Restaurant" is 160×120 in size. From the sequence, we pick 100 continuous images and convert them to gray images to form the original matrix B, which is 100×19200 in size, where each row is a vector converted from an image.
2. Two types of non-Gaussian noise are added to B. The first type of noise is drawn from the chi-square distribution, with four degree of freedom; the second type of noise is drawn from Student's t distribution, with three degrees of freedom. Then, the two types of noise are simultaneously rescaled by $s_n = \{0.01, 0.02, 0.05\}$. The last 50% of the entries are missing randomly.
3. RMC-ℓ_σ-IHT and RMC-ℓ_1-ADM are used to deal with this problem. We set $R = 1$ in RMC-ℓ_σ-IHT. The initial guess is the zero matrix. The stopping criterion is $\|X^{(k+1)} - X^{(k)}\|_F \leq 10^{-2}$, or the iterations exceed 200.

The running time and relative error are reported in Table 5. From the table, we see that the proposed approach is faster and gives smaller relative errors. To give an intuitive impression, we choose five frames from each image sequence, as shown in Figure 5, from which we can observe that when the image sequences are corrupted by noise ($s_n = 0.05$) and missing entries, both of the methods can successfully extract the background and foreground images, and it seems that our method performs better because the details of the background images are recovered well, whereas the LAD-based

approach does not seem to perform as well as ours where some details of the background are added to the foreground. It can be also observed that none of the two methods can recover the missing entries in the foreground. In order to achieve this, maybe more effective approaches are needed.

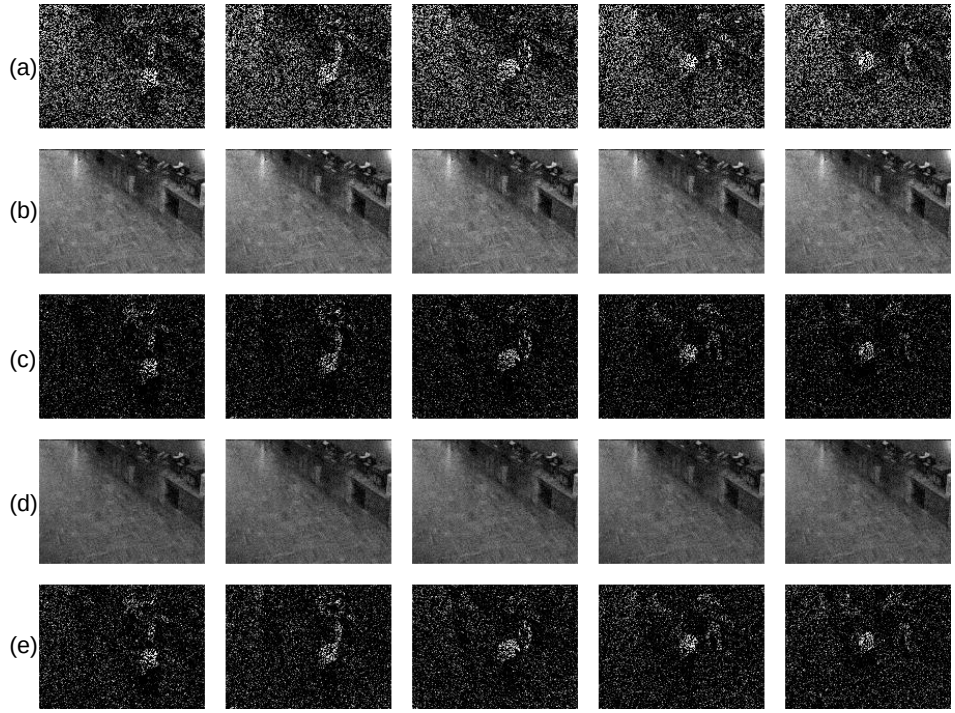

Figure 5. Comparison between RMC-ℓ_σ-IHT (Algorithm 1) and RMC-ℓ_1-ADM [17] on extracting the image sequence "Restaurant" with $\rho_m = 50\%$ and contaminated by two types of non-Gaussian noise with $s_n = 0.05$. (**a**) The original image sequence; (**b**) the image sequence with missing entries and contaminated by noise; (**c**) background extracted by RMC-ℓ_σ-IHT (Algorithm 1); (**d**) foreground extracted by RMC-ℓ_σ-IHT (Algorithm 1); (**e**) background extracted by RMC-ℓ_1-ADM [17]; (**f**) foreground extracted by RMC-ℓ_1-ADM [17].

Table 5. Experiment results on "Restaurant" contaminated by non-Gaussian noise and 50% missing entries.

s_n	Method	Time	rel.err
0.01	RMC-ℓ_σ-IHT (Algorithm 1)	**70.58**	$\mathbf{9.77}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	229.88	1.14×10^{-01}
0.02	RMC-ℓ_σ-IHT (Algorithm 1)	**58.51**	$\mathbf{9.78}\times10^{-02}$
	RMC-ℓ_1-ADM [17]	230.24	1.30×10^{-01}
0.05	RMC-ℓ_σ-IHT (Algorithm 1)	**99.87**	$\mathbf{1.14}\times10^{-01}$
	RMC-ℓ_1-ADM [17]	221.60	2.37×10^{-01}

6. Concluding Remarks

The correntropy loss function has been studied in the literature [20,21] and has found many successful applications [29–31]. Learning with correntropy-induced losses could be resistant to non-Gaussian noise and outliers while ensuring good prediction accuracy simultaneously with properly chosen parameter σ. This paper addressed the robust matrix completion problem based on the correntropy loss. The proposed approach was shown to be efficient to deal with non-Gaussian noise and sparse gross errors. The nonconvexity of the proposed approach was due to using the ℓ_σ loss. Based on the above approach, we proposed two nonconvex optimization models and extend them to the more general robust affine rank minimization problems. Two gradient-based iterative schemes

to solve the nonconvex optimization problems were offered, with convergence rate results being obtained under proper assumptions. It would be interesting to investigate similar convergence and recoverability results for other redescending-type loss functions-based models. Numerical experiments verified the improved performance of our methods, where empirically, the parameter σ for ℓ_σ is set to 0.5 and λ for the nuclear norm model (6) is $\lambda = \frac{\min\{m,n\}}{10\sqrt{\max\{m,n\}}}$.

Acknowledgments: The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC AdGA-DATADRIVE-B (290923). This paper reflects only the authors' views; the Union is not liable for any use that may be made of the contained information; Research Council KUL: GOA/10/09 MaNet, CoEPFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO: PhD/Postdoc grants, projects: G.0377.12 (Structured systems), G.088114N (Tensor-based data similarity); IWT: PhD/Postdoc grants, projects: SBOPOM(100031); iMinds Medical Information Technologies SBO 2014; Belgian Federal Science Policy Office: IUAPP7/19 (DYSCO, Dynamical systems, control and optimization, 2012–2017).

Author Contributions: Y.Y., Y.F., and J.A.K.S. proposed and discussed the idea; Y.Y. and Y.F. conceived and designed the experiments; Y.Y. performed the experiments; Y.Y. and Y.F. analyzed the data; J.A.K.S. contributed analysis tools; Y.Y. and Y.F. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Lipschitz Continuity of the Gradient of ℓ_σ and Some Propositions

The propositions given in the Appendix hold for both ℓ_σ and $\tilde{\ell}_\sigma$. For simplicity, we only present the formulas for $\tilde{\ell}_\sigma$. We first give some notations. Let $\mathrm{vec}(\cdot)$ be the vectorization operator over any matrix space $\mathbb{R}^{s\times t}$, with $\mathrm{vec}(B) \in \mathbb{R}^{st}$ and:

$$\mathrm{vec}(B)_{(i-1)t+j} = B_{ij},\ 1 \le i \le s,\ 1 \le j \le t,\ \forall B \in \mathbb{R}^{s\times t}.$$

We further define matrix $A \in \mathbb{R}^{p\times mn}$, where:

$$A^T = \left[\mathrm{vec}\left(A^1\right), \mathrm{vec}\left(A^2\right), \ldots, \mathrm{vec}\left(A^p\right)\right].$$

Based on the above notations, the vectorized form of $\mathcal{A}(X)$ is written as:

$$\mathrm{vec}(\mathcal{A}(X)) = A\mathrm{vec}(X),$$

and the gradient of ℓ_σ at X can be rewritten as:

$$\mathrm{vec}\left(\nabla \ell_\sigma(X)\right) = A^T \Lambda \left(A\mathrm{vec}(X) - b\right),$$

where $\Lambda \in \mathbb{R}^{p\times p}$ is a diagonal matrix with:

$$\Lambda_{ii} = \exp\left(-(\langle A^i, X\rangle - b_i)^2/\sigma^2\right), \quad 1 \le i \le p.$$

Let $\|A\|_2$ be the spectral norm of A. The following proposition shows that the gradient of $\tilde{\ell}_\sigma$ is Lipschitz continuous.

Proposition A1. *The gradient of $\tilde{\ell}_\sigma$ is Lipschitz continuous. That is, for any $X, Y \in \mathbb{R}^{m\times n}$, it holds that:*

$$\|\nabla\tilde{\ell}_\sigma(X) - \nabla\tilde{\ell}_\sigma(Y)\|_F \le \|A\|_2^2 \|X - Y\|_F.$$

Proof. With notations introduced above, we know that:

$$\begin{aligned}\|\nabla\tilde{\ell}_\sigma(X) - \nabla\tilde{\ell}_\sigma(Y)\|_F &= \|A^T\Lambda_X\left(A\mathrm{vec}(X) - b\right) - A^T\Lambda_Y\left(A\mathrm{vec}(Y) - b\right)\|_F \\ &\le \|A\|_2\|\Lambda_X\left(A\mathrm{vec}(X) - b\right) - \Lambda_Y\left(A\mathrm{vec}(Y) - b\right)\|_F,\end{aligned}$$

where Λ_X and Λ_Y are the diagonal matrices corresponding to $\nabla\tilde{\ell}_\sigma(X)$ and $\nabla\tilde{\ell}_\sigma(Y)$. It remains to show that:

$$\|\Lambda_X\left(A\mathrm{vec}(X)-b\right)-\Lambda_Y\left(A\mathrm{vec}(Y)-b\right)\|_F \le \|A\|_2\|X-Y\|_F.$$

By letting $z_1 = A\mathrm{vec}(X)-b$ and $z_2 = A\mathrm{vec}(Y)-b$, we observe that:

$$\begin{aligned} & \|\Lambda_X\left(A\mathrm{vec}(X)-b\right)-\Lambda_Y\left(A\mathrm{vec}(Y)-b\right)\|_F^2 \\ = & \sum_{i=1}^{p}\left(\exp\left(-z_{1,i}^2/\sigma^2\right)z_{1,i}-\exp\left(-z_{2,i}^2/\sigma^2\right)z_{2,i}\right)^2. \end{aligned}$$

Combining with the fact that for any $t_1, t_2 \in \mathbb{R}$ and $\sigma > 0$,

$$|\exp(-t_1^2/\sigma^2)t_1-\exp(-t_2^2/\sigma^2)t_2| \le |t_1-t_2|,$$

we have:

$$\begin{aligned} & \|\Lambda_X\left(A\mathrm{vec}(X)-b\right)-\Lambda_Y\left(A\mathrm{vec}(Y)-b\right)\|_F^2 \\ \le & \|A\mathrm{vec}(X)-A\mathrm{vec}(Y)\|_F^2 \le \|A\|_2^2\|X-Y\|_F^2. \end{aligned}$$

As a result, $\|\tilde{\nabla}\ell_\sigma(X)-\tilde{\nabla}\ell_\sigma(Y)\|_F \le \|A\|_2^2\|X-Y\|_F$. This completes the proof. □

The following conclusion is a consequence of Proposition A1.

Proposition A2. *For any $X, Y \in \mathbb{R}^{m\times n}$, it holds that:*

$$\tilde{\ell}_\sigma(X) \le \tilde{\ell}_\sigma(Y)+\langle\nabla\tilde{\ell}_\sigma(Y), X-Y\rangle+\|A\|_2^2/2\|X-Y\|_F^2. \tag{A1}$$

Proposition A3. *Let $\{X^{(k)}\}$ be generated by Algorithms 1 or 3 with $\alpha > L = \|A\|_2$. Then, it holds that:*

$$\|X^{(k)}-X^{(k+1)}\|_F \to 0.$$

Proof. We first consider $\{X^{(k)}\}$ generated by Algorithm 1. Following from the fact that $\mathrm{rank}(X^{(k)}) \le R$ and $X^{(k+1)}$ is the best rank-R approximation of $Y^{(k+1)}$, we know that:

$$\begin{aligned} & \frac{\alpha}{2}\|X^{(k+1)}-X^{(k)}\|_F^2+\langle\nabla\tilde{\ell}_\sigma(X^{(k)}), X^{(k+1)}-X^{(k)}\rangle \\ = & \frac{\alpha}{2}\left\|X^{(k+1)}-X^{(k)}+\frac{1}{\alpha}\nabla\ell_\sigma(X^{(k)})\right\|_F^2-\frac{\alpha}{2}\left\|\frac{1}{\alpha}\nabla\ell_\sigma(X^{(k)})\right\|_F^2 \le 0. \end{aligned}$$

This together with (A1) gives:

$$\tilde{\ell}_\sigma(X^{(k+1)}) \le \tilde{\ell}_\sigma(X^{(k)})-\frac{\alpha-L}{2}\|X^{(k+1)}-X^{(k)}\|_F^2,$$

which implies that the sequence $\{\tilde{\ell}_\sigma(X^{(k)})\}$ is monotonically decreasing. Due to the lower boundness of $\tilde{\ell}_\sigma$, we see that $\lim_{k\to\infty}\|X^{(k+1)}-X^{(k)}\|_F = 0$.

When $\{X^{(k)}\}$ is generated by Algorithm 3, after simple computation, we have that $X^{(k+1)}$ is the minimizer of:

$$\min_X \frac{1}{2}\|X-Y^{(k+1)}\|_F^2+\frac{\lambda}{\alpha}\|X\|_*.$$

we thus have:

$$\begin{aligned} & \frac{\alpha}{2}\|X^{(k+1)} - X^{(k)}\|_F^2 + \langle \nabla \tilde{\ell}_\sigma(X^{(k)}), X^{(k+1)} - X^{(k)} \rangle + \lambda \|X^{(k+1)}\|_* - \lambda \|X^{(k)}\|_* \\ = \quad & \frac{\alpha}{2}\|X^{(k+1)} - Y^{(k+1)}\|_F^2 + \lambda \|X^{(k+1)}\|_* - \frac{\alpha}{2}\left\| \frac{1}{\alpha} \nabla \ell_\sigma(X^{(k)}) \right\|_F^2 - \lambda \|X^{(k)}\|_* \leq 0. \end{aligned}$$

This in connection with Proposition A2 reveals:

$$\tilde{\ell}_\sigma(X^{(k+1)}) + \lambda \|X^{(k+1)}\|_* \leq \tilde{\ell}_\sigma(X^{(k)}) + \lambda \|X^{(k)}\|_* - \frac{\alpha - L}{2}\|X^{(k+1)} - X^{(k)}\|_F^2.$$

Analogously, we have $\lim_{k\to\infty} \|X^{(k+1)} - X^{(k)}\|_F = 0$. This completes the proof. □

References

1. Srebro, N.; Jaakkola, T. Weighted low-rank approximations. In Proceedings of the 20th International Conference on Machine Learning, Copenhagen, Denmark 11–12 June 2003; Volume 3, pp. 720–727.
2. Prize Website, N. Available online: http://www.netflixprize.com (accessed on 2 March 2018).
3. Komodakis, N. Image completion using global optimization. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; Volume 1, pp. 442–452.
4. Bertalmio, M.; Sapiro, G.; Caselles, V.; Ballester, C. Image inpainting. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, LA, USA, 23–28 July 2000; pp. 417–424.
5. Ji, H.; Liu, C.; Shen, Z.; Xu, Y. Robust video denoising using low rank matrix completion. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1791–1798.
6. Candès, E.J.; Plan, Y. Matrix completion with noise. *Proc. IEEE* **2010**, *98*, 925–936.
7. Candès, E.J.; Recht, B. Exact matrix completion via convex optimization. *Found. Comput. Math.* **2009**, *9*, 717–772.
8. Gross, D. Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Inf. Theory* **2011**, *57*, 1548–1566.
9. Keshavan, R.H.; Montanari, A.; Oh, S. Matrix completion from noisy entries. *J. Mach. Learn. Res.* **2010**, *99*, 2057–2078.
10. Koltchinskii, V.; Lounici, K.; Tsybakov, A.B. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Stat.* **2011**, *39*, 2302–2329.
11. Signoretto, M.; Van de Plas, R.; De Moor, B.; Suykens, J.A.K. Tensor versus matrix completion: A comparison with application to spectral data. *IEEE Signal Process. Lett.* **2011**, *18*, 403–406.
12. Hu, Y.; Zhang, D.; Ye, J.; Li, X.; He, X. Fast and Accurate Matrix Completion via Truncated Nuclear Norm Regularization. *IEEE Trans. Pattern Anal.* **2013**, *35*, 2117–2130.
13. Cai, J.F.; Candès, E.J.; Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **2010**, *20*, 1956–1982.
14. Goldfarb, D.; Ma, S. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Found. Comput. Math.* **2011**, *11*, 183–210.
15. Ji, S.; Ye, J. An accelerated gradient method for trace norm minimization. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; pp. 457–464.
16. Ma, S.; Goldfarb, D.; Chen, L. Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* **2011**, *128*, 321–353.
17. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *J. ACM (JACM)* **2011**, *58*, 11.
18. Hastie, T. Matrix Completion and Large-Scale SVD Computations. Available online: http://www.stanford.edu/~hastie/TALKS/SVD_hastie.pdf (accessed on 21 February 2018).

19. Nie, F.; Wang, H.; Cai, X.; Huang, H.; Ding, C. Robust matrix completion via joint Schatten p-norm andlp-norm minimization. In Proceedings of the 2012 IEEE 12th International Conference on Data Mining (ICDM), Brussels, Belgium, 10–13 December 2012; pp. 566–574.
20. Liu, W.; Pokharel, P.P.; Príncipe, J.C. Correntropy: Properties and applications in non-Gaussian signal processing. *IEEE Trans. Signal Process.* **2007**, *55*, 5286–5298.
21. Príncipe, J.C. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*; Springer Science & Business Media: Berlin, Germany, 2010.
22. Feng, Y.; Huang, X.; Shi, L.; Yang, Y.; Suykens, J.A. Learning with the maximum correntropy criterion induced losses for regression. *J. Mach. Learn. Res.* **2015**, *16*, 993–1034.
23. Jain, P.; Meka, R.; Dhillon, I.S. Guaranteed Rank Minimization via Singular Value Projection. In Proceedings of the Advances in Neural Information Processing Systems, Hyatt Regency, VAN, Canada, 6–11 December 2010; Volume 23, pp. 937–945.
24. Beck, A.; Teboulle, M. A linearly convergent algorithm for solving a class of nonconvex/affine feasibility problems. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*; Springer: Berlin, Germany, 2011; pp. 33–48.
25. Rohde, A.; Tsybakov, A.B. Estimation of high-dimensional low-rank matrices. *Ann. Stat.* **2011**, *39*, 887–930.
26. Chen, Y.; Xu, H.; Caramanis, C.; Sanghavi, S. Robust Matrix Completion with Corrupted Columns. *arXiv* **2011**, arXiv:1102.2254.
27. Huber, P.J. *Robust Statistics*; Springer: Berlin, Germany, 2011.
28. Warmuth, M.K. From Relative Entropies to Bregman Divergences to the Design of Convex and Tempered Non-Convex Losses. Available online: http://classes.soe.ucsc.edu/cmps290c/Spring13/lect/9/holycow.pdf (accessed on 21 February 2018).
29. Chen, B.; Xing, L.; Liang, J.; Zheng, N.; Príncipe, J.C. Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion. *IEEE Signal Process. Lett.* **2014**, *21*, 880–884.
30. Chen, B.; Xing, L.; Zhao, H.; Zheng, N.; Príncipe, J.C. Generalized correntropy for robust adaptive filtering. *IEEE Trans. Signal Process.* **2016**, *64*, 3376–3387.
31. Chen, B.; Liu, X.; Zhao, H.; Príncipe, J.C. Maximum correntropy Kalman filter. *Automatica* **2017**, *76*, 70–77.
32. Feng, Y.; Fan, J.; Suykens, J. A Statistical Learning Approach to Modal Regression. *arXiv* **2017** arXiv:1702.05960.
33. Recht, B.; Fazel, M.; Parrilo, P.A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* **2010**, *52*, 471–501.
34. Bolte, J.; Sabach, S.; Teboulle, M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Math. Program.* **2014**, *146*, 459–494.
35. Li, L.; Huang, W.; Gu, I.Y.H.; Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.* **2004**, *13*, 1459–1472.
36. Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In Proceedings of the Advances in Neural Information Processing Systems,Vancouver, BC, Canada, 7–8 December 2009; pp. 2080–2088.