8

Robust Matrix Completion through Nonconvex Approaches and Efficient Algorithms

Yuning Yang
KU Leuven, ESAT-STADIUS, Belgium

Yunlong Feng
KU Leuven, ESAT-STADIUS, Belgium

J.A.K. Suykens
KU Leuven, ESAT-STADIUS, Belgium

8.1 Introduction

The goal of matrix completion is to impute missing values of a possibly low-rank matrix with only partial entries observed. This problem arises in online recommendation systems, computer vision, etc. In real-world applications, the matrix to be recovered might be contaminated by noise or outliers, where robust techniques are needed. In this chapter, we introduce a robust matrix completion model, where the robustness benefits from a nonconvex loss function. Efficient algorithms are proposed to solve the introduced robust matrix completion model. Experiments are carried out on synthetic as well as real datasets to validate the efficiency and effectiveness of the proposed models and algorithms.
The problem of matrix completion aims at recovering a matrix from a sampling of its entries, which has arisen from a variety of real-world applications including online recommendation systems [26, 30], image impainting [1, 20], computer vision, and video denoising [17]. The problem itself could be an ill-posed problem without further constraints since we have fewer samples than entries. However, in many applications including those mentioned-above, it is common that the matrix that we are going to recover has some special structures; for example, low-rank or approximately low-rank, which makes it possible to search within all possible completions.

In matrix completion problems, one tries to approximate the observed entries of the matrix as well as possible while also preserving the low-rank property of the recovered matrix. Mathematically, the problem can be formulated as

$$\min_{L\in\mathbb{R}^{m\times n}} \operatorname{rank}(L) \quad \text{s.t. } L_{ij} = B_{ij}, \quad (i,j)\in\Omega,$$

where $L,\ B \in \mathbb{R}^{m\times n}$, and Ω are an index set. Due to the nonconvexity of the rank function $\mathrm{rank}(\cdot)$, solving this minimization problem is NP-hard in general. To obtain a tractable convex relaxation, the nuclear norm heuristic is usually employed, which also imposes the low-rank property. To measure the approximation ability of a candidate matrix on the observed entries, the least-squares criterion is usually employed in the data fidelity term. In a seminal work, [3] showed that most low-rank matrices can be recovered exactly from partial sampled entries that may have surprisingly small cardinality by using the convex relaxation introduced in [6], and many algorithms have been introduced to solve the convex optimization problem.

In the noisy setting of the matrix completion problem, the corresponding observed matrix turns out to be

$$B_\Omega = L_\Omega + S,$$

where B_Ω denotes the projection of B onto Ω, and S refers to the noise.

When the observed data are corrupted by gross errors, the resulting matrix could be far away from the ground-truth due to the utilization of the least-squares criterion, which is non-robust. To address this problem, some efforts have been made in the literature. In a seminal work, [2] proposed a robust matrix completion approach, in which the model takes the following form

$$\min_{L,S\in\mathbb{R}^{m\times n}} \ \|S\|_1 + \lambda\|L\|_* \quad \text{s.t. } \ L_\Omega + S = B_\Omega. \tag{8.1}$$

The above model can be further formulated as

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

where $\lambda > 0$ is a regularization parameter. The robustness of the model (8.1) results from using the least absolute deviation loss (LAD). This model was later applied to the column-wise robust matrix completion problem in [4].

By further decomposing S into $S = S_1 + S_2$, where S_1 refers to the noise and S_2 stands for the outliers, [10] proposed the following robust reconstruction model

$$\min_{L,S_2\in\mathbb{R}^{m\times n}} \ \|L_\Omega - B_\Omega - S_2\|_F^2 + \lambda\|L\|_* + \gamma\|S_2\|_1,$$

where $\lambda, \gamma > 0$ are regularization parameters. They further showed that the above estimator is equivalent to the one obtained by using the Huber's criterion when evaluating the data-fitting risk. The Huber's criterion was adopted in [10] to introduce robustness into matrix completion. [25] proposed to use an L_p $(0 < p \leq 1)$ loss to enhance the robustness. However, none of the above approaches can be sufficiently robust to gross errors due to the unboundedness of these loss functions, which cannot remove the impact of the gross errors on the output.

We also note that, to enhance the robustness, several approaches have been proposed in low-rank matrix approximation problems, especially for PCA [2, 35, 36]. However, it is necessary to point out the difference between the PCA and the matrix completion. As suggested in [35], the essential difference lies in that in matrix completion problems the support of missing entries is given, whereas in PCA, corrupted entries are never known. From a statistical learning viewpoint, PCA is a typical unsupervised learning problem while the matrix completion can be interpreted as a supervised learning scenario, e.g., the trace regression problem [19, 28] mentioned above, or a transductive learning scenario [29].

In this chapter, motivated by theoretical investigations presented in [7, 33] and empirical success reported in [11, 21], we propose a nonconvex approach by employing an exponential squared type loss, namely, the Welsch loss, which will be introduced later. The Welsch loss was originally introduced in robust statistics to form robust estimators in linear models [12]. In this chapter, we will show that it can also work efficiently in matrix completion problems and bring us robust output. Moreover, for the proposed nonconvex matrix completion problems, we propose efficient algorithms, where at each iteration, the algorithms first compute a rank-one matrix, and then update the new trial as a linear combination of the current trial and the newly generated rank-one matrix. The rank-one matrix is related to the left and right singular vectors of the leading singular value of a certain matrix, which can be found efficiently by the power method or Lanczos method. Therefore, the whole algorithms are also efficient. We also show the sublinear convergence rate of the proposed algorithms.

We would also like to mention the differences between this chapter and our previous works [37, 38], which are focused on robust and low-rank matrix/tensor completion problems. The models in [37] are similar to those presented in this chapter; however, the computational algorithms proposed here are different from those in [37], along with different convergence analysis. Furthermore, we present more numerical experiments than [37]. The work in [38] is focused on robust tensor completion, while this chapter is restricted to robust matrix completion.

This chapter is organized as follows. In Section 8.2, we formally formulate the proposed robust matrix completion problem and introduce the Welsch loss that will be used in our study. Section 8.3 presents the proposed algorithms. We give convergence analysis of the proposed algorithms in Section 8.4. Numerical experiments are carried out in Section 8.5 to validate the efficiency and effectiveness of the proposed algorithms. We end this chapter with conclusions in Section 8.6.

8.2 Problem Formulation

Formally, the matrix completion problem can be formulated as follows:

$$\min_{L\in\mathbb{R}^{m\times n}} \quad \operatorname{rank}(L) \quad \text{s.t.} \quad L_\Omega = B_\Omega, \tag{8.2}$$

where Ω is the set of indices that indicates the observed entries. When there is noise or outliers, a certain loss function should be introduced to penalize the noise or outliers. Previous work usually employs a least absolute deviation (LAD) loss $|\cdot|$. The advantage of the LAD loss is that its resulting problem is convex, and has a theoretical recovery result [2]. However, it is not as resistant to outliers as some nonconvex robust losses [12]. In view of this, our problem will be formulated as follows: to measure or to penalize the difference between L and B, we adopt the following loss function

$$\ell_\sigma(t) = \sigma^2/2\left(1-\exp(-t^2/\sigma^2)\right),$$

which is known as the Welsch loss in robust statistics [13]. Here $\sigma > 0$ is a parameter. In the following, we would like to mention some properties of this loss from different aspects.

- σ controls the robustness. The smaller the parameter σ, the more robustness it gives the problem.
- The influence function of $\ell_\sigma(t)$ is given by

  $$\psi_\sigma(t) = \exp(-t^2/\sigma^2)t.$$

  In fact, the value $\psi_\sigma(t)/t$ can be regarded as a weight of t. One can observe that as t increases, $\psi_\sigma(t)/t$ decreases sharply, which gives a small weight to the value t. On the other hand, the influence function of the LAD loss is only bounded instead of converging to zero.

- Another property of the Welsch loss is that its influence functions are Lipschitz continuous, with Lipschitz constant 1, i.e., for any $t_1, t_2 \in \mathbb{R}$, it holds that
$$|\psi_\sigma(t_1) - \psi_\sigma(t_2)| \leq |t_1 - t_2|.$$
This property is very important and serves as a basis for the convergence of the algorithms presented later.
- As $t \to 0$, ℓ_σ approximates the least-squares loss, which can be seen from their Taylor series. Given a fixed $\sigma > 0$, it holds that $\ell_\sigma(t) = t^2/2 + o(t^2/\sigma^2)$. Therefore, $\ell_\sigma(t) \approx t^2/2$ provided that $t/\sigma \to 0$. This also reminds us that a large σ can lead to the closeness between ℓ_σ and the least-squares loss. Such a property gives more flexibility to the Welsch loss than the LAD loss.

By letting $t_{ij} = L_{ij} - B_{ij}$ where $(i,j) \in \Omega$, and by summing ℓ_σ over all the indices in Ω, we arrive at the following cost function
$$F_\sigma(L) = \sum_{(i,j)\in\Omega} \frac{\sigma^2}{2} \left(1 - \exp\left(-(L_{ij} - B_{ij})^2/\sigma^2\right)\right).$$
Therefore, if the rank information is known a priori, then we can model the problem as
$$\min_{L\in\mathbb{R}^{m\times n}} \ F_\sigma(L) \ \text{ s.t. } \ \text{rank}(L) \leq R. \tag{8.3}$$
Otherwise, we can constrain $F_\sigma(\cdot)$ by nuclear norm constraint, i.e.,
$$\min_{L\in\mathbb{R}^{m\times n}} \ F_\sigma(L) \ \text{ s.t. } \ \|L\|_* \leq \beta, \tag{8.4}$$
where $\beta > 0$ is a parameter to control the complexity of the model.

8.2.1 Extending to the Affine Rank Minimization Problem

It is known that matrix completion is a special case of the following affine rank minimization problem [9, 16, 18, 22, 24, 27]
$$\min_{L\in\mathbb{R}^{m\times n}} \quad \text{rank}(L) \quad \text{s.t.} \quad \mathcal{A}(L) = b, \tag{8.5}$$
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. (8.5) can be reduced to matrix completion if we set $p = \text{card}(\Omega)$, 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.

(8.3) and (8.4) can be naturally extended to handle cases with noise and outliers of (8.5). Denote the cost function as as follows
$$F_\sigma(L) = \frac{\sigma^2}{2} \sum_{i=1}^{p} \left(1 - \exp\left(-\left(\langle A^i, L\rangle - b_i\right)^2/\sigma^2\right)\right). \tag{8.6}$$
The rank constrained problem can be formulated as
$$\min_{L\in\mathbb{R}^{m\times n}} \ F_\sigma(L) \ \text{ s.t. } \ \text{rank}(L) \leq R, \tag{8.7}$$
and the nuclear norm constrained problem takes the form
$$\min_{L\in\mathbb{R}^{m\times n}} \ F_\sigma(L) \ \text{ s.t. } \ \lambda\|L\|_* \leq \beta. \tag{8.8}$$

8.3 Computational Algorithms

In robust regression, problems associated with a robust loss are usually solved by iterative reweighted least squares approaches [13]. Here, since we want to find low-rank solutions to (8.3) and (8.4), it would be proper to explore the structures of these two problems, and consider different algorithms. The algorithms proposed here are called rank-one matrix updating algorithms. The main idea is that, at each iteration, the algorithms compute a rank-one matrix, which is formed by the left and right singular vectors corresponding to the leading singular value of the matrix $\nabla F_\sigma(L^{(k)})$, by using power method or the Lanczos method. Then the algorithms update the new trial via certain linear combinations of the current trial and the newly generated rank-one matrix. In general, the algorithm framework is presented in Algorithm 8.1.

Algorithm 8.1 - Rank-one matrix updating algorithms for solving (8.3) and (8.4)

```
Input: Zero matrix $L^{(0)} = 0$.
Output: $L^{(k+1)}$.
for $k = 1$ to ... do
    • Compute a normalized rank-one matrix $W^{(k)} = \mathbf{u}^{(k)}(\mathbf{v}^{(k)})^\top$:
          $(\mathbf{u}^{(k)}, \mathbf{v}^{(k)}) = \arg \max_{\|\mathbf{u}\|_F=1, \|\mathbf{v}\|_F=1} \mathbf{u}^\top \nabla F_\sigma(L^{(k)}) \mathbf{v}.$    (8.9)
    • Select suitable stepsizes (weights) $(\overline{\alpha}_1, \overline{\alpha}_2)$ and update
          $L^{(k+1)} = \overline{\alpha}_1 L^{(k)} + \overline{\alpha}_2 W^{(k)}.$
end for
```

Algorithm 8.1 is related to some recently developed algorithms in the literature. In [15], a simple algorithm for nuclear norm regularized problems was proposed, where at each iteration, the algorithm also computes a rank-one matrix by using the power method or the Lanczos method. Another rank-one matrix updating algorithm was proposed in [34], where the weights are computed as the matching pursuit type methods [23, 32]. Other rank-one updating algorithms have also been developed; see, e.g., [5, 31, 39]. However, a limitation of the above methods is that they are designed for problems with a convex cost function $F(\cdot)$, while in our case, the Welsch loss-based $F_\sigma(\cdot)$ is highly nonconvex.

When solving (8.9), both power method and Lanczos method can be applied and scaled well to large-scale problems. However, to further improve the efficiency of the proposed algorithms, (8.9) may not be solved exactly. In these cases, performing only a few power iterations may be enough to obtain an acceptable output $W^{(k)}$.

The computational complexity of solving (8.9) is at most $O(mn)$. Furthermore, if the matrix $\nabla F_\sigma(L^{(k)})$ is sparse with N nonzero entries, as in our case, then the complexity can be reduced to $O(N)$ [14]. Therefore, the computational complexity of the whole algorithm might be low.

We also note that the proposed algorithms can be easily extended to solving the affine rank minimization problems (8.7) and (8.8), only by replacing the gradient $\nabla F_\sigma(\cdot)$ in (8.9) by the gradient of the cost function defined in (8.6).

In the following, we specify the ways of choosing $(\overline{\alpha}_1, \overline{\alpha}_2)$ in Algorithm 8.1 for the two different problems (8.3) and (8.4). For (8.3), the weights are chosen by the following simple rule:

$$\overline{\alpha}_2 = -\left\langle F_\sigma(L^{(k)}), W^{(k)} \right\rangle / \|W^{(k)}\|_\Omega^2, \quad \overline{\alpha}_1 = 1, \tag{8.10}$$

where $\|W^{(k)}\|_\Omega = \|W_\Omega^{(k)}\|_F$. For (8.4), we first denote

$$D^{(k)} := -\beta \cdot W^{(k)} - L^{(k)}, \tag{8.11}$$

and let

$$L^{(k+1)} = L^{(k)} + \overline{\alpha} D^{(k)},$$

where $\overline{\alpha} \in (0,1)$ is selected by Armijo search rule:

Fixed scalars $l \in (0,1), \mu \in (0,1)$, and we choose the step-size $\overline{\alpha} = l^m$, where m is the first non-negative integer m such that

$$F_\sigma(L^{(k)} + l^m D^{(k)}) - F_\sigma(L^{(k)}) \leq \mu l^m \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle. \tag{8.12}$$

The idea behind (8.10) is that we want to minimize a quadratic function that majorizes $F_\sigma(\cdot)$ at $L^{(k+1)}$, which forces $\{F_\sigma(L^{(k)})\}$ to decrease, as will be shown in the next section. The idea behind (8.12) follows the Frank-Wolfe method [8]. First, we notice that

$$\begin{aligned} -\beta \cdot W^{(k)} &= -\beta \cdot \arg \max_{\|\mathbf{u}\|_F = 1, \|\mathbf{v}\|_F = 1} \langle \nabla F_\sigma(L^{(k)}), W \rangle \\ &= -\beta \arg \min_{\|W\|_* \leq 1} \langle \nabla F_\sigma(L^{(k)}), W \rangle \\ &= \arg \min_{\|W\|_* \leq \beta} \langle \nabla F_\sigma(L^{(k)}), W \rangle, \end{aligned}$$

where the second equality follows from the duality between the matrix spectral norm and the nuclear norm. As a result, $-\beta W^{(k)}$ lies in the nuclear norm ball $\|W\|_* \leq \beta$, and $D^{(k)}$ is a descent direction of $F_\sigma(\cdot)$ at $L^{(k)}$. Then, a suitable $\overline{\alpha}$ can be chosen by (8.12) to get a sufficient decrease from $F_\sigma(L^{(k)})$ to $F_\sigma(L^{(k+1)})$. In the next section, we will present their convergence analysis.

8.4 Convergence Analysis

In this section, we will establish the convergence results of (8.1). The convergence analysis is based on the fact that the gradient of $F_\sigma(\cdot)$ is Lipschitz continuous with constant 1, as will be shown in Proposition 8.1. We first present the gradient of $F_\sigma(\cdot)$ at L, which is given by

$$\nabla F_\sigma(L) = \Lambda \circ (L - B),$$

where $\Lambda \in \mathbb{R}^{m \times n}$ is a matrix such that if $(i,j) \in \Omega$, then $\Lambda_{ij} = \exp(-(L_{ij} - B_{ij})^2/\sigma^2)$, and $\Lambda_{ij} = 0$ if $(i,j) \notin \Omega$; $\circ$ denotes the Hadamard operator, i.e., entry-wise product.

PROPOSITION 8.1 [[37], Proposition 1] For any matrices $X, Y \in \mathbb{R}^{m \times n}$, there holds

$$\|\nabla F_\sigma(X) - \nabla F_\sigma(Y)\|_F \leq \|X - Y\|_\Omega. \tag{8.13}$$

PROOF 8.1 The proof uses the fact that the influence function $\psi_\sigma(t)$ is Lipschitz continuous, i.e.,

$$|\psi_\sigma(x) - \psi_\sigma(y)| \leq |x - y|, \quad \forall\, x, y \in \mathbb{R}.$$

To verify the above inequality, it suffices to verify that the magnitude of

$$\psi_\sigma^{'}(t) = \exp(-t^2/\sigma^2) - 2\exp(-t^2/\sigma^2)t^2/\sigma^2$$

can be upper bounded by 1. We can denote $u = t^2/\sigma^2 \geq 0$, and in fact, the maximum of the function $|\exp(-u)(1-2u)|$ is 1 at $u = 0$. This shows that for any $t \in \mathbb{R}$ and any $\sigma > 0$, there holds $\left|\psi_\sigma^{'}(t)\right| \leq 1$. As a result, it follows

$$\begin{aligned} &\|\nabla F_\sigma(X) - \nabla F_\sigma(Y)\|_F^2 \\ &= \sum_{(i,j)\in\Omega} (\psi_\sigma(X_{ij}) - \psi_\sigma(Y_{ij}))^2 \\ &\leq \|X - Y\|_\Omega^2, \end{aligned}$$

as desired.

Following (8.13) we immediately have

PROPOSITION 8.2 For any matrices $X, Y \in \mathbb{R}^{m\times n}$, there holds

$$F_\sigma(X) \leq F_\sigma(Y) + \langle \nabla F_\sigma(Y), X - Y\rangle + \frac{\|X - Y\|_\Omega^2}{2}. \tag{8.14}$$

Following Proposition 8.2 we have the following convergence result on applying Algorithm 8.1 with strategy (8.10) to solve (8.3).

THEOREM 8.1 *[Convergence result on applying Algorithm 8.1 with strategy* (8.10) *to solve* (8.3)*] Let* $\{L^{(k)}\}$ *be a sequence generated by Algorithm 8.1 with strategy* (8.10)*. Then* $\{F_\sigma(L^{(k)})\}$ *is nonincreasing.*

PROOF 8.2 Strategy (8.10) tells us that

$$L^{(k+1)} = L^{(k)} - \frac{\langle \nabla F_\sigma(L^{(k)}), W^{(k)}\rangle}{\|W^{(k)}\|_\Omega^2} W^{(k)},$$

which together with (8.14) implies that

$$\begin{aligned} F_\sigma(L^{(k+1)}) &\leq F_\sigma(L^{(k)}) + \langle \nabla F_\sigma(L^{(k)}), L^{(k+1)} - L^{(k)}\rangle + \frac{\|L^{(k+1)} - L^{(k)}\|_\Omega^2}{2} \\ &= F_\sigma(L^{(k)}) - \frac{\langle \nabla F_\sigma(L^{(k)}), W^{(k)}\rangle^2}{2\|W^{(k)}\|_\Omega^2}, \end{aligned}$$

which shows that $\{F_\sigma(L^{(k)})\}$ is nonincreasing. The proof is completed.

We then consider the convergence result on applying Algorithm 8.1 with strategy (8.12) to solve (8.4).

THEOREM 8.2 *[Convergence result on applying Algorithm 8.1 with strategy* (8.12) *to solve* (8.4)*] Let* $\left\{L^{(k)}\right\}$ *be a sequence generated by Algorithm 8.1 with strategy* (8.12) *to solve* (8.4)*. Then every limit point of* $\left\{L^{(k)}\right\}$ *is a critical point of problem* (8.4).

To prove Theorem 8.2, we need some observations and lemmas first.

PROPOSITION 8.3 Let $\overline{W} = \arg\min_{\|W\|_* \leq \beta} \langle \nabla F_\sigma(L), W \rangle$. If L is not a critical point of problem (8.4), then

$$\langle \nabla F_\sigma(L), \overline{W} - L \rangle < 0.$$

PROOF 8.3 Suppose $\langle \nabla F_\sigma(L), \overline{W} - L \rangle \geq 0$. Then it follows

$$\langle \nabla F_\sigma(L), W - L \rangle \geq \langle \nabla F_\sigma(L), \overline{W} - L \rangle \geq 0, \quad \forall\ W \text{ satisfying } \|W\|_* \leq \beta,$$

which implies that L is a critical point of (8.4), deducing a contradiction.

LEMMA 8.1 Let $\{L^{(k)}\}$ be a sequence generated by Algorithm 8.1 with strategy (8.12) to solve (8.4). Then there holds

$$F_\sigma(L^{(k+1)}) - F_\sigma(L^{(k)}) \leq -\frac{2l\mu(1-\mu)\langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle^2}{\|D^{(k)}\|_\Omega^2},$$

where $D^{(k)}$ is defined in (8.11).

PROOF 8.4 The Armijo search rule (8.12) implies that

$$F_\sigma(L^{(k)} + \frac{\overline{\alpha}}{l} D^{(k)}) - F_\sigma(L^{(k)}) > \mu \frac{\overline{\alpha}}{l} \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle.$$

Together with Proposition 8.2, it follows

$$\frac{\overline{\alpha}}{l} \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle + \frac{\overline{\alpha}^2}{2l^2} \|D^{(k)}\|_\Omega^2 > \mu \frac{\overline{\alpha}}{l} \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle.$$

Rearranging the terms and noticing Proposition 8.3, we get

$$\overline{\alpha} > \frac{2l(1-\mu) \left| \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle \right|}{\|D^{(k)}\|_\Omega^2}. \tag{8.15}$$

The Armijo search rule again tells us that

$$F_\sigma(L^{(k+1)}) - F_\sigma(L^{(k)}) \leq \mu \overline{\alpha} \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle. \tag{8.16}$$

Noticing the non-positivity of $\langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle$ and plugging (8.15) into (8.16), we obtain

$$F_\sigma(L^{(k+1)}) - F_\sigma(L^{(k)}) \leq -\frac{2l\mu(1-\mu)\langle \nabla F_\sigma(L^{(k)}, D^{(k)} \rangle^2}{\|D^{(k)}\|_\Omega^2},$$

as desired.

PROOF 8.5 [Proof of Theorem 8.2] Denote

$$\mathbf{W} := \{W \mid \|W\|_* \leq \beta\},$$

and

$$\operatorname{diam}(\mathbf{W}) := \max_{W_1, W_2 \in \mathbf{W}} \|W_1 - W_2\|$$

as the diameter of $\mathbf{W}$. Since $\mathbf{W}$ is compact, $\mathrm{diam}(\mathbf{W})$ is finite. From the definition of $D^{(k)}$, it follows

$$\|D^{(k)}\|_\Omega \leq \|D^{(k)}\|_F \leq \mathrm{diam}(\mathbf{W}).$$

This together with Lemma 8.1 shows that

$$F_\sigma(L^{(k+1)}) - F_\sigma(L^{(k)}) < -\frac{2l\mu(1-\mu)\langle \nabla F_\sigma(L^{(k)}, D^{(k)}\rangle^2}{\mathrm{diam}^2(\mathbf{W})}. \tag{8.17}$$

Lemma 8.1 also tells us that $\{F_\sigma(L^{(k)})\}$ is a monotonically decreasing sequence. Since $F_\sigma(\cdot) \geq 0$, we have

$$F_\sigma(L^{(k+1)}) - F_\sigma(L^{(k)}) \to 0,$$

which together with (8.17) implies that

$$|\langle \nabla F_\sigma(L^{(k)}), D^{(k)}\rangle| \to 0.$$

Let L^* be a limit point of $\{L^{(k)}\}$ and let $\{L^{(k)}\}_{\mathbf{K}}$ be a subsequence of $\{L^{(k)}\}$ such that $\{L^{(k)}\}_{\mathbf{K}} \to L^*$. Furthermore, let $\overline{\mathbf{K}}$ be a subset of $\mathbf{K}$ such that there exists a subsequence $\{W^{(k)}\}_{\overline{\mathbf{K}}}$ of $\{W^{(k)}\}_{\mathbf{K}}$ such that $\{W^{(k)}\}_{\overline{\mathbf{K}}} \to W^*$. Without loss of generality we can assume $\overline{\mathbf{K}}$ is $\mathbf{K}$ itself. Then it follows

$$\langle \nabla F_\sigma(L^*), W^* - L^*\rangle = 0.$$

We claim that W^* is a minimizer of $\min_{W\in\mathbf{W}}\langle \nabla F_\sigma(L^*), W\rangle$. Otherwise suppose $\overline{W}$ is a minimizer. Then $\langle \nabla F_\sigma(L^*), W^* - \overline{W}\rangle > 0$. Since $\{\langle \nabla F_\sigma(L^{(k)}), W^{(k)} - \overline{W}\rangle\}_{k\in\mathbf{K}} \to \langle \nabla F_\sigma(L^*), W^* - \overline{W}\rangle$, when k is sufficiently large, it follows

$$\langle \nabla F_\sigma(L^{(k)}), W^{(k)} - \overline{W}\rangle > 0,$$

which shows that $W^{(k)}$ is not a minimizer of $\min_{W\in\mathbf{W}}\langle \nabla F_\sigma(L^{(k)}), W\rangle$, deducing a contradiction. As a result, by the property of W^* we have

$$\langle \nabla F_\sigma(L^*), W - L^*\rangle \geq \langle \nabla F_\sigma(L^*), W^* - L^*\rangle = 0, \quad \forall\, W \in \mathbf{W},$$

which shows that L^* is a critical point of (8.4). The proof is completed.

Next we estimate the rate of convergence in terms of $|\langle \nabla F_\sigma(L^{(k)}), D^{(k)}\rangle|$. We have the following results.

THEOREM 8.3 *Let $\{L^{(k)}\}$ be a sequence generated by Algorithm 8.1 with strategy* (8.12) *to solve* (8.4). *Then for every $K \geq 1$, we have*

$$\min_{0\leq k\leq K} |\langle \nabla F_\sigma(L^{(k)}), D^{(k)}\rangle|^2 \leq \frac{\mathrm{diam}^2(\mathbf{W})(F_\sigma(L^{(0)}) - F_\sigma^*)}{2l\mu(1-\mu)K},$$

where F_σ^* is the limit of $F_\sigma(L^{(k)})$.

PROOF 8.6 Let $C := \frac{\mathrm{diam}^2(\mathbf{W})}{2l\mu(1-\mu)}$. Then it follows form Lemma 8.1 that

$$\langle \nabla F_\sigma(L^{(k)}), D^{(k)}\rangle^2 \leq C(F_\sigma(L^{(k)}) - F_\sigma(L^{(k+1)})).$$

Summing the above inequality from 0 to K, we get

$$\sum_{k=0}^{K} \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle^2 \leq C(F_\sigma(L^{(0)}) - F_\sigma(L^{(K)}),$$

which implies that

$$\min_{k=0,\ldots,K} \langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle^2 \leq \frac{C(F_\sigma(L^{(0)}) - F_\sigma^*)}{K},$$

as desired.

REMARK 8.1 Theorem 8.3 tells us that $|\langle \nabla F_\sigma(L^{(k)}), D^{(k)} \rangle| \to 0$ with rate $O(1/\sqrt{K})$. This together with Theorem 8.2 yields the result that Algorithm 8.1 with the Armijo search rule finds a critical point of (8.4) with convergence rate $O(1/\sqrt{K})$.

8.5 Numerical Experiments

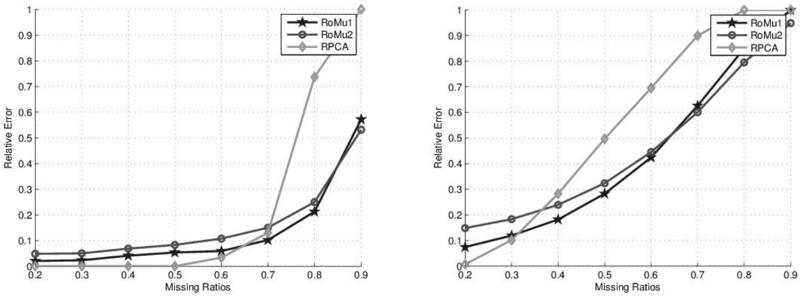
In this section, we present some numerical experiments on synthetic data as well as real data. All the numerical computations are conducted on an Intel i7-3770 CPU desktop computer with 16 GB of RAM. The supporting software is MATLAB R2013a.

8.5.1 Algorithms Setting

We mainly compare the proposed algorithms with RPCA [2], which solves convex optimization problems and employs the LAD loss to penalize the noise or outliers. Algorithm 8.1 with strategy (8.10) for solving the rank constrained problem (8.3) is denoted by RoMu1 for short, while Algorithm 8.1 with strategy (8.12) for solving the nuclear norm constrained problem (8.4) is denoted as RoMu2 for short. The max iteration for all the methods is 400. The stopping criterion for all the methods is that the difference between the current trial and the previous trial is less than a threshold, where the threshold is set to $\epsilon = 10^{-4}$. Parameters are tuned via 5-fold cross validation. All the results are averaged over ten instances.

8.5.2 Synthetic Data

We randomly generate some matrices of size 500 × 500, and then truncate them to be low rank, where we consider rank $\in \{10, 50\}$. Next, 20% of the entries are contaminated by outliers in $[-10, 10]$. Finally, some entries are randomly missing, where the missing ratio (MR for short) varies between $\{0.2, 0.3, 0.5, 0.7, 0.9\}$. The relative error relerr $= \|X^* - B\|_F / \|B\|_F$ will be used to evaluate performances of the algorithms. Results are reported in Figure 8.1, where the blue curve represents the performance of RoMu1, the red one is that of RoMu2, and the green one stands for RPCA. First we look at Figure 8.1.a, which is the case that rank = 10. We observe that when the MR value is less than 0.65, RPCA is better than our methods. However, its performance decreases sharply as the MR value increases, and it achieves 1 when the MR value is 0.9. On the other hand, our method is more stable. Comparing between RoMu1 and RoMu2, we see that RoMu1 is better than RoMu2 when the MR value is less than 0.85. The reason might be that the weights chosen by RoMu1 are more greedy, which leads to a sufficient decrease of the cost function, whereas for RoMu2, choosing the weights has restrictions in $(0, 1)$, as shown in (8.12). We then consider Figure 8.1.b ,



a) rank = 50

b) rank = 50

FIGURE 8.1 Performance comparisons of RoMu1, RoMu2 and RPCA [2] on synthetic data $(500 \times 500), \text{rank} = \{10, 50\}$. The x-axis is the MR value; the y-axis stands for the relative error.

i.e., the case rank = 50. One first notices that all the methods perform worse than the case rank = 10. The reason is evident, since low-rank methods perform better when the rank is not high. One then observes that our methods outperform RPCA when the MR value is larger than 0.4. In summary, our methods are more stable when there are missing values and outliers.

We also report the computational time of the three methods in Table 8.1. From the table, we observe that RoMu1 performs the fastest, followed by RoMu2. This observation confirms the efficiency of our methods. Particularly, an appealing feature of our methods is that as the MR value increases, the computational time decreases, which is due to the fact that the simple structure of Algorithm 8.1 can utilize the sparsity of the matrix: When computing the rank-one matrix, only sparse matrix-vector multiplications are needed; computing the weights is also fast as it can be given by the inner product of sparse matrices, as shown in (8.10).

TABLE 8.1 Efficiency comparisons of RoMu1, RoMu2, and RPCA [2] on synthetic data $(500 \times 500), \text{rank} \in \{10, 50\}$.

MR (%)		20	30	40	50	60	70	80	90
rk=10	RoMu1	1.49	1.32	1.15	0.89	0.71	0.56	0.38	0.20
	RoMu2	6.30	5.54	4.77	3.59	2.90	2.26	1.58	0.94
	RPCA [2]	3.77	4.40	5.06	9.26	41.00	40.91	40.83	40.74
rk=50	RoMu1	6.04	5.31	4.57	3.47	2.78	2.14	1.51	0.88
	RoMu2	6.28	5.59	4.82	3.64	2.97	2.26	1.58	0.92
	RPCA [2]	40.93	41.29	41.48	40.64	31.13	36.08	40.59	40.80

8.5.3 Real Data

Image/Video recovery

Gray images can be seen as matrices, while a video can also be treated as a matrix by vectorizing every frame into a vector and arranging them one by one. In real-world applications, due to some reasons, a large fraction of entries of image/video may be missing and may be contaminated by noise or outliers. The goal of this section is to recover such kinds of images/videos. The following datasets are selected: Facade (493 × 517), Hyperspectral images (50430 × 96), Brain MRI (39277 × 181), Incisix (16384 × 166), and Ocean (17920 × 32). The first dataset is a gray image, while the last four can be seen as videos. Then 20% of the entries are contaminated by outliers in $[-256, 256]$. Last, some entries are randomly missing, where the missing ratio varies between $\{0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$. The relative error will also be used to evaluate performances of the algorithms.

The performances are reported in Table 8.2. From the table, we can observe that in most cases, RoMu2 outperforms others, followed by RoMu1. RPCA only performs better than our methods in the image Facade, which has a relatively small size compared to other datasets. This observation implies that our methods may be suitable for large-scale problems. In Figure 8.2 and Figure 8.3, we present part of the recovery results of RoMu1, RoMu2, and RPCA on the datasets Facade and Incisix to intuitively illustrate their performances. On Facade, although Table 8.2 shows that RPCA performs better, from Figure 8.2 it seems that to penalize the outliers, RPCA has to remove more details from the image, while our methods retain more details. Figure 8.3 shows that RPCA cannot correctly recover the dataset. On the other hand, the efficiency is also reported in Table 8.2, where our methods are again much faster than RPCA.

Yale face

As with [2], the goal of this application is to remove shadows from faces, where the datasets are chosen from the extended Yale face database B. We choose two datasets, each of which consists of 64 faces of a person under 64 illumination conditions, and the size of each image is 192 × 168. We do not add outliers to the datasets, because the shadows in the faces can be regarded as noise or outliers. There do not have to be missing values as well. To show the results, from each dataset we select four images, which are shown in Figure 8.4 and Figure 8.5. From the results, we can observe that all the three methods can remove shadows, while from the second row of Figure 8.4, it seems that our methods perform slightly better than RPCA, as the left eye of the person recovered by RPCA cannot be seen clearly. The first row of Figure 8.5 also indicates that our methods perform better, as the lines in the face have been totally removed by our methods. Finally, we find that empirically we only need the linear combination of around ten rank-one matrices to yield the recovery results, which means that our methods can be stopped within ten iterations, implying that our methods are very efficient.

8.6 Conclusion

In this chapter, we proposed a nonconvex approach for robust matrix completion. Along with the approach, we presented two solution methods, one for solving the rank constrained model, the other one for solving the nuclear norm constrained model. The convergence of the algorithms were verified; particularly, for the second algorithm, we proved that it converges to a stationary point, and showed the iteration complexity, which is $O(1/\sqrt{K})$. Finally, numerical experiments show that the proposed models and algorithms are comparable and better than the state-of-the-art method.

Acknowledgment

EU: 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 AdG A-DATADRIVE-B (290923). This article 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, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants; Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor-based data similarity); PhD/Postdoc grants; IWT: projects: SBO POM (100031); PhD/Postdoc grants; iMinds Medical Information Technologies SBO 2014; Belgian Federal

Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012–2017). Johan Suykens is a professor at KU Leuven, Belgium.

TABLE 8.2 Performances and efficiency comparisons of RoMu1, RoMu2, and RPCA [2] on some real datasets.

		RoMu1		RoMu2		RPCA [2]	
Dataset		relerr	time	relerr	time	relerr	time
Facade (493 × 517)	0.50	9.84E-02	2.57	9.60E-02	3.54	**3.55E-02**	42.02
	0.60	1.14E-01	2.09	1.09E-01	2.96	**4.17E-02**	42.14
	0.70	1.45E-01	1.76	1.30E-01	2.36	**5.22E-02**	42.12
	0.80	1.73E-01	1.16	1.48E-01	1.56	**8.14E-02**	41.46
	0.90	2.63E-01	0.67	**1.83E-01**	0.93	2.15E-01	40.89
	0.95	3.98E-01	0.43	**2.21E-01**	0.60	1.00E+00	2.55
Hyperspectral (50430 × 96)	0.50	2.18E-02	52.40	1.66E-02	57.19	**1.17E-02**	141.14
	0.60	3.12E-02	43.70	2.34E-02	50.60	**2.03E-02**	143.09
	0.70	4.76E-02	39.76	**3.85E-02**	34.61	3.91E-02	148.80
	0.80	8.42E-02	27.37	**6.22E-02**	26.24	1.18E-01	156.54
	0.90	1.95E-01	15.83	**1.53E-01**	10.41	1.00E+00	14.73
	0.95	5.37E-01	8.14	**3.39E-01**	10.92	1.00E+00	5.72
Brain (39277 × 181)	0.50	4.44E-02	58.86	**3.77E-02**	86.69	3.03E-01	264.82
	0.60	6.43E-02	49.22	**5.56E-02**	72.10	3.85E-01	265.54
	0.70	9.79E-02	38.58	**8.74E-02**	58.28	5.19E-01	260.55
	0.80	1.61E-01	28.59	**1.46E-01**	42.47	8.48E-01	262.08
	0.90	3.12E-01	14.78	**2.79E-01**	23.69	1.00E+00	256.96
	0.95	4.93E-01	7.19	**4.41E-01**	12.88	1.00E+00	237.04
Incisix (16384 × 166)	0.50	2.48E-01	37.94	**2.22E-01**	52.76	2.67E-01	99.42
	0.60	2.80E-01	31.87	**2.58E-01**	44.52	2.99E-01	101.58
	0.70	3.19E-01	23.60	**3.01E-01**	33.97	3.60E-01	100.22
	0.80	3.88E-01	15.84	**3.68E-01**	23.38	4.60E-01	98.81
	0.90	5.00E-01	8.55	**4.76E-01**	12.75	1.00E+00	30.83
	0.95	6.02E-01	4.64	**5.74E-01**	7.42	1.00E+00	6.10
Ocean (17920 × 32)	0.50	1.02E-01	7.39	**9.82E-02**	11.39	1.34E-01	17.14
	0.60	1.21E-01	5.70	**1.18E-01**	7.77	1.87E-01	17.01
	0.70	1.51E-01	4.51	**1.43E-01**	4.62	3.02E-01	16.74
	0.80	2.12E-01	3.03	**1.86E-01**	5.34	1.00E+00	13.55
	0.90	4.06E-01	1.82	**3.54E-01**	3.72	1.00E+00	6.87
	0.95	6.52E-01	1.15	**5.97E-01**	1.84	1.00E+00	3.43

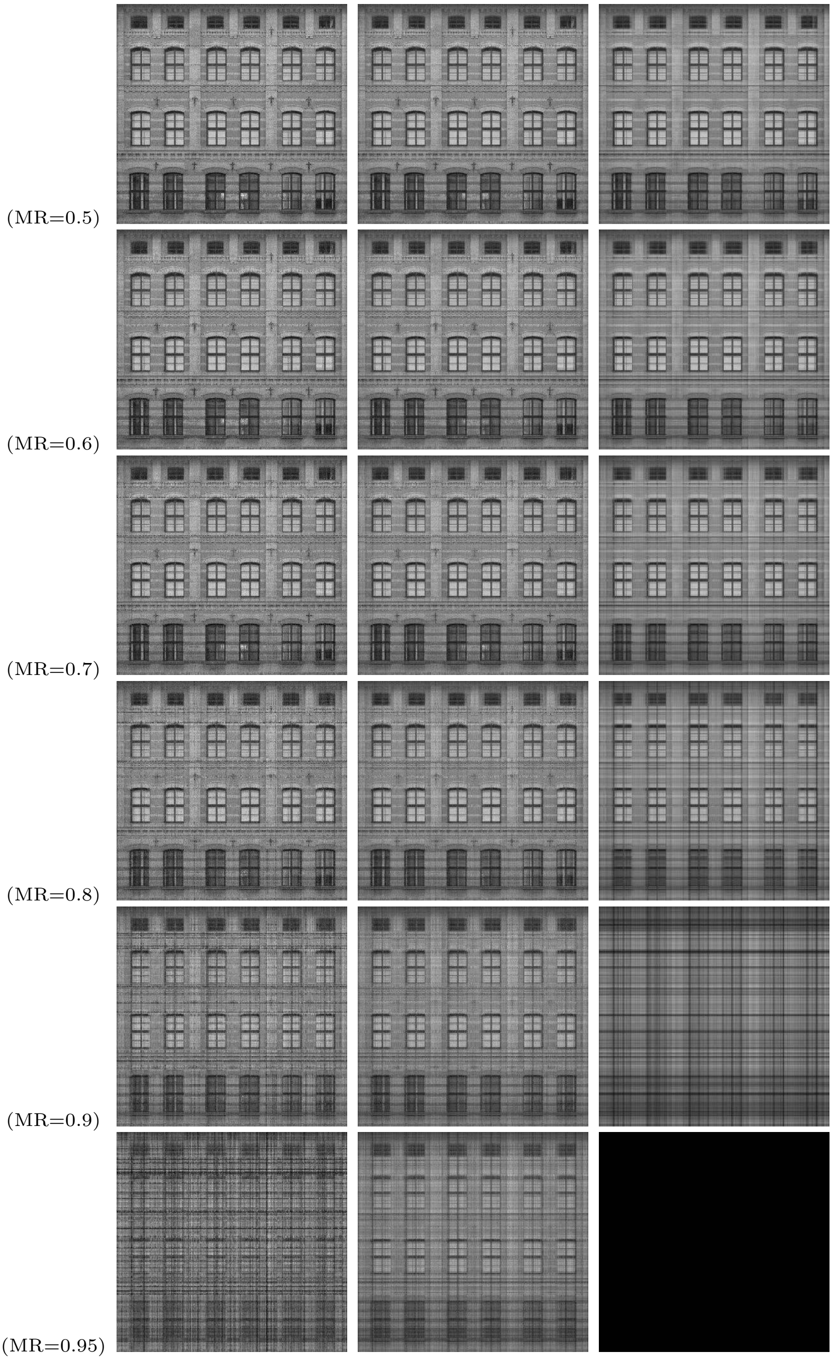


FIGURE 8.2 Comparison of RoMu1 (Column 1), RoMu2 (Column 2), and RPCA [2] (Column 3) on recovering the gray image Facade.

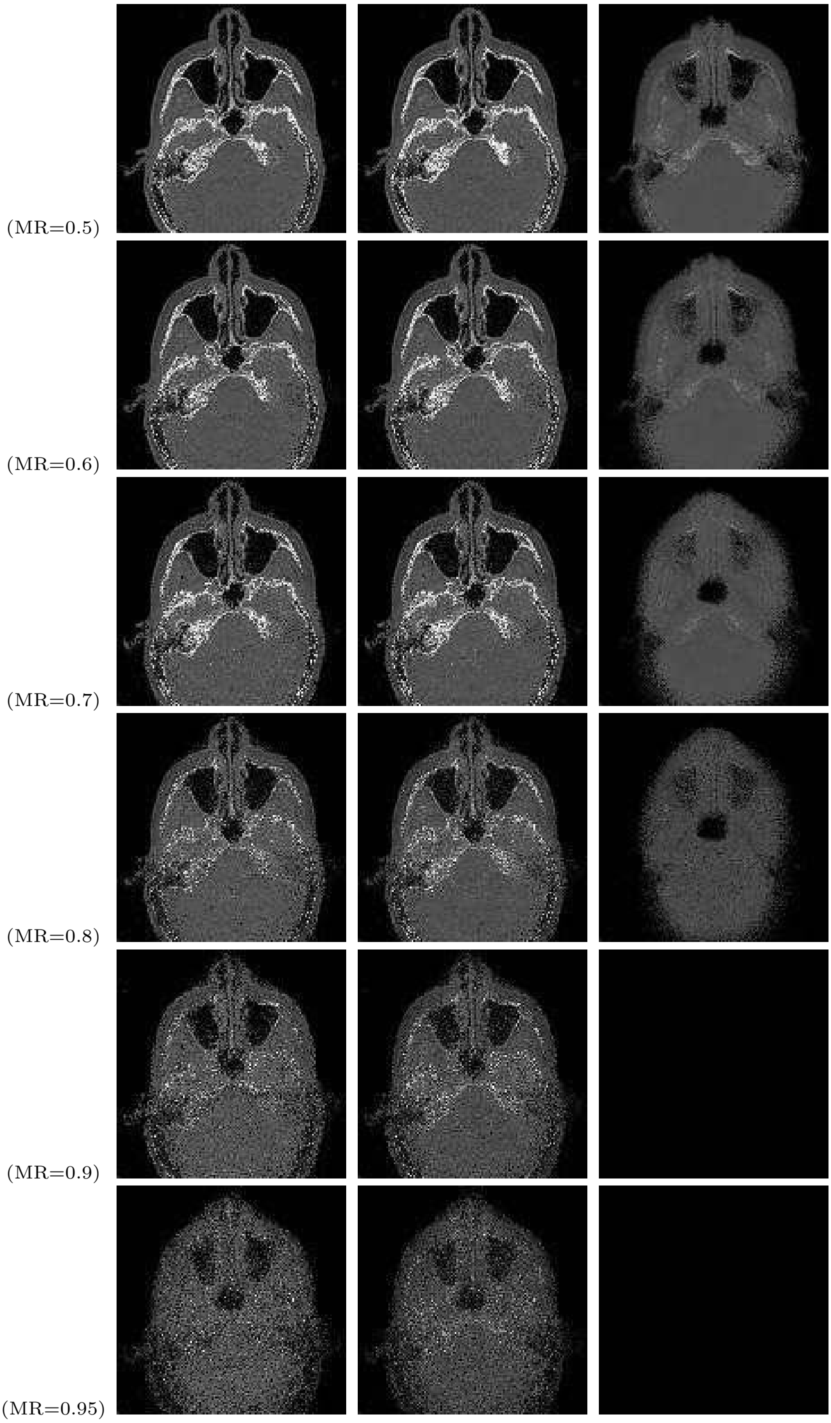
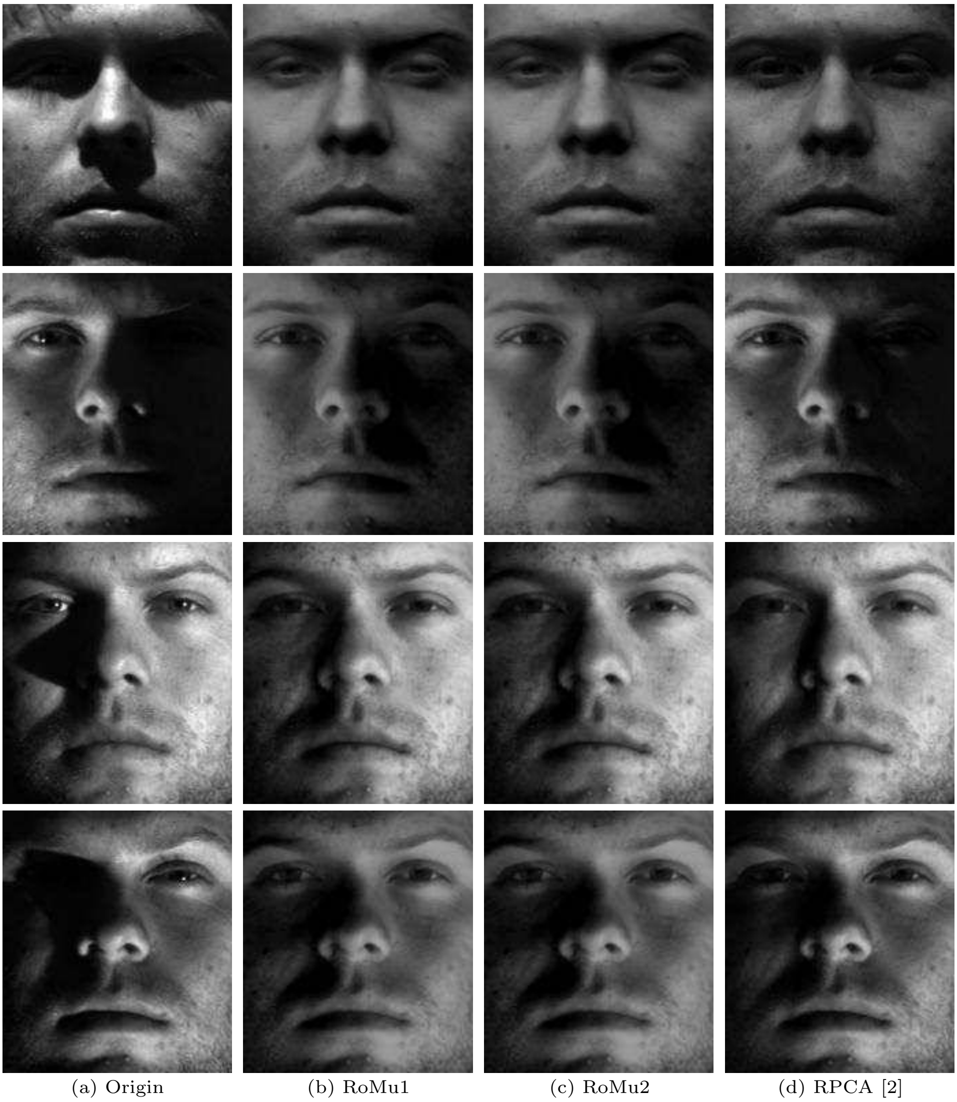


FIGURE 8.3 Comparison of RoMu1 (Column 1), RoMu2 (Column 2), and RPCA [2] (Column 3) on recovering one slide of the Incisix dataset.



(a) Origin (b) RoMu1 (c) RoMu2 (d) RPCA [2]

FIGURE 8.4 Comparison of RoMu1 (Column 1), RoMu2 (Column 2), and RPCA [2] (Column 3) on removing shadows from faces.

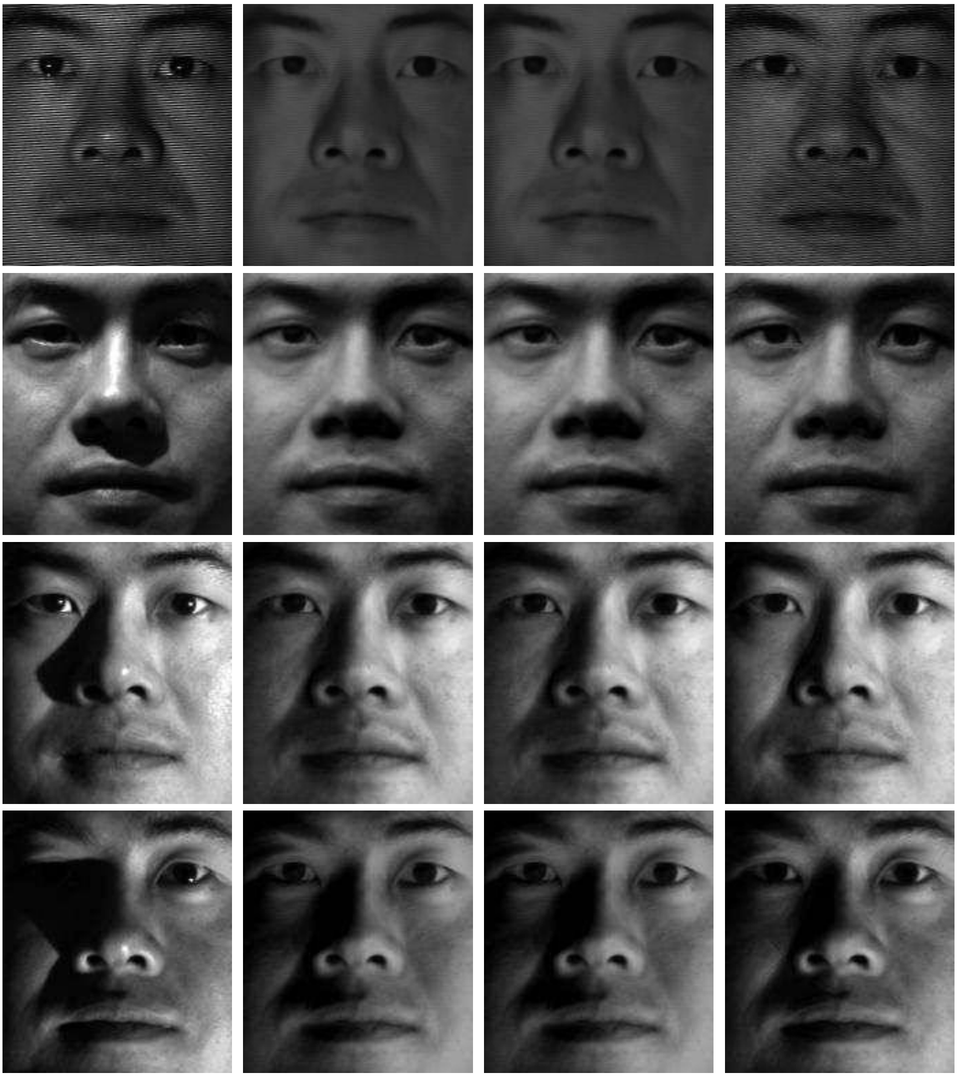


FIGURE 8.5 Comparison of RoMu1 (Column 1), RoMu2 (Column 2), and RPCA [2] (Column 3) on removing shadows from faces.

References

1. M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Annual conference on computer graphics and interactive techniques*, pages 417–424, 2000.
2. E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11, 2011.
3. E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.
4. Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi. Robust matrix completion with corrupted columns. *arXiv preprint arXiv:1102.2254*, 2011.
5. M. Dudik, Z. Harchaoui, and J. Malick. Lifted coordinate descent for learning with trace-norm regularization. In *International Conference on Artificial Intelligence and Statistics, AISTATS 2012*, volume 22, pages 327–336, 2012.
6. M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference, ACC 2001*, volume 6, pages 4734–4739, 2001.
7. Y. Feng, X. Huang, L. Shi, Y. Yang, and J. Suykens. Learning with the maximum correntropy criterion induced losses for regression. *Journal of Machine Learning Research*, 2015.
8. M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.
9. D. Goldfarb and S. Ma. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Foundations of Computational Mathematics*, 11(2):183–210, 2011.
10. T. Hastie. Matrix completion and large-scale SVD computations. *Slides*, May 2012.
11. R. He, W. Zheng, T. Tan, and Z. Sun. Half-quadratic based iterative minimization for robust sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):261–275, 2013.
12. P. Holland and R. Welsch. Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6:813–827, 1977.
13. P. Huber. *Robust statistics*. Springer, 2011.
14. M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. *International Conference on Machine Learning, ICML 2013*, pages 427–435, 2013.
15. M. Jaggi and M. Sulovsk. A simple algorithm for nuclear norm regularized problems. *International Conference on Machine Learning, ICML 2010*, pages 471–478, 2010.
16. P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In *NIPS*, volume 23, pages 937–945, 2010.
17. H. Ji, C. Liu, Z. Shen, and Y. Xu. Robust video denoising using low rank matrix completion. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010*, pages 1791–1798. IEEE, 2010.
18. S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Annual International Conference on Machine Learning*, pages 457–464. ACM, 2009.
19. V. Koltchinskii, K. Lounici, and A. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.
20. N. Komodakis. Image completion using global optimization. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2006*, volume 1, pages 442–452. IEEE, 2006.
21. W. Liu, P. Pokharel, and J. Principe. Correntropy: properties and applications in non-Gaussian signal processing. *IEEE Transactions on Signal Processing*, 55(11):5286–5298, 2007.
22. 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.

23. S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.
24. K. Mohan and M. Fazel. Iterative reweighted algorithms for matrix rank minimization. *Journal of Machine Learning Research*, 13:3441–3473, 2012.
25. F. Nie, H. Wang, X. Cai, H. Huang, and C. Ding. Robust matrix completion via joint schatten p-norm and l_p-norm minimization. In *IEEE International Conference on Data Mining, ICDM 2012*, pages 566–574. IEEE, 2012.
26. Netflix prize website. http://www.netflixprize.com. *Netflix*, 2009.
27. B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.
28. A. Rohde and A. Tsybakov. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887–930, 2011.
29. M. Signoretto, Q. Dinh, L. De Lathauwer, and J.A.K Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning*, 94(3):303–351, 2014.
30. N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *International Conference on Machine Learning, ICML 2003*, volume 3, pages 72–727, 2003.
31. A. Tewari, P. Ravikumar, and I. Dhillon. Greedy algorithms for structurally constrained high-dimensional problems. In *Advances in Neural Information Processing Systems*, pages 882–890, 2011.
32. J. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.
33. X. Wang, Y. Jiang, M. Huang, and H. Zhang. Robust variable selection with exponential squared loss. *Journal of the American Statistical Association*, 108(502):632–643, 2013.
34. Z. Wang, M. Lai, Z. Lu, and J. Ye. Orthogonal rank-one matrix pursuit for low rank matrix completion. *SIAM Journal on Scientific Computing*, 37:A488A514, 2015.
35. J. Wright, A. Ganesh, S. Raor, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.
36. H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.
37. Y. Yang, Y. Feng, and J. Suykens. A nonconvex relaxation approach to robust matrix completion. *Internal Report 14-61, ESAT-SISTA, KU Leuven*, 2014.
38. Y. Yang, Y. Feng, and J. Suykens. Robust low rank tensor recovery with regularized redescending M-estimator. *Internal Report 14-97, ESAT-SISTA, KU Leuven*, 2014.
39. X. Zhang, D. Schuurmans, and Y. Yu. Accelerated training for matrix-norm regularization: A boosting approach. In *Advances in Neural Information Processing Systems*, pages 2906–2914, 2012.