Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Nonconvex Relaxation Approaches to Robust Matrix Recovery

Shusen Wang and **Dehua Liu** and **Zhihua Zhang**
College of Computer Science & Technology
Zhejiang University
Hangzhou, China 310027
{wss,dehua,zhzhang}@zju.edu.cn

Abstract

Motivated by the recent developments of nonconvex penalties in sparsity modeling, we propose a nonconvex optimization model for handing the low-rank matrix recovery problem. Different from the famous robust principal component analysis (RPCA), we suggest recovering low-rank and sparse matrices via a nonconvex loss function and a nonconvex penalty. The advantage of the nonconvex approach lies in its stronger robustness. To solve the model, we devise a majorization-minimization augmented Lagrange multiplier (MM-ALM) algorithm which finds the local optimal solutions of the proposed nonconvex model. We also provide an efficient strategy to speedup MM-ALM, which makes the running time comparable with the state-of-the-art algorithm of solving RPCA. Finally, empirical results demonstrate the superiority of our nonconvex approach over RPCA in terms of matrix recovery accuracy.

1 Introduction

In many computer vision and machine learning problems, a data matrix can be represented as a low-rank component plus a sparse component. For example, video surveillance is the superposition of low-rank background and sparse foreground; corrupted images can be approximated by low-rank images plus sparse data noises; collaborative filtering problems can be solved by recovering a low-rank rating matrix and a sparse noise matrix from a partially observed data matrix. Therefore, it is of great interest to recover the low-rank component and the sparse component from the observed data matrix. However, solving matrix recovery problems is not trivial.

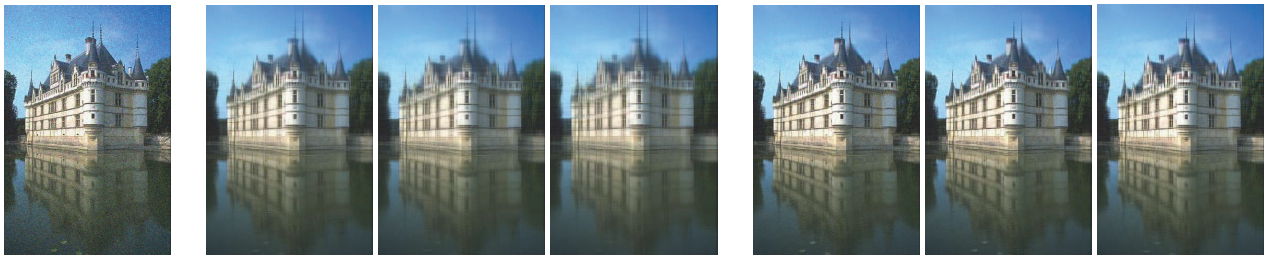
An intuitive idea is to formulate matrix recovery as a minimization problem with ℓ_0-norm (i.e., the number of nonzero entries) loss and matrix rank penalty. Unfortunately, this optimization problem is NP-hard. An important alternative, well-known as robust principal component analysis (RPCA) [Candès *et al.*, 2011; Wright *et al.*, 2009], relaxes this problem into minimizing matrix ℓ_1-norm loss and nuclear-norm penalty. This alternative is tractable because the resulting optimization problem is convex. When the rank of the underlying low-rank matrix is sufficiently low, RPCA can exactly recover the low-rank and sparse matrices with high probability [Candès *et al.*, 2011].

RPCA has become an effective tool for computer vision applications. Beyond decomposing a data matrix into a low-rank component and a sparse component, RPCA can also be reformulated to handle a variety of computer vision problems, and it achieved state of the art results. For example, Liu *et al.* extended RPCA to solve subspace clustering problems; Peng *et al.* reformulated RPCA to align images; Zhang *et al.* introduced a variant of RPCA to camera calibration problems; Cheng *et al.* resorted to RPCA for solving image segmentation problems; Wang and Zhang employed RPCA to solve colorization problems.

However, RPCA has two major limitations. First, the rank of the underlying low-rank component is not sufficiently low in many applications, which violates the assumptions of RPCA. For example, natural images cannot be effectively approximated by matrices with very low rank, so RPCA may not be an effective tool for image inpainting. Under such a circumstance RPCA has relatively low performance as verified by our experiments.

Second, RPCA does not have good performance in robust dimensionality reduction. As was mentioned in [Wright *et al.*, 2009; Candès *et al.*, 2011], RPCA seeks to overcome the brittleness of the conventional PCA when the data are grossly corrupted. Unfortunately, the performance of RPCA in dimensionality reduction has neither theoretical guarantee nor experiment verification. Unlike truncated singular value decomposition (SVD) which discards only the small singular values, RPCA shrinks all the singular values equally. As is discussed in the following paragraphs, such behavior of RPCA hurts the "good" singular values and thereby leads to biased results. The empirical results in Figure 1(b) 1(c) 1(d) verify this point.

Since RPCA is an important tool in computer vision and machine learning, fundamental improvements in RPCA methodology can significantly contribute to many real world applications. This work seeks to better recover the low-rank and sparse matrices via a more robust and less biased formulation. Our method is motivated by the nonconvex sparsity-inducing penalties in [Fan and Li, 2001; Zhang, 2010a]. Figure 1 demonstrates that our method fulfills matrix recovery and dimensionality reduction much more effectively than RPCA.



(a) observed (b) rank 200 (c) rank 100 (d) rank 50 (e) rank 200 (f) rank 100 (g) rank 50

Figure 1: Results on matrix dimenionality deduction. 1(a) a color image with 50% entries added with Gaussian noises i.i.d. from $N(0, 0.1^2)$; 1(b) 1(c) 1(d) are the low-rank components recovered from 1(a) by RPCA, each of which is of rank 200, 100, and 50, respectively. 1(e) 1(f) 1(g) are recovered by our method (NRMR).

Fan and Li pointed out that the ℓ_1-norm penalty over-penalizes large entries of vectors. Moreover, they proposed three criteria for good penalty functions: unbiasedness, sparsity and continuity at the origin. The ℓ_1-norm satisfies both sparsity and continuity, but it is biased. Based on these three properties, Fan and Li proposed a new penalty function called the smoothly clipped absolute deviation penalty (SCAD). Recently, Zhang proved that a so-called minmax concave plus (MCP) penalty also possesses the three properties and achieves better performance than SCAD. Both SCAD and MCP are nonconvex and nearly unbiased. Extensive experiments in [Breheny and Huang, 2011; Fan and Li, 2001; Zhang, 2010a; Shi *et al.*, 2011; Zhang *et al.*, 2012; Zhang and Tu, 2012] have demonstrated the superiority of SCAD and MCP over the ℓ_1-norm penalty.

It is a well known fact that the matrix rank is the number of nonzero singular values of a matrix, and the nuclear norm is the sum of all the singular values of a matrix. We can thereby relate the matrix rank to the ℓ_0-norm of a vector; in the same way we relate the matrix nuclear norm to the ℓ_1-norm of a vector. On one hand, this relation implies that the nuclear norm over-penalizes large singular values in the same way that the ℓ_1-norm over-penalizes large entries; in other words, the nuclear norm results in a biased estimator as well as the ℓ_1-norm does. On the other hand, this relation encourages us to devise a new nonconvex loss function or penalty for dealing with bias in the rank-minimization problems [Mazumder *et al.*, 2010].

In particular, we consider the natural extension of MCP functions on matrices. Moreover, we also propose and study a matrix γ-norm as a nonconvex relaxation of the matrix rank. Roughly speaking, the matrix γ-norm is the minimax concave (MCP) function of the matrix singular values. The γ-norm is characterized by a positive factor (say γ), and it is tighter to the matrix rank than the nuclear norm is. Moreover, its limit at $\gamma \to \infty$ is the nuclear norm.

We apply the matrix MCP function as a nonconvex relaxation of the ℓ_0-norm loss and the matrix γ-norm as a nonconvex relaxation of the matrix rank penalty to handle robust matrix recovery. Our work offers the following contributions.

- We introduce the nonconvex sparsity-inducing penalty MCP function to the rank-minimization problem.
- We develop the *Nonconvex relaxation based Robust Matrix Recovery* (NRMR) model for the matrix recovery problem. NRMR alleviates over-penalization on "good" variables and thus better recovers the low-rank and sparse matrices.
- We devise a *majorization-minimization augmented Lagrange multiplier algorithm (MM-ALM)* to solve the NRMR model. We also propose a speedup strategy which solves NRMR as efficiently as the ALM algorithm solves RPCA.

The remainder of this paper is organized as follows. In Section 3 we formulate the matrix recovery problem. In Section 4 we apply MCP function to rank-minimization problems and obtain a called γ-norm with which we formulate the NRMR model. In Section 5 we devise an algorithm called MM-ALM to solve the NRMR model. In Section 6 we empirically evaluate the performance of NRMR mainly in comparison with RPCA.

2 Notations

The notations in this paper are defined as follows. For a matrix $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{m\times n}$, let $\|\mathbf{A}\|_0$ be the ℓ_0-norm (i.e., the number of nonzero entries of $\mathbf{A}$), $\|\mathbf{A}\|_1 = \sum_{i,j} |A_{ij}|$ be the ℓ_1-norm, $\|\mathbf{A}\|_F = (\sum_{i,j} A_{ij}^2)^{1/2} = (\sum_{i=1}^r \sigma_i^2(\mathbf{A}))^{1/2}$ be the Frobenius norm, and $\|\mathbf{A}\|_* = \sum_{i=1}^r \sigma_i(\mathbf{A})$ be the nuclear norm, where $\sigma_i(\mathbf{A})$ be the i-th largest singular value of $\mathbf{A}$ and $r = \min\{m, n\}$. Finally, let $\mathbf{I}$ denote the identity matrix with appropriate size, $\mathbf{1}_n = [1, 1, \cdots, 1]^T \in \mathbb{R}^n$ denote the all-one vector.

3 Problem Formulation

Given an $m\times n$ matrix $\mathbf{D}$, we are concerned with the problem of recovering a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ such that $\mathbf{L} + \mathbf{S} = \mathbf{D}$. It can be formulated as the following optimization problem:

$$\min_{\mathbf{L},\mathbf{S}} \ \mathrm{rank}(\mathbf{L}) + \lambda\|\mathbf{S}\|_0; \qquad \text{s.t. } \mathbf{L} + \mathbf{S} = \mathbf{D}. \tag{1}$$

Since the problem in (1) is NP-hard, it is intractable in practical applications. Based on the fact that $\|\mathbf{L}\|_*$ and $\|\mathbf{S}\|_1$ are

the best convex approximations of $\mathrm{rank}(\mathbf{L})$ and $\|\mathbf{S}\|_0$, respectively, Candès *et al.* proposed the RPCA model, which is defined by

$$\min_{\mathbf{L},\mathbf{S}} \ \|\mathbf{L}\|_* + \lambda\|\mathbf{S}\|_1; \qquad \text{s.t. } \mathbf{L}+\mathbf{S}=\mathbf{D}. \tag{2}$$

An efficient algorithm of solving RPCA is an augmented Lagrange multiplier algorithm (ALM), which was studied in [Lin *et al.*, 2009]. Like other algorithms for a penalized nuclear norm minimization problem, the ALM algorithm is based on the singular value shrinkage operator [Cai *et al.*, 2010].

4 Methodology

In Section 4.1 we introduce a nonconvex function that called the matrix MCP norm. In Section 4.2 we apply MCP to the rank minimization problem and obtained a nonconvex low-rank-inducing penalty called the *γ-norm*. With the matrix MCP norm and the γ-norm, in Section 4.3 we formulate the Nonconvex relaxation based Robust Matrix Recovery (NRMR) model. In Section 4.4 we discuss how to tune the parameters of NRMR.

4.1 The Matrix MCP Norm

Given a vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T \in \mathbb{R}^p$, $\lambda > 0$, and $\gamma > 1$, the MCP function [Zhang, 2010a] is defined by

$$M_{\lambda,\gamma}(\boldsymbol{\beta}) \ = \ \sum_{i=1}^{p} \psi_{\lambda,\gamma}(\beta_i),$$

where

$$\psi_{\lambda,\gamma}(t) = \lambda \int_0^t \Big[1 - x/(\gamma\lambda)\Big]_+ dx = \begin{cases} \gamma\lambda^2/2 & \text{if } |t| \geq \gamma\lambda, \\ \lambda|t| - \frac{t^2}{2\gamma} & \text{otherwise.} \end{cases}$$

Here $(z)_+ = \max\{z, 0\}$. The matrix MCP norm is naturally defined by

$$M_{\lambda,\gamma}(\mathbf{A}) = \sum_{i,j} \psi_{\lambda,\gamma}(A_{i,j}). \tag{3}$$

Here and later, we denote $\psi_\gamma(t) = \psi_{1,\gamma}(t)$ and $M_\gamma(\mathbf{A}) = M_{1,\gamma}(\mathbf{A})$ for notational simplicity. The function $M_\gamma(\cdot)$ is continuous w.r.t. γ. When $\gamma \to \infty$, $\psi_\gamma(t) \to |t|$ and $M_\gamma(\cdot)$ becomes the ℓ_1-norm; when $\gamma \to 1$, it gives rise to a hard threshold operator corresponding to the ℓ_0 norm. Thus, $M_\gamma(\cdot)$ bridges the ℓ_1 and ℓ_0 norm. The matrix MCP norm enjoys several properties as follows.

Proposition 1. *The matrix MCP norm defined in (3) satisfies the following properties:*

(1) $M_\gamma(\mathbf{A}) \geq 0$, *with equality iff* $\mathbf{A} = \mathbf{0}$;

(2) $M_\gamma(\mathbf{A})$ *is concave w.r.t.* $|\mathbf{A}|$, *where* $|\mathbf{A}| = \big[|A_{ij}|\big]$;

(3) $M_\gamma(\mathbf{A})$ *is increasing in* γ, $M_\gamma(\mathbf{A}) \leq \|\mathbf{A}\|_1$, *and* $\lim_{\gamma\to\infty} M_\gamma(\mathbf{A}) = \|\mathbf{A}\|_1$.

It is worth pointing out that like the matrix rank and the ℓ_0-norm, the matrix MCP norm and the following matrix γ-norm are not real norms, because they are nonconvex and do not satisfy the triangle inequality of a norm.

4.2 The Matrix γ-Norm

Let $\boldsymbol{\sigma}(\mathbf{A}) = \big(\sigma_1(\mathbf{A}), \ldots, \sigma_r(\mathbf{A})\big)^T$ be a function from $\mathbb{R}^{m\times n}$ to $\mathbb{R}^r_+$ where $r = \min\{m, n\}$. Interestingly, we have that $\mathrm{rank}(\mathbf{A}) = \|\boldsymbol{\sigma}(\mathbf{A})\|_0$, $\|\mathbf{A}\|_* = \|\boldsymbol{\sigma}(\mathbf{A})\|_1$, and $\|\mathbf{A}\|_F = \|\boldsymbol{\sigma}(\mathbf{A})\|_2$. This can help us better understand the connection between the vector ℓ_1-norm and the matrix nuclear norm.

Fan and Li (2001) and Zhang (2010) proved that the ℓ_1-norm over-penalizes large components, leading to a biased estimator. Moreover, they showed that a nonconvex penalty such as SCAD and MCP yields an asymptotically unbiased estimator. This motivates us to apply the MCP function to the rank minimization problem to better approximate the matrix rank. We refer to this nonconvex function as the *matrix γ-norm* and denote it by $\|\mathbf{A}\|_\gamma$.

Particularly, the γ-norm of an $m\times n$ matrix $\mathbf{A}$ is defined by

$$\begin{aligned} \|\mathbf{A}\|_\gamma \ &:= \ \sum_{i=1}^{r} \int_0^{\sigma_i(\mathbf{A})} \Big(1 - \frac{u}{\gamma}\Big)_+ du \\ &= \ \sum_{i=1}^{r} \psi_{1,\gamma}\Big(\sigma_i(\mathbf{A})\Big) \ = \ M_\gamma\Big(\boldsymbol{\sigma}(\mathbf{A})\Big), \quad \gamma > 1. \end{aligned}$$

Clearly, $\|\mathbf{A}\|_\gamma$ is nonconvex w.r.t. $\mathbf{A}$. Moreover, it is easy to verify that it possesses the following properties.

Proposition 2. *For $\gamma \in (1, \infty)$, then*

(1) $\|\mathbf{A}\|_\gamma \geq 0$, *with equality iff* $\mathbf{A} = \mathbf{0}$;

(2) $\|\mathbf{A}\|_\gamma$ *is an increasing function of* γ, $\|\mathbf{A}\|_\gamma \leq \|\mathbf{A}\|_*$, *and* $\lim_{\gamma\to+\infty} = \|\mathbf{A}\|_*$;

(3) $\|\mathbf{A}\|_\gamma$ *is unitarily invariant; that is,* $\|\mathbf{UAV}\|_\gamma = \|\mathbf{A}\|_\gamma$ *whenever* $\mathbf{U}$ $(m\times m)$ *and* $\mathbf{V}$ $(n\times n)$ *are orthonormal.*

4.3 Formulation of NRMR

We use the matrix MCP norm as a surrogate of the ℓ_0-norm and the γ-norm as a surrogate of the matrix rank, simultaneously. Accordingly, we have

$$\min_{\mathbf{L},\mathbf{S}} \ f(\mathbf{L},\mathbf{S}) = \|\mathbf{L}\|_{\gamma_1} + \lambda M_{\gamma_2}(\mathbf{S}); \ \text{ s.t. } \mathbf{L}+\mathbf{S}=\mathbf{D}, \tag{4}$$

where $\|\mathbf{L}\|_{\gamma_1}$ and $M_{\gamma_2}(\mathbf{S})$ are characterized by parameters γ_1 and γ_2, respectively. Since our model is based on the idea of nonconvex relaxation, we call it *Nonconvex relaxation based Robust Matrix Recovery* (NRMR). From the properties of the matrix MCP norm and the γ-norm, it is clear that NRMR is a tighter approximation to Problem 1 than RPCA.

4.4 The Tuning Parameters

Finally, compared with RPCA which has only one tuning parameter λ, NRMR appears to be more complicated. It is therefore useful to discuss how to tune the parameters λ, γ_1 and γ_2. The parameter λ is the most significant among the three parameters and should be carefully chosen. Similar to RPCA, λ should be selected from the neighborhood of $1/\sqrt{\max\{m,n\}}$. Smaller λ gives rise to lower rank of $\mathbf{L}^*$. As for γ_1 and γ_2, experiments show that the results are insensitive to them in a large range, but γ_1 and γ_2 should be neither

too large nor too small. When $\gamma \to +\infty$, NRMR simply becomes RPCA; when $\gamma \to 1$, the local convergence results in poor solutions. Empirically speaking, γ_1 and γ_2 should be set to be small real numbers strictly greater than 1. We fix $\gamma_1 = \gamma_2 = 4$ in our experiments.

5 Algorithms

This section is organized as follows: in Sections 5.1 we discuss the generalized singular value shrinkage operator for the γ-norm; in Section 5.2 we present the MM-ALM algorithm for solving NRMR; in Section 5.3 we give a more efficient algorithm which is based on MM-ALM.

5.1 Generalized Singular Value Shrinkage Operator

The Singular Value Threshold (SVT) algorithm is proposed in [Cai *et al.*, 2010] for solving the nuclear norm penalized problems. For $\tau \geq 0$, the singular value shrinkage operator $\mathcal{S}_\tau$ is defined by

$$\begin{aligned} \mathcal{S}_\tau(\mathbf{X}) &= \mathbf{U}_\mathbf{X}\mathcal{D}_\tau(\mathbf{\Sigma}_\mathbf{X})\mathbf{V}_\mathbf{X}^T, \quad \text{where} \\ \big[\mathcal{D}_\tau(\mathbf{A})\big]_{ij} &= \operatorname{sgn}(A_{ij})\big(|A_{ij}| - \tau\big)_+, \end{aligned} \tag{5}$$

where $\mathbf{X} = \mathbf{U}_\mathbf{X}\mathbf{\Sigma}_\mathbf{X}\mathbf{V}_\mathbf{X}^T$ is the singular value decomposition (SVD) of $\mathbf{X}$. Similarly, we define the generalized singular value shrinkage operator $\mathcal{S}_{\tau,\mathbf{\Lambda}}$ and generalized shrinkage operator $\mathcal{D}_{\tau,\mathbf{W}}$ by

$$\begin{aligned} \mathcal{S}_{\tau,\mathbf{\Lambda}}(\mathbf{X}) &= \mathbf{U}_\mathbf{X}\mathcal{D}_{\tau,\mathbf{\Lambda}}(\mathbf{\Sigma}_\mathbf{X})\mathbf{V}_\mathbf{X}^T, \quad \text{where} \\ \big[\mathcal{D}_{\tau,\mathbf{W}}(\mathbf{A})\big]_{ij} &= \operatorname{sgn}(A_{ij})\big(|A_{ij}| - \tau W_{ij}\big)_+. \end{aligned} \tag{6}$$

Here $\mathbf{\Lambda}$ and $\mathbf{W}$ are arbitrary elementwise nonnegative matrices. The optimality of $\mathcal{S}_{\tau,\mathbf{\Lambda}}$ and $\mathcal{D}_{\tau,\mathbf{W}}$ is shown in Theorem 3.

Theorem 3. *For each $\tau \geq 0$, $\gamma > 1$, $\mathbf{X}, \mathbf{Y}, \mathbf{X}^{old} \in \mathbb{R}^{m\times n}$, let $\mathbf{\Lambda} = \big(\mathbf{I} - \mathbf{\Sigma}_{\mathbf{X}^{old}}/\gamma\big)_+$, $\mathbf{W} = \big(\mathbf{1}_m\mathbf{1}_n^T - |\mathbf{X}^{old}|/\gamma\big)_+$, then $\mathcal{S}_{\tau,\mathbf{\Lambda}}$ and $\mathcal{D}_{\tau,\mathbf{W}}$ defined in (6) obey*

$$\mathcal{S}_{\tau,\mathbf{\Lambda}}(\mathbf{Y}) = \operatorname*{argmin}_\mathbf{X} \frac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \tau Q_\gamma\Big(\boldsymbol{\sigma}(\mathbf{X})\Big|\boldsymbol{\sigma}(\mathbf{X}^{old})\Big),$$

$$\mathcal{D}_{\tau,\mathbf{W}}(\mathbf{Y}) = \operatorname*{argmin}_\mathbf{X} \frac{1}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \tau Q_\gamma\Big(\mathbf{X}\Big|\mathbf{X}^{old}\Big),$$

where

$$Q_\gamma(\mathbf{A}|\mathbf{A}^{old}) = M_\gamma(\mathbf{A}^{old}) + \sum_{i,j}(1 - |A_{ij}^{old}|/\gamma)_+(|A_{ij}| - |A_{ij}^{old}|).$$

is the locally linear approximation (LLA) of $M_\gamma(\mathbf{A})$ given $\mathbf{A}^{old}$.

5.2 The MM-ALM Algorithm

The majorization-minimization (MM) method in [Hunter and Li, 2005] is a family of algorithms such as the locally quadratic approximation algorithm (LQA) in [Fan and Li, 2001], the locally linear approximation algorithm (LLA) in [Zou and Li, 2008; Zhang, 2010b], etc. Many optimization problems with nonconvex penalties can be solved by MM, e.g. those in [Fan and Li, 2001; Zou and Li, 2008; Zhang, 2010a; 2010b]. MM proceeds by iteratively solving a convex problem which locally approximates the original nonconvex problem in each step.

Algorithm 1 NRMR via MM-ALM Algorithm

Input: Observation $\mathbf{D} \in \mathbb{R}^{m\times n}$, parameters λ, γ_1, γ_2.
Initialize $\mathbf{Y}^{(0)}$, $\mathbf{L}_0^*$, $\mathbf{S}_0^*$; $\mu > 0$; $k = 0$;
repeat
  $\mathbf{L}_{k+1}^{(0)} \leftarrow \mathbf{L}_k^*$; $\mathbf{S}_{k+1}^{(0)} \leftarrow \mathbf{S}_k^*$;
  $\mathbf{\Lambda} \leftarrow \mathsf{Diag}\big(\mathbf{1}_n - \boldsymbol{\sigma}(\mathbf{L}_k^*)/\gamma_1\big)_+$;
  $\mathbf{W} \leftarrow \big(\mathbf{1}_m\mathbf{1}_n^T - |\mathbf{S}_k^*|/\gamma_2\big)_+$;
  repeat
    $\mathbf{L}_{k+1}^{(j+1)} \leftarrow \mathcal{S}_{1/\mu,\mathbf{\Lambda}}\big(\mathbf{D} - \mathbf{S}_{k+1}^{(j)} + \mathbf{Y}^{(j)}/\mu\big)$;
    $\mathbf{S}_{k+1}^{(j+1)} \leftarrow \mathcal{D}_{\lambda/\mu,\mathbf{W}}\big(\mathbf{D} - \mathbf{L}_{k+1}^{(j+1)} + \mathbf{Y}^{(j)}/\mu\big)$;
    $\mathbf{Y}^{(j+1)} \leftarrow \mathbf{Y}^{(j)} + \mu\big(\mathbf{D} - \mathbf{L}_{k+1}^{(j+1)} - \mathbf{S}_{k+1}^{(j+1)}\big)$;
    $j \leftarrow j + 1$;
  until converged
  $\mathbf{L}_{k+1}^* \leftarrow \mathbf{L}_{k+1}^{(j+1)}$; $\mathbf{S}_{k+1}^* \leftarrow \mathbf{S}_{k+1}^{(j+1)}$; $k \leftarrow k + 1$;
until converged
return $\mathbf{L}_{k+1}^*$, $\mathbf{S}_{k+1}^*$.

Our MM-ALM algorithm is a kind of the MM method. MM-ALM consists of an outer loop and an inner loop. In each iteration, the outer loop replaces the nonconvex problem by its LLA to form a weighted RPCA, while the inner loop is an ALM algorithm solving the resulting weighted RPCA problem. MM-ALM repeats these two loops until converged. MM-ALM is described in Algorithm 1.

LLA based outer loop

The outer loop is based on LLA. The basic idea is shown in the following proposition which indicates that the LLA majorizes the original problem.

Proposition 4. *Let $p(x)$ be a concave function on $(0, \infty)$. Then*

$$p(x) \leq p(x_0) + p'(x_0)(x - x_0), \tag{7}$$

with equality if $x = x_0$.

Since the objective function $f(\mathbf{L}, \mathbf{S})$ in Problem 4 is concave w.r.t. $(\boldsymbol{\sigma}(\mathbf{L}), |\mathbf{S}|)$, in each step we approximate $\|\mathbf{L}\|_{\gamma_1} + \lambda M_{\gamma_2}(\mathbf{S})$ by its LLA at $(\boldsymbol{\sigma}(\mathbf{L}^{\text{old}}), |\mathbf{S}^{\text{old}}|)$ and obtain the following problem:

$$\begin{aligned} \min_{\mathbf{L},\mathbf{S}} \quad & Q_{\gamma_1}\Big(\boldsymbol{\sigma}(\mathbf{L})\Big|\boldsymbol{\sigma}(\mathbf{L}^{\text{old}})\Big) + \lambda Q_{\gamma_2}\Big(\mathbf{S}\Big|\mathbf{S}^{\text{old}}\Big); \\ \text{s.t.} \quad & \mathbf{L} + \mathbf{S} = \mathbf{D}, \end{aligned} \tag{8}$$

which majorizes the NRMR problem in (4). The problem above is indeed a weighted RPCA. Thus we can resort to the ALM algorithm in [Lin *et al.*, 2009], which solves RPCA, to solve Problem 8.

The ALM algorithm based inner loop

We employ the ALM algorithm to solve Problem 8. The augmented Lagrangian function is

$$\begin{aligned} L_\mu\big(\mathbf{L},\mathbf{S},\mathbf{Y}\big|\mathbf{L}^{\text{old}},\mathbf{S}^{\text{old}}\big) &= Q_{\gamma_1}\big(\boldsymbol{\sigma}(\mathbf{L})\big|\boldsymbol{\sigma}(\mathbf{L}^{\text{old}})\big) \\ &\quad + \lambda Q_{\gamma_2}\big(\mathbf{S}\big|\mathbf{S}^{\text{old}}\big) + \langle \mathbf{Y}, \mathbf{D}-\mathbf{L}-\mathbf{S}\rangle + \frac{\mu}{2}\|\mathbf{D}-\mathbf{L}-\mathbf{S}\|_F^2. \end{aligned}$$

The ALM algorithm solves Problem 8 by alternately minimizing L_μ w.r.t. $\mathbf{L}$ and $\mathbf{S}$, and maximizing w.r.t. $\mathbf{Y}$; each step is guaranteed by Theorem 3. This constitutes the inner loop of MM-ALM.

Local Convergence

According to Proposition 4 and the properties of the ALM algorithm, we have that the objective function values of Problem 4 in each iteration of Algorithm 1 obey

$$\begin{aligned} f(\mathbf{L}^*_{k+1}, \mathbf{S}^*_{k+1}) &\leq Q_{\gamma_1}\left(\boldsymbol{\sigma}(\mathbf{L}^*_{k+1})\middle|\boldsymbol{\sigma}(\mathbf{L}^*_k)\right) + \lambda Q_{\gamma_2}\left(\mathbf{S}^*_{k+1}\middle|\mathbf{S}^*_k\right) \\ &\leq Q_{\gamma_1}\left(\boldsymbol{\sigma}(\mathbf{L}^*_k)\middle|\boldsymbol{\sigma}(\mathbf{L}^*_k)\right) + \lambda Q_{\gamma_2}\left(\mathbf{S}^*_k\middle|\mathbf{S}^*_k\right) \\ &= f(\mathbf{L}^*_k, \mathbf{S}^*_k). \end{aligned}$$

Thus the objective function value is monotonically nonincreasing. If we denote the optimal low rank matrix by $\mathbf{L}^*$ and sparse matrix by $\mathbf{S}^*$, then $(\mathbf{L}^*, \mathbf{S}^*)$ is a fixed point of Algorithm 1, that is, if we denote $(\mathbf{L}_{k+1}, \mathbf{S}_{k+1}) = N(\mathbf{L}_k, \mathbf{S}_k)$, then we have $(\mathbf{L}^*, \mathbf{S}^*) = N(\mathbf{L}^*, \mathbf{S}^*)$.

Though the convergence is local, local optimal solutions are still effective. The empirical results in [Breheny and Huang, 2011; Fan and Li, 2001; Shi *et al.*, 2011; Zhang, 2010a; Zhang *et al.*, 2012; Gong *et al.*, 2012; Xiang *et al.*, 2012] all show that the local optimal solutions to the nonconvex problems usually outperform global optimal solution to the LASSO [Tibshirani, 1996] problem. In Section 6 we will empirically demonstrate that the local optimal solutions to NRMR are superior over the global optimal solutions to RPCA.

5.3 Speedup Strategy for MM-ALM

As we see, each step of the MM procure results in a weighted RPCA; the MM-ALM algorithm repeatedly calls the ALM algorithm to solve the weighted RPCA problem in the inner loop. Thus the computation costs can be many times more than the ALM algorithm solving RPCA. To alleviate the computations, we propose an efficient strategy.

A feasible strategy is the *one-step LLA* studied in [Zou and Li, 2008]. The one-step LLA runs the outer loop only once instead of waiting to converge. In comparison, Algorithm 1 is actually the multi-stage LLA in [Zhang, 2010b]. Here we use the solutions to RPCA for initialization. Our off-line experiments reveal that the full MM-ALM leads to only marginal improvement over one-step LLA by sacrificing much more time. Experiments show that one-step LLA at most doubles the time of the ALM algorithm for solving RPCA.

6 Experiments

In this section, we conduct experiments mainly in comparison with RPCA, which is the state-of-the-art low-rank matrix recovery method. Though there are several nonconvex approaches such as log-det heuristic [Fazel *et al.*, 2003], Matrix ALPS [Kyrillidis and Cevher, 2012], and SpaRCS [Waters *et al.*, 2011], none of them were claimed to outperform RPCA in terms of accuracy, so we do not compare with these nonconvex methods.

In our experiments, RPCA is solved by the ALM algorithm in [Lin *et al.*, 2009], and our NRMR is solved by the one-step LLA algorithm; we do not use the full MM-ALM algorithm because it is time consuming. We fix $\gamma_1 = \gamma_2 = 4$ for NRMR in all the following experiments. We use the relative square error (RSE) to evaluate matrix recovery accuracy. The RSE

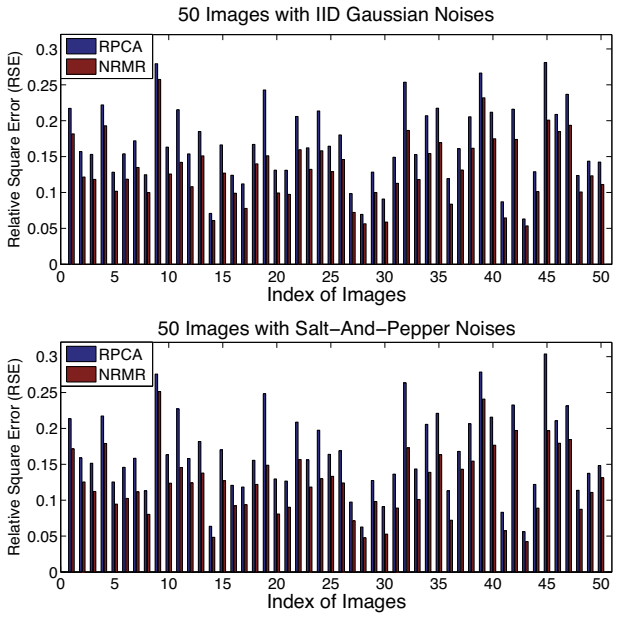


Figure 2: Results on the 50 test images, each contaminated with i.i.d. Gaussian noises or salt-pepper noises. On average, RCPA conducts 39.0 times SVD, while NRMR conducts 72.5 times SVD.

is defined as follows:

$$\text{RSE} = \|\mathbf{L}^* - \mathbf{L}\|_F / \|\mathbf{L}\|_F, \tag{9}$$

where $\mathbf{L}^*$ is the solution to RPCA or NRMR, and $\mathbf{L}$ is the ground truth. We use the number of singular value decompositions (SVD) to evaluate time efficiency, because the running time of all the aforementioned algorithms are dominated by the SVD in each iteration.

In Section 6.1 we conduct experiments on a set of synthetic data. In Section 6.2 we conduct experiments on natural image for the sake of presenting and comparing the results intuitively.

Table 1: Comparisons between RPCA and NRMR on the synthetic data

$\text{rank}(\mathbf{L})$ (r)	RSE		# SVD		$\text{rank}(\mathbf{L}^*)$	
	RPCA	NRMR	RPCA	NRMR	RPCA	NRMR
	$m = 500$, $\|\mathbf{S}\|_0 = 0.05m^2$					
50	8.58×10^{-7}	8.29×10^{-7}	21	40	50	50
100	7.94×10^{-7}	9.97×10^{-7}	29	68	100	100
150	3.30×10^{-2}	7.39×10^{-5}	36	72	419	153
200	0.165	0.119	36	73	471	475
	$m = 500$, $\|\mathbf{S}\|_0 = 0.2m^2$					
50	1.68×10^{-3}	2.91×10^{-6}	32	60	61	50
100	0.278	0.167	34	70	282	195
150	0.460	0.394	34	73	328	259
200	0.550	0.536	34	73	325	266

6.1 Synthetic Data

We first compare between RPCA and NRMR on a set of synthetic data. The data are generated in the following way. We

(a) original (b) noisy (c) RPCA (d) NRMR (e) RPCA (f) NRMR

Figure 3: Denoising results on images with Gaussian noises. 3(a) original images; 3(b) noisy images; 3(c) and 3(d) are obtained by tuning λ such that the RSEs are minimized; 3(e) and 3(f) are of the same rank ($\mathrm{rank}(\mathbf{L}^*) = 100$).

generate $\mathbf{L}$ as a product of two $m \times r$ matrices whose entries are sampled i.i.d. from Gaussian distribution $\mathcal{N}(0, 1/m)$. The sparse component $\mathbf{S}$ is constructed by setting a proportion of entries to be ± 1 and the rest to be zeros. Then we use RPCA and NRMR to recover the low-rank and sparse components from the data matrix $\mathbf{D} = \mathbf{L} + \mathbf{S}$. We tune the parameters λ of both RPCA and NRMR and report the results where RSE is minimized. We report the RSE, number of SVD, and $\mathrm{rank}(\mathbf{L}^*)$ in Table 1.

6.2 Natural Images

We use 50 images from the Berkeley Segmentation Dataset [Martin *et al.*, 2001]. Each image consists of three components: $\mathbf{L}_{red}, \mathbf{L}_{green}, \mathbf{L}_{blue} \in \mathbb{R}^{m \times n}$, each entry of which takes value in $[0, 1)$. Our off-line experiments show that direct concatenation of the three components to a matrix leads to better results than segmenting the image into patches followed by concatenating the patches. So we directly stack the three components into an $m \times 3n$ matrix $\mathbf{L}$:

$$\mathbf{L} = [\mathbf{L}_{red}, \mathbf{L}_{green}, \mathbf{L}_{blue}] \in \mathbb{R}^{m \times 3n}. \tag{10}$$

Then we add noises to $\mathbf{L}$ to generate contaminated data matrix $\mathbf{D}$. We add two kinds of noises: i.i.d. Gaussian noises $\mathcal{N}(0, 0.2^2)$ to 50% pixels of the images, or salt and pepper noises to 20% pixels.

For both RPCA and NRMR, we tune the parameter λ that the recovered low-rank matrix has a rank of $\mathrm{rank}(\mathbf{L}^*) = 100$. We report the resulting RSE and average number of SVD in Figure 2. We also give in Figure 3 some visual comparisons among the recovered low-rank matrices, where λ is tuned such that RSE is minimized or that $\mathrm{rank}(\mathbf{L}^*) = 100$.

We can see from Table 1 and Figure 2 that the local optimal of NRMR always outperforms the global optimal of RPCA (in terms of RSE), without adding much computation costs. The recovered images in Figure 3 all demonstrate that the results of our NRMR better approximate the original images.

7 Concluding Remarks

In this paper we have proposed a novel approach to the matrix recovery problem. In particular, we have defined robust matrix recovery as an optimization problem with a nonconvex loss function and a nonconvex penalty function. We have also devised a majorization-minimization augmented Lagrange multiplier algorithm for solving this nonconvex problem. Experiment results demonstrate that our nonconvex approach significantly outperforms RPCA—a classical convex approach—in terms of accuracy without much extra computation cost.

Our work has shed light on some important properties of the nonconvex low-rank-inducing penalty. The matrix γ-norm studied in this paper is a tighter approximation to the matrix rank than the nuclear norm is, and it broadens our vision on matrix low-rank learning — from a convex relaxation to a more general nonconvex relaxation. Furthermore, it might be interesting to further explore the γ-norm as well as other potential nonconvex low-rank-inducing penalties to better solve the low-rank matrix learning problem.

Acknowledgments

This work has been supported in part by the Natural Science Foundations of China (No. 61070239) and the Scholarship Award for Excellent Doctoral Student granted by Ministry of Education.

References

[Breheny and Huang, 2011] P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics*, 5(1):232–253, 2011.

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

[Candès *et al.*, 2011] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), May 2011.

[Cheng *et al.*, 2011] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[Fan and Li, 2001] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its Oracle properties. *Journal of the American Statistical Association*, 96:1348–1361, 2001.

[Fazel *et al.*, 2003] M. Fazel, H. Hindi, and S.P. Boyd. Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices. In *Proceedings of the American Control Conference*. IEEE, 2003.

[Gong *et al.*, 2012] P. Gong, J. Ye, and C. Zhang. Multi-stage multi-task feature learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[Hunter and Li, 2005] R. Hunter and R. Li. Variable selection using MM algorithm. *Annals of Statistics*, pages 1617–1642, 2005.

[Kyrillidis and Cevher, 2012] A. Kyrillidis and V. Cevher. Matrix ALPS: Accelerated low rank and sparse matrix reconstruction. *arXiv preprint arXiv:1203.3864*, 2012.

[Lin *et al.*, 2009] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report, UILU-ENG-09-2215*, 2009.

[Liu *et al.*, 2010] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning (ICML)*, 2010.

[Martin *et al.*, 2001] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423, July 2001.

[Mazumder *et al.*, 2010] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(2):2287–2322, 2010.

[Peng *et al.*, 2010] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[Shi *et al.*, 2011] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang. A non-convex relaxation approach to sparse dictionary learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[Tibshirani, 1996] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society B*, pages 267–288, 1996.

[Wang and Zhang, 2012] S. Wang and Z. Zhang. Colorization by matrix completion. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2012.

[Waters *et al.*, 2011] A.E. Waters, A.C. Sankaranarayanan, and R.G. Baraniuk. SpaRCS: Recovering low-rank and sparse matrices from compressive measurements. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[Wright *et al.*, 2009] J. Wright, A. Ganesh, S. Rao, 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 (NIPS)*, 2009.

[Xiang *et al.*, 2012] S. Xiang, X. Shen, and J. Ye. Efficient sparse group feature selection via nonconvex optimization. *arXiv preprint arXiv:1205.5075*, 2012.

[Zhang and Tu, 2012] Z. Zhang and B. Tu. Nonconvex penalization using laplace exponents and concave conjugates. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[Zhang *et al.*, 2011] Z. Zhang, Y. Matsushita, and Y. Ma. Camera calibration with lens distortion from low-rank textures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[Zhang *et al.*, 2012] Z. Zhang, S. Wang, D. Liu, and M. I. Jordan. EP-GIG priors and applications in Bayesian sparse learning. *Journal of Machine Learning Research*, 13:2031–2061, 2012.

[Zhang, 2010a] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38:894–942, 2010.

[Zhang, 2010b] T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010.

[Zou and Li, 2008] H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics*, 36(4):1509–1533, 2008.