Informed Non-Convex Robust Principal Component Analysis with Features

Niannan Xue,[1] Jiankang Deng,[1] Yannis Panagakis,[12] Stefanos Zafeiriou[13]
[1]Department of Computing, Imperial College London, UK
[2]Department of Computer Science, Middlesex University, UK
[3] Center for Machine Vision and Signal Analysis, University of Oulu, Finland
{n.xue15, j.deng16, i.panagakis, s.zafeiriou}@imperial.ac.uk

Abstract

We revisit the problem of robust principal component analysis with features acting as prior side information. To this aim, a novel, elegant, non-convex optimization approach is proposed to decompose a given observation matrix into a low-rank core and the corresponding sparse residual. Rigorous theoretical analysis of the proposed algorithm results in exact recovery guarantees with low computational complexity. Aptly designed synthetic experiments demonstrate that our method is the first to wholly harness the power of non-convexity over convexity in terms of both recoverability and speed. That is, the proposed non-convex approach is more accurate and faster compared to the best available algorithms for the problem under study. Two real-world applications, namely image classification and face denoising further exemplify the practical superiority of the proposed method.

Introduction

Several machine learning and signal processing tasks involve the separation of a data matrix into a low-rank matrix and a matrix with sparse support (i.e., a sparse matrix) containing entries of arbitrary magnitude. Robust principal component analysis (RPCA) (Candes et al. 2011; Chandrasekaran et al. 2011) offers a provably correct and convenient way to solve this matrix separation problem, when certain incoherence conditions on the data hold. In fact, RPCA solves a convex relaxation of the natural rank minimization problem which is regularized by the sparsity promoting ℓ_0-(quasi) norm.

Nevertheless, prior side information, oftentimes in the form of features, may also be present in practice. For instance, features are available for the following tasks:

– Collaborative filtering: apart from ratings of an item by other users, the profile of the user and the description of the item can also be exploited in making recommendations (Chiang, Hseih, and Dhillon 2015);

– Relationship prediction: user behaviours and message exchanges can assist in finding missing links on social media networks (Xu, Jin, and Zhou 2013);

– Person-specific facial deformable models: an orthonormal subspace learnt from manually annotated data captured in-the-wild, when fed into an image congealing procedure, can help produce more correct fittings (Sagonas et al. 2014).

It is thus reasonable to investigate whether it is propitious for RPCA to exploit the available side information by incorporating features. Indeed, recent results (Liu, Liu, and Li 2017), indicate that in case of union of multiple subspaces where RPCA degrades due to the increasing row-coherence (when the number of subspaces grows), the use of features as side information allow accurate low-rank recovery by removing its dependency on the row-coherence. Despite the theoretical and practical merits of convex variants of RPCA with features, such as LRR (Liu, Lin, and Yu 2010) and PCPF (Chiang, Hsieh, and Dhillon 2016), convex relaxations of the rank function and l_0-norm result into *algorithm weakening* (Chandrasekarana and Jordan 2013).

On a separate note, recent advances in non-convex optimization algorithms continue to undermine their convex counterparts (Gong et al. 2013; Ge, Lee, and Ma 2016; Kohler and Lucchi 2017). In particular, non-convex RPCA algorithms such as fast RPCA (Yi et al. 2016) and AltProj (Netrapalli et al. 2014) exhibit better properties than the convex formulation. Most recently, (Niranjan, Rajkumar, and Tulabandhula 2017) embedded features into a non-convex RPCA framework known as IRPCA-IHT with faster speed. However, it remains unclear, how to exploit side information in non-convex RPCA and whether it facilitates provably correct, fast, and more accurate algorithms.

In this work, we give positive answers to the above questions by proposing a novel, non-convex scheme that fully leverages side information (features) regarding row and column subspaces of the low-rank matrix. Even though the proposed algorithm is inspired by the recently proposed fast RPCA (Yi et al. 2016), our contributions are by no means trivial, especially from a theoretical perspective. First, fast RPCA cannot be easily extended to consistently take account of features. Second, as we show in this paper, incoherence assumptions on the observation matrix and features play a decisive role in determining the corruption bound and the computational complexity of the non-convex algorithm. Third, fast RPCA is limited to a corruption rate of 50% due to their choice of the hard threshold, whereas our algorithm ups this rate to 90%. Fourth, we prove that the costly projection onto factorized spaces is entirely optional when features satisfy certain incoherence conditions. Although our algorithm maintains the same corruption rate of $O(\frac{n}{r^{1.5}})$ and complexity of $O(rn^2\log(\frac{1}{\epsilon}))$ as fast RPCA, we show empirically that massive gains in accuracy and speed can still be obtained. Besides, the transfer of coherence dependency from observation to features means that our algorithm is capable of dealing with highly incoherent data.

Unavoidably, features adversely affect tolerance to corruption in IRPCA-IHT ($O(\frac{n}{d})$) compared to its predecessor AltProj ($O(\frac{n}{r})$). This is not always true with our algorithm in relation to fast RPCA. And when the underlying rank is low but features are only weakly informative, *i.e.* $r \ll d$, which is often the case, our tolerance to corruption is arguably better. IRPCA-IHT also has a higher complexity of $O((dn^2+d^2r)\log(\frac{1}{\epsilon}))$ than that of our algorithm. Although feature-free convex and non-convex algorithms have higher asymptotic error bounds than our algorithm, we show in our experiments that this does not translate as accuracy in reality. Our algorithm still has the best performance in recovering accurately the low-rank part from highly corrupted matrices. This may be attributed to the fact that our bounds are not tight. Besides, PCPF and AltProj have much higher complexity ($O(\frac{n^3}{\sqrt{\epsilon}})$ and $O(r^2n^2\log(\frac{1}{\epsilon}))$) than ours. For PCPF, there does not exist any theoretical analysis under the deterministic sparsity model. Nonetheless, we show in our experiments that our algorithm is superior with regard to both recoverability and running time.

The contributions of this paper are summarised as follows:

- A novel non-convex algorithm integrating features with informed sparsity is proposed in order to solve RPCA problem.
- We establish theoretical guarantees of exact recovery under different assumptions regarding the incoherence of features and observation.
- Extensive experimental results on synthetic data indicate that the proposed algorithm is faster and more accurate in low-rank matrix recovery than the compared state-of-the-art convex and non-convex methods for RPCA (with and without features).
- Experiments on two real-world datasets, namely MNIST and Yale B database demonstrate the practical merits of the proposed algorithm.

Notations

Lowercase letters denote scalars and uppercase letters denote matrices, unless otherwise stated. $\mathbf{A}i\cdot$ and $\mathbf{A}\cdot j$ represent the i^{th} row and the j^{th} column of $\mathbf{A}$. Projection onto support set Ω is given by $\mathbf{\Pi}_\Omega$. $|\mathbf{A}|$ is the element-wise absolute value of matrix $\mathbf{A}$. For norms of matrix $\mathbf{A}$, $\|\mathbf{A}\|_F$ is the Frobenius norm; $\|\mathbf{A}\|_*$ is the nuclear norm; $\|\mathbf{A}\|_2$ is the largest singular value; otherwise, $\|\mathbf{A}\|_p$ is the l_p-norm of vectorized $\mathbf{A}$; and $\|\mathbf{A}\|_{2,\infty}$ is the maximum of matrix row l_2-norms. Moreover, $\langle\mathbf{A},\mathbf{B}\rangle$ represents tr($\mathbf{A}^T\mathbf{B}$) for real matrices $\mathbf{A},\mathbf{B}$. Additionally, σ_i is the i^{th} largest singular value of a matrix.

The Euclidean metric is not applicable here because of the non-uniqueness of the bi-factorisation $\mathbf{L}^*=\mathbf{A}^*\mathbf{B}^{*T}$, which corresponds to a manifold rather than a point. Hence, we define the following distance between $(\mathbf{A},\mathbf{B})$ and any of the optimal pair $(\mathbf{A}^*,\mathbf{B}^*)$ such that $\mathbf{L}^*=\mathbf{A}^*\mathbf{B}^{*T}$:

$$d(\mathbf{A},\mathbf{B},\mathbf{A}^*,\mathbf{B}^*)=\min_{\mathbf{R}}\sqrt{\|\mathbf{A}-\mathbf{A}^*\mathbf{R}\|_F^2+\|\mathbf{B}-\mathbf{B}^*\mathbf{R}\|_F^2}, \tag{1}$$

where $\mathbf{R}$ is an $r\times r$ orthogonal matrix.

Related Work

RPCA concerns a known observation matrix $\mathbf{M}$ which we are seeking to decompose into matrices $\mathbf{L}^*$, $\mathbf{S}^*$ such that $\mathbf{L}^*$ is low-rank and $\mathbf{S}^*$ is sparse and of arbitrary magnitude. Conceptually, it is equivalent to solving the following optimization problem:

$$\min_{\mathbf{L},\mathbf{S}}\ \text{rank}(\mathbf{L})+\gamma\|\mathbf{S}\|_0 \quad \text{subject to} \quad \mathbf{L}+\mathbf{S}=\mathbf{M}, \tag{2}$$

for appropriate γ. This problem, regrettably, is NP-hard.

PCP (Wright et al. 2009) replaces (2) with convex heuristics:

$$\min_{\mathbf{L},\mathbf{S}}\ \|\mathbf{L}\|_*+\gamma\|\mathbf{S}\|_1 \quad \text{subject to} \quad \mathbf{L}+\mathbf{S}=\mathbf{M}, \tag{3}$$

for some γ. In spite of the simplification, PCP can exactly recover the solution of RPCA under the random model (Candes et al. 2011) and the deterministic model (Chandrasekaran et al. 2011; Hsu, Kakade, and Zhang 2011).

If feasible feature dictionaries, $\mathbf{X}$ and $\mathbf{Y}$, regarding row and column spaces are available, PCPF (Chiang, Hsieh, and Dhillon 2016) makes use of these to generalize (3) to the below objective:

$$\min_{\mathbf{H},\mathbf{S}}\ \|\mathbf{H}\|_*+\gamma\|\mathbf{S}\|_1 \quad \text{subject to} \quad \mathbf{X}\mathbf{H}\mathbf{Y}^T+\mathbf{S}=\mathbf{M}, \tag{4}$$

for the same γ as in (3). Convergence to the RPCA solution has only been established for the random sparsity model.

AltProj (Netrapalli et al. 2014) addresses RPCA by minimizing an entirely different objective:

$$\begin{aligned}&\min_{\mathbf{L},\mathbf{S}}\ \|\mathbf{M}-\mathbf{L}-\mathbf{S}\|_F\\ \text{subject to}\quad &\mathbf{L}\in\ \text{set of low-rank matrices}\\ &\mathbf{S}\in\ \text{set of sparse matrices,}\end{aligned} \tag{5}$$

where the search consists of alternating non-convex projections. That is, during each cycle, hard-thresholding takes place first to remove large entries and projection of appropriate residuals onto the set of low-rank matrices with increasing ranks is carried out next. Exact recovery has also been established.

Fast RPCA (Yi et al. 2016) follows yet another non-convex approach to solve RPCA. After an initialization stage, fast RPCA updates bilinear factors $\mathbf{U}$, $\mathbf{V}$ such that $\mathbf{L}=\mathbf{U}\mathbf{V}^T$ through a series of projected gradient descent and sparse estimations, where $\mathbf{U}$, $\mathbf{V}$ minimize the following loss:

$$\min_{\mathbf{U},\mathbf{V}}\frac{1}{2}\|\mathbf{U}\mathbf{V}^T+\mathbf{S}-\mathbf{M}\|_F^2+\frac{1}{8}\|\mathbf{U}^T\mathbf{U}-\mathbf{V}^T\mathbf{V}\|_F^2, \tag{6}$$

for $\mathbf{U}$, $\mathbf{V}$ properly constrained. Recovery guarantee is ensured.

IRPCA-IHT (Niranjan, Rajkumar, and Tulabandhula 2017) includes features $\mathbf{X}, \mathbf{Y}$ in an iterative non-convex projection algorithm. Similar to AltProj, at each step, a new sparse estimate is calculated from hard thresholding via a monotonically decreasing threshold. After that, spectral hard thresholding takes place to attain the low-rank estimate. IRPCA-IHT provably converges to the solution of RPCA.

We also mention here several works of non-convex objectives (Oh et al. 2015; Shang et al. 2017), though exact recovery guarantees are lacking.

Problem Setup

Suppose that there is a known data matrix $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2}$, which can be decomposed into a low-rank component $\mathbf{L}^*$ and a sparse error matrix $\mathbf{S}^*$ of compatible dimensions. Our aim is to identify these underlying matrices and hence robustly recover the low-rank component with the help of available side information in the form of feature matrices $\mathbf{X}$ and $\mathbf{Y}$.

Concretely, let $\mathbf{L}^* = \mathbf{U}^* \mathbf{\Sigma}^* \mathbf{V}^{*T}$be the singular value decomposition and $\mathbf{P}^* = \mathbf{X}^T \mathbf{U}^* \mathbf{\Sigma}^{*\frac{1}{2}}$ and $\mathbf{Q}^* = \mathbf{Y}^T \mathbf{V}^* \mathbf{\Sigma}^{*\frac{1}{2}}$. $\mathbf{S}^*$ follows the random sparsity model. That is, the support of $\mathbf{S}^*$ is chosen uniformly at random from the collection of all support sets of the same size. Furthermore, let us be informed of the proportion of non-zero entries per row and column, denoted by α. Assume that there are also available features $\mathbf{X} \in \mathbb{R}^{n_1 \times d_1}$ and $\mathbf{Y} \in \mathbb{R}^{n_2 \times d_2}$ such that they are feasible, i.e. col($\mathbf{X}$)$\supseteq$col($\mathbf{U}^*$) and col($\mathbf{Y}$)$\supseteq$col($\mathbf{V}^*$) where col($\mathbf{A}$) is the column space of $\mathbf{A}$ and $\mathbf{X}^T\mathbf{X} = \mathbf{Y}^T\mathbf{Y} = \mathbf{I}$[1].

In this paper, we discuss robust low-rank recovery using the above mentioned features and three different incoherence conditions: (i) $\|\mathbf{U}^*\|_{2,\infty} \leq \sqrt{\frac{\mu_1 r}{n_1}}$ and $\|\mathbf{V}^*\|_{2,\infty} \leq \sqrt{\frac{\mu_1 r}{n_2}}$; (ii) $\|\mathbf{X}\|_{2,\infty} \leq \sqrt{\frac{\mu_2 d_1}{n_1}}$ and $\|\mathbf{Y}\|_{2,\infty} \leq \sqrt{\frac{\mu_2 d_2}{n_2}}$; (iii) both (i) and (ii), where r is the given rank of $\mathbf{L}^*$ and μ_1, μ_2 are constants.

Algorithm

We use a non-convex approach to achieve the above objective. The algorithm consists of an initialization phase followed by a gradient descent phase. At each stage, we keep track of the factors $\mathbf{P}$, $\mathbf{Q}$ such that $\mathbf{L} = \mathbf{X}\mathbf{P}\mathbf{Q}^T\mathbf{Y}^T$.

Hard-thresholding

We first introduce the sparse estimator via hard-thresholding which is used in both phases. Given a threshold θ, $\mathcal{T}_\theta(\mathbf{A})$ removes elements of $\mathbf{A}$ that are not among the largest θ-fraction of elements in their respective rows and columns, breaking ties arbitrarily for equal elements:

$$\mathcal{T}_\theta(\mathbf{A})_{ij} = \begin{cases} 0 & \text{if } |\mathbf{A}_{ij}| \leq \mathbf{A}^\theta i\cdot \text{ or } |\mathbf{A}_{ij}| \leq \mathbf{A}^\theta \cdot j, \\ \mathbf{A}_{ij} & \text{otherwise,} \end{cases} \tag{7}$$

where $\mathbf{A}^\theta i\cdot, \mathbf{A}^\theta \cdot j$ are the $(n_2\theta)^{\text{th}}$ and $(n_1\theta)^{\text{th}}$ largest element in absolute value in row i and column j respectively.

Initialization

$\mathbf{S}$ is first initialized as $\mathbf{S}_0 = \mathcal{T}_\alpha(\mathbf{M})$. Next, we obtain $\mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^T$ as the r-truncated SVD of $\mathbf{L}_0$, which is calculated via $\mathbf{L}_0 = \mathbf{M} - \mathbf{S}_0$. We can then construct $\mathbf{P}_0 = \mathbf{X}^T\mathbf{U}_0\mathbf{\Sigma}_0^{\frac{1}{2}}$ and $\mathbf{Q}_0 = \mathbf{Y}^T\mathbf{V}_0\mathbf{\Sigma}_0^{\frac{1}{2}}$. Such an initialization scheme gives $\mathbf{P}$, $\mathbf{Q}$ the desirable properties for use in the second phase.

Gradient Descent

In case (i), we need the following sets:

$$\mathcal{P} = \{\mathbf{A} \in \mathbb{R}^{d_1 \times r} | \|\mathbf{X}\mathbf{A}\|_{2,\infty} \leq \sqrt{\frac{2\mu_1 r}{n_1}} \|\mathbf{P}_0\|_2\}, \tag{8}$$

$$\mathcal{Q} = \{\mathbf{A} \in \mathbb{R}^{d_2 \times r} | \|\mathbf{Y}\mathbf{A}\|_{2,\infty} \leq \sqrt{\frac{2\mu_1 r}{n_2}} \|\mathbf{Q}_0\|_2\}. \tag{9}$$

Otherwise, we can simply take $\mathcal{P}$ as $\mathbb{R}^{d_1 \times r}$ and $\mathcal{Q}$ as $\mathbb{R}^{d_2 \times r}$.

To proceed, we first regularise $\mathbf{P}_0$ and $\mathbf{Q}_0$:

$$\mathbf{P} = \mathbf{\Pi}_\mathcal{P}(\mathbf{P}_0), \ \mathbf{Q} = \mathbf{\Pi}_\mathcal{Q}(\mathbf{Q}_0). \tag{10}$$

At each iteratiion, we first update $\mathbf{S}$ with the sparse estimator using a threshold of $\alpha + \min(10\alpha + 0.1)$:

$$\mathbf{S} = \mathcal{T}_{\alpha + \min(10\alpha, 0.1)}(\mathbf{M} - \mathbf{X}\mathbf{P}\mathbf{Q}^T\mathbf{Y}^T). \tag{11}$$

For $\mathbf{P}$, $\mathbf{Q}$, we define the following objective function

$$\mathcal{L}(\mathbf{P}, \mathbf{Q}) = \frac{1}{2}\|\mathbf{X}\mathbf{P}\mathbf{Q}^T\mathbf{Y}^T + \mathbf{S} - \mathbf{M}\|_F^2 + \frac{1}{64}\|\mathbf{P}^T\mathbf{P} - \mathbf{Q}^T\mathbf{Q}\|_F^2. \tag{12}$$

$\mathbf{P}$ and $\mathbf{Q}$ are updated by minimizing the above function subject to the constraints imposed by the sets $\mathcal{P}$ and $\mathcal{Q}$. That is,

$$\mathbf{P} = \mathbf{\Pi}_\mathcal{P}(\mathbf{P} - \eta\nabla_\mathbf{P}\mathcal{L}), \tag{13}$$

$$\mathbf{Q} = \mathbf{\Pi}_\mathcal{Q}(\mathbf{Q} - \eta\nabla_\mathbf{Q}\mathcal{L}), \tag{14}$$

where the step size η is determined analytically below. With properly initialized $\mathbf{P}$ and $\mathbf{Q}$, such an optimization design converges to $\mathbf{P}^*$ and $\mathbf{Q}^*$. The procedure is summarized in Algorithm 1.

Analysis

We first provide theoretical justification of our proposed approach. Then we evaluate its computational complexity. The proofs can be found in the supplementary material.

Convergence

The initialization phase provides us with the following guarantees on $\mathbf{P}$ and $\mathbf{Q}$.

Theorem 1 *In cases (i) and (iii), if $\alpha \leq \frac{1}{16\kappa r\mu_1}$, we have*

$$d(\mathbf{P}_0, \mathbf{Q}_0, \mathbf{P}^*, \mathbf{Q}^*) \leq 18\alpha r\mu_1\sqrt{r\kappa\sigma_1^*}. \tag{15}$$

In case (ii), if $\alpha \leq \frac{1}{16\kappa\mu_2\sqrt{d_1 d_2}}$, we have

$$d(\mathbf{P}_0, \mathbf{Q}_0, \mathbf{P}^*, \mathbf{Q}^*) \leq 18\alpha\mu_2\sqrt{r d_1 d_2 \kappa\sigma_1^*}, \tag{16}$$

where κ is the condition number of L^* and d is a distance metric defined in the appendix.

[1] This can always achieved via orthogonalisation.

Algorithm 1 Non-convex solver for robust principal component analysis with features

```
Input: Observation $\mathbf{M}$, features $\mathbf{X}, \mathbf{Y}$, rank $r$, corruption
    approximation $\alpha$ and step size $\eta$.
    Initialization:
    $\mathbf{S} = \mathcal{T}_\alpha(\mathbf{M})$
    $\mathbf{U\Sigma V}^T = r\text{-SVD}(\mathbf{M} - \mathbf{S})$
    $\mathbf{P} = \mathbf{X}^T\mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}$
    $\mathbf{Q} = \mathbf{Y}^T\mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}$
    Gradient descent:
    $\mathbf{P} = \mathbf{\Pi}_\mathcal{P}(\mathbf{P})$
    $\mathbf{Q} = \mathbf{\Pi}_\mathcal{Q}(\mathbf{Q})$
    while not converged do
        $\mathbf{S} = \mathcal{T}_{\alpha+\min(10\alpha,0.1)}(\mathbf{M} - \mathbf{XPQ}^T\mathbf{Y}^T)$
        $\mathbf{P} = \mathbf{\Pi}_\mathcal{P}(\mathbf{P} - \eta\nabla_\mathbf{P}\mathcal{L})$
        $\mathbf{Q} = \mathbf{\Pi}_\mathcal{Q}(\mathbf{Q} - \eta\nabla_\mathbf{Q}\mathcal{L})$
    end while
Return: $\mathbf{L} = \mathbf{XPQ}^T\mathbf{Y}^T, \mathbf{S}$
```

Theorem 2 *For $\eta \leq \frac{1}{192\|\mathbf{L}_0\|_2}$, there exist constants $c_1 > 0$, $c_2 > 0$, $c_3 > 0$, $c_4 > 0$, $c_5 > 0$ and $c_6 > 0$ such that, in case (i), when $\alpha \leq \frac{c_1}{\mu_1(\kappa r)^{\frac{3}{2}}}$, we have the following relationship*

$$d(\mathbf{P}_t, \mathbf{Q}_t, \mathbf{P}^*, \mathbf{Q}^*)^2 \leq (1 - c_2\eta\sigma_r^*)^t d(\mathbf{P}_0, \mathbf{Q}_0, \mathbf{P}^*, \mathbf{Q}^*)^2, \tag{17}$$

in case (ii), when $\alpha \leq \frac{c_3}{\mu_2 d r^{\frac{1}{2}} \kappa^{\frac{3}{2}}}$, we have

$$d(\mathbf{P}_t, \mathbf{Q}_t, \mathbf{P}^*, \mathbf{Q}^*)^2 \leq (1 - c_4\eta\sigma_r^*)^t d(\mathbf{P}_0, \mathbf{Q}_0, \mathbf{P}^*, \mathbf{Q}^*)^2. \tag{18}$$

and in case (iii), when $\alpha \leq c_5 \min(\frac{1}{\mu_2 d\kappa}, \frac{1}{\mu_1(\kappa r)^{\frac{3}{2}}})$, we have

$$d(\mathbf{P}_t, \mathbf{Q}_t, \mathbf{P}^*, \mathbf{Q}^*)^2 \leq (1 - c_6\eta\sigma_r^*)^t d(\mathbf{P}_0, \mathbf{Q}_0, \mathbf{P}^*, \mathbf{Q}^*)^2. \tag{19}$$

Complexity

From **Theorem 2**, it follows that our algorithm converges at a linear rate under assumptions (ii) and (iii). To converge below ϵ of the initial error, $O(\log(\frac{1}{\epsilon}))$ iterations are needed. At each iteration, the most costly step is matrix multiplication which takes $O(rn^2)$ time. Overall, our algorithm has total running time of $O(rn^2\log(\frac{1}{\epsilon}))$.

Experimental results

We have found that when the step size is set to 0.5, reasonable results can be obtained. For all algorithms in comparison, we run a total of 3000 iterations or until $\|\mathbf{M} - \mathbf{L} - \mathbf{S}\|_F / \|\mathbf{M}\|_F < 10^{-7}$ is met.

Phase transition

Here, we vary the rank and the error sparsity to investigate the behavior of both our algorithm and existing state-of-art algorithms in terms of recoverability. True low-rank matrices are created via $\mathbf{L}^* = \mathbf{JK}^T$, where $200 \times r$ matrices $\mathbf{J}, \mathbf{K}$ have independent elements drawn randomly from a Gaussian distribution of mean 0 and variance $5 \cdot 10^{-3}$ so r becomes the rank of $\mathbf{L}^*$. Next, we corrupt each column of $\mathbf{L}^*$ such that α of the elements are set independently with magnitude $\mathcal{U}(0, \frac{r}{40})$. However, this does not guarantee α row corruption. We thus select only matrices whose maximum row corruption does not exceed $\alpha + 6.5\%$ but we still feed α to the algorithms in order to demonstrate that our algorithm does not need the exact value of corruption ratio. We consider two types of signs for error: Bernoulli ± 1 and $\text{sgn}(\mathbf{L}^*)$. The resulting $\mathbf{M}$ thus becomes the simulated observation. In addition, let $\mathbf{L}^* = \mathbf{U\Sigma V}^T$ be the SVD of $\mathbf{L}^*$. Feature $\mathbf{X}$ is formed by randomly interweaving column vectors of $\mathbf{U}$ with 5 arbitrary orthonormal bases for the null space of $\mathbf{U}^T$, while permuting the expanded columns of $\mathbf{V}$ with 5 random orthonormal bases for the kernel of $\mathbf{V}^T$ forms feature $\mathbf{Y}$. Hence, the feasibility conditions are fulfilled: $\text{col}(\mathbf{X}) \supseteq \text{col}(\mathbf{L}_0)$, $\text{col}(\mathbf{Y}) \supseteq \text{col}(\mathbf{L}_0^T)$. For each (r, α) pair, three observations are constructed. The recovery is successful if for all these three problems,

$$\frac{\|\mathbf{L} - \mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F} < 10^{-3} \tag{20}$$

from the recovered $\mathbf{L}$.

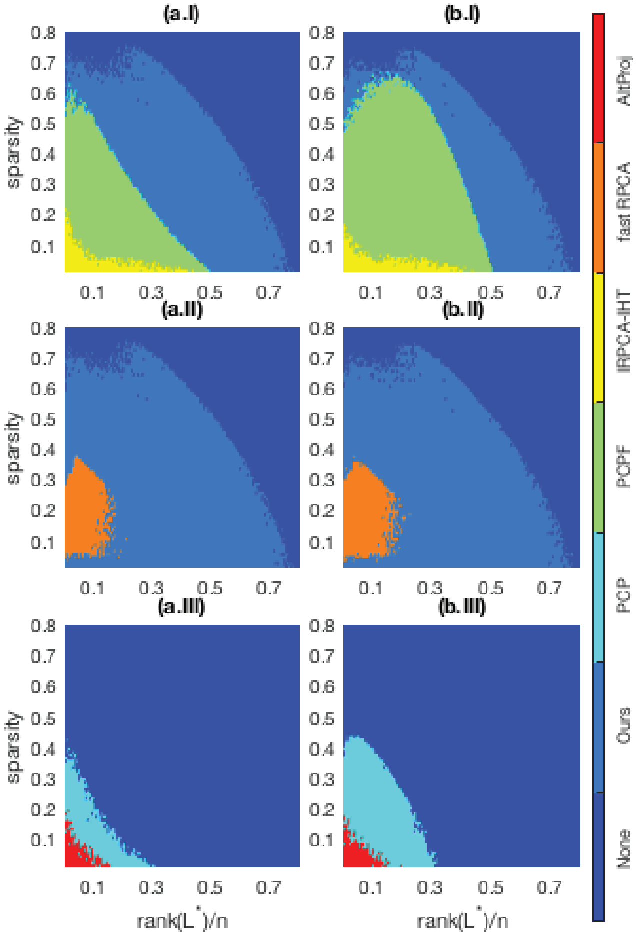

Figure 1: Domains of recovery by various algorithms: (a) for random signs and (b) for coherent signs.

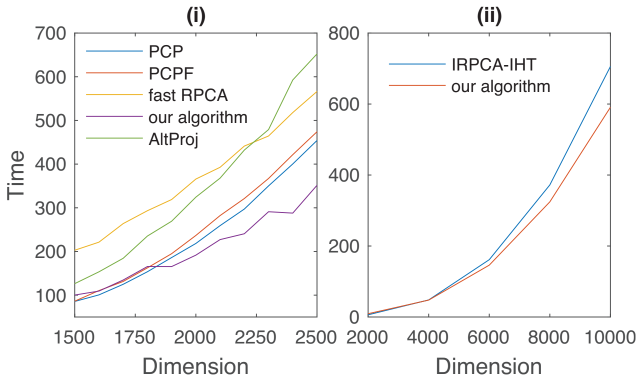

Figure 2: (i) Running times for observation matrices of increasing dimensions for (i) PCP, PCPF, fast RPCA, AltProj, our algorithm and (ii) IRPCA-IHT and our algorithm when $\frac{\|\mathbf{L}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F} \leq 1\%$.

Figures 1(I) plot results from algorithms incorporating features. Besides, our algorithm contrasts with fast RPCA in Figure 1(II). Other feature-free algorithms are investigated in Figure 1(III). Figures 1(a) illustrate the random sign model and Figures 1(b) for the coherent sign model. All previous non-convex attempts fail to outperform their convex equivalents. IRPCA-IHT is unable to deal with even moderate levels of corruption. The frontier of recoverability that has been advanced by our algorithm over PCPF is phenomenal, massively ameliorating fast RPCA. The anomalous asymmetry in the two sign models is no longer observed in non-convex algorithms.

Running Time

Next, we highlight the speed of our algorithm for large-scale matrices, typical of video sequences (Xiong, Liu, and Tao 2016). 1500×1500 to 2500×2500 random observation matrices are generated, where the rank is chosen to be 20% of the column number and random sign error corrupts 11% of the entries, with features $\mathbf{X}, \mathbf{Y}$ having a dimension of 50% of the column number. The running times of all algorithms except IRPCA-IHT are plotted in 2 (i) because IRPCA-IHT is not able to achieve a relative error ($\frac{\|\mathbf{L}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F}$) less than 1% for larger matrices. For fair comparison, we have relaxed the rank to 0.3% of the column number and error rate to 0.1% to compare our algorithm with IRPCA-IHT for matrices ranging from 2000×2000 to 10000×10000. We have used features $\mathbf{X}, \mathbf{Y}$ having a dimension of 80% of the column number to speed up the process. The result is shown in Figure 2 (ii). All times are averaged over three trials. It is evident that, for large matrices, our algorithm overtakes all existing algorithms in terms of speed. Note that features in PCPF even slow down the recovery process.

Image Classification

Once images are denoised, classification can be performed on them. The classification results directly reflect the image denoising ability. For a set of correlated images, low-rank

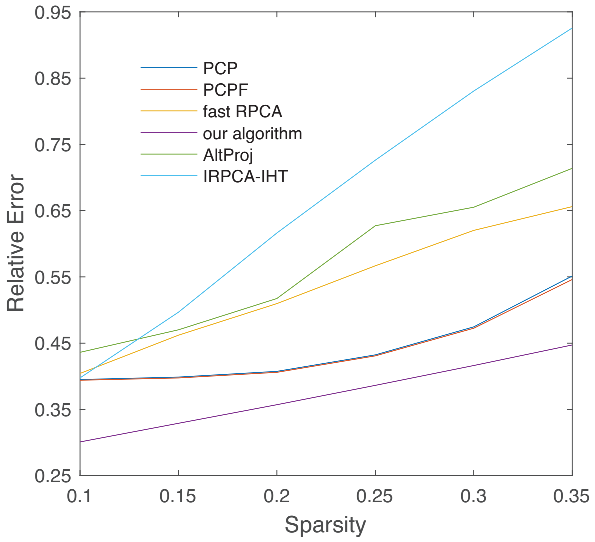

Figure 3: Relative error ($\frac{\|\mathbf{L}-\mathbf{L}^*\|_F}{\|\mathbf{L}^*\|_F}$) for sparsity values: 10%, 15%, 20%, 25%, 30%, 35%.

algorithms are normally used to remove noise that is sparse. The same classifier is thus able to compare the different low-rank models.

The MNIST dataset is such an example which contains hand-written digits divided into training and testing sets. Let the observation matrix be composed of 2000 vectorized random images from the test set stacked column-wise. In this case, the left feature obtained from the training set is also applicable to the test set because of the Eigendigit nature. This imparts our algorithm to supervised learning where there are clean related training samples available. The right feature does not posses such property and is set to the identity matrix. We add a range of sparse noise to the test set separately where the noise sets the pixel to 255. For PCPF, we take $d = 300$ as in (Chiang, Hsieh, and Dhillon 2016) and for IRPCA-IHT and our algorithm we use $d = 150$ instead.

The relative error between the recovered matrix by the competing algorithms and the clean test matrix is plotted in Figure 3. Our algorithm is most accurate in removing the added artificial noise. To evaluate how classifiers perform on the recovered matrices, we train the linear and kernel SVM using the training set and test the corresponding models on the recovered images. Table 1 tabulates the linear SVM. Table 2 tabulates the kernel SVM. Both classifiers confirm the recovery result obtained by various models corroborating our algorithm's pre-eminent accuracy.

Face denoising

It is common practice to decompose raw facial images as a low-rank component for faithful face representation and a sparse component for defects. This is because the face is a convex Lambertian surface which under distant and isotropic lighting has an underlying model that spans a 9-D linear subspace (Basri and Jacobs 2003), but theoreti-

α	clean	noisy	PCP	PCPF	AltProj	IRPCA-IHT	fast RPCA	our algorithm
10	89.65	30.45	82.75	83.35	81.4	65.2	81.1	**86.9**
15		25.1	82.95	83.4	81.15	49.65	79.65	**84.8**
20		23.15	83.5	84	79.3	37.8	78.65	**83.8**
25		18.65	81.35	82.65	74.05	30.35	75.3	**83.15**
30		18.6	77.95	79	71.5	24.1	72.9	**82.05**
35		16.95	71.2	73.4	67.75	21.05	71.45	**79.05**

Table 1: Classification results obtained by a linear SVM.

α	clean	noisy	PCP	PCPF	AltProj	IRPCA-IHT	fast RPCA	our algorithm
10	92.25	87	87.25	87.3	86.45	89.3	89.25	**90.3**
15		75.85	87.15	87.4	86.75	82.85	87.2	**89.8**
20		64.35	87.6	87.55	84.65	71.2	85.55	**88.55**
25		55.85	87	86.95	79.4	62.35	82.65	**87.8**
30		47.15	81.15	81.55	76.75	53.5	78.3	**85.65**
35		40.55	74.8	75.7	71	47.4	76.75	**85.15**

Table 2: Classification results obtained by an SVM with RBF kernel.

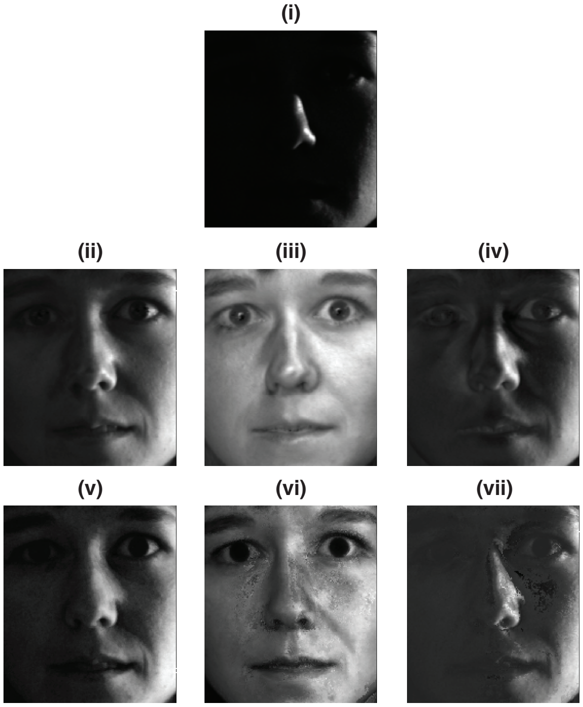

Figure 4: (i) original; (ii) PCPF; (iii) our algorithm; (iv) IRPCA-IHT; (v) PCP; (vi) fast RPCA; (vii) AltProj.

cal lighting conditions cannot be realised and there are unavoidable occlusion and albedo variations in real images. We demonstrate that there can be a substantial boost to the performance of facial denoising by leveraging dictionaries learnt from the images themselves.

The extended Yale B database is used as our observation which consists images under different illuminations for a fixed pose. We study all 64 images of a randomly chosen person. A 32556×64 observation matrix is formed by vectorizing each 168×192 image. For fast RPCA and our algorithm, a sparsity of 0.2 is adopted. We learn the feature dictionary as in (Xue, Panagakis, and Zafeiriou 2017). In a nutshell, the feature learning process can be treated as a sparse encoding problem. More specifically, we simultaneously seek a dictionary $\mathbf{D} \in \mathbb{R}^{n_1 \times c}$ and a sparse representation $\mathbf{B} \in \mathbb{R}^{c \times n_2}$ such that:

$$\begin{aligned} &\underset{\mathbf{D},\mathbf{B}}{\text{minimize}} \quad \|\mathbf{M} - \mathbf{DB}\|_F^2 \\ &\text{subject to} \quad \gamma_i \leq t \text{ for } i = 1 \dots n_2, \end{aligned} \tag{21}$$

where c is the number of atoms, γ_i's count the number of non-zero elements in each sparsity code and t is the sparsity constraint factor. This can be solved by the K-SVD algorithm. Here, feature $\mathbf{X}$ is the dictionary $\mathbf{D}$, feature $\mathbf{Y}$ corresponds to a similar solution using the transpose of the observation matrix as input. We set c to 40, t to 40 and used 10 iterations.

As a visual illustration, recovered images from all algorithms are exhibited in Figure 4. For this challenging scenario, our algorithm totally removed all shadows. PCPF is smoother than PCP but still suffers from shade. AltProj and fast RPCA both introduced extra artefacts. Although IRPCA-IHT managed to remove the shadows but brought back a severely distorted image. To quantitatively verify the improvement made by our proposed method, we examine the structural information contained within the denoised eigenfaces. Singular values of the recovered low-rank matrices from all algorithms are plotted in Figure 5. All nonconvex algorithms are competent in incorporating the rank information to keep only 9 singular values, vastly outperforming convex approaches. Among them, our algorithm has the most rapid decay that is found naturally (Wright et al. 2011).

Conclusion

This work proposes a new non-convex algorithm to solve RPCA with the help of features when the error sparsity is

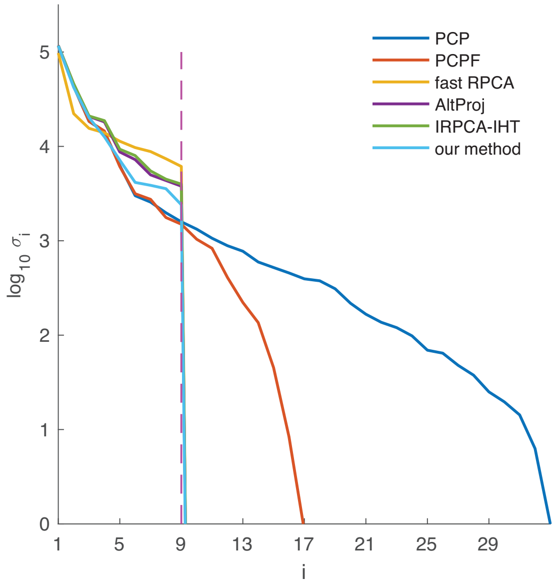

Figure 5: Log-scale singular values of the denoised matrices.

roughly known. Exact recovery guarantee has been established for three different assumptions about the incoherence conditions on features and the data observation matrix. Simulation experiments suggest that our algorithm is able to recover matrices of higher ranks corrupted by errors of higher sparsity than previous state-of-the-art approaches. Large synthetic matrices also show that our algorithm scales best with observation matrix dimension. MNIST and Yale B datasets further justify that our algorithm leads other approaches by a fair margin. Future work may involve finding a more accurate initialization scheme.

Acknowledgement

The work of Stefanos Zafeiriou was partially funded by EPSRC project EP/N007743/1 (FACER2VM).

References

Basri, R., and Jacobs, D. W. 2003. Lambertian reflectance and linear subspaces. *TPAMI* 25:218–233.

Candes, E. J.; Li, X.; Ma, Y.; and Wright, J. 2011. Robust principal component analysis? *Journal of the ACM* 58:11:1–11:37.

Chandrasekaran, V.; Sanghavi, S.; Parrilo, P. A.; and Willsky, A. S. 2011. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.* 21:572–596.

Chandrasekarana, V., and Jordan, M. I. 2013. Computational and statistical tradeoffs via convex relaxation. *PNAS* 110:E1181–E1190.

Chiang, K.; Hseih, C.; and Dhillon, I. 2015. Matrix completion with noisy side information. In *NIPS*.

Chiang, K.; Hsieh, C.; and Dhillon, I. 2016. Robust principal component analysis with side information. In *ICML*.

Ge, R.; Lee, J.; and Ma, T. 2016. Matrix completion has no spurious local minimum. In *NIPS*.

Gong, P.; Zhang, C.; Lu, Z.; Huang, J.; and Ye, J. 2013. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*.

Hsu, D.; Kakade, S. M.; and Zhang, T. 2011. Robust matrix decomposition with sparse corruptions. *IEEE Transactions on Information Theory*.

Kohler, J. M., and Lucchi, A. 2017. Sub-sampled cubic regularization for non-convex optimization. In *ICML*.

Liu, G.; Lin, Z.; and Yu, Y. 2010. Robust subspace segmentation by low-rank representation. In *ICML*.

Liu, G.; Liu, Q.; and Li, P. 2017. Blessing of dimensionality: Recovering mixture data via dictionary pursuit. *TPAMI* 39:47–60.

Netrapalli, P.; N, N. U.; Sanghavi, S.; Anandkumar, A.; and Jain, P. 2014. Non-convex robust pca. In *NIPS*.

Niranjan, U.; Rajkumar, A.; and Tulabandhula, T. 2017. Provable inductive robust pca via iterative hard thresholding. In *UAI*.

Oh, T.; Tai, Y.; Bazin, J.; Kim, H.; and Kweon, I. S. 2015. Partial sum minimization of singular values in robust pca: Algorithm and applications. *TPAMI* 38:744–758.

Sagonas, C.; Panagakis, Y.; Zafeiriou, S.; and Pantic, M. 2014. Raps: Robust and efficient automatic construction of person-specific deformable models. In *CVPR*.

Shang, F.; Cheng, J.; Liu, Y.; Luo, Z.; and Lin, Z. 2017. Bilinear factor matrix norm minimization for robust pca: Algorithms and applications. *TPAMI* PP:1–1.

Wright, J.; Ganesh, A.; Rao, S.; and Ma, Y. 2009. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *NIPS*.

Wright, J.; Ganesh, A.; Yang, A.; Zhou, Z.; and Ma, Y. 2011. Sparsity and robustness in face recognition. *arXiv:1111.1014*.

Xiong, H.; Liu, T.; and Tao, D. 2016. Diversified dynamical gaussian process latent variable model for video repair. In *AAAI*.

Xu, M.; Jin, R.; and Zhou, Z. 2013. Speedup matrix completion with side information: application to multi-label learning. In *NIPS*.

Xue, N.; Panagakis, Y.; and Zafeiriou, S. 2017. Side information in robust principal component analysis: Algorithms and applications. In *ICCV*.

Yi, X.; Park, D.; Chen, Y.; and Caramanis, C. 2016. Fast algorithms for robust pca via gradient descent. In *NIPS*.