RAP: Scalable RPCA for Low-rank Matrix Recovery

Chong Peng, Zhao Kang, Ming Yang, and Qiang Cheng
Southern Illinois University
Carbondale, IL, USA, 62901
{pchong, zhao.kang, ming.yang, qcheng}@siu.edu

ABSTRACT

Recovering low-rank matrices is a problem common in many applications of data mining and machine learning, such as matrix completion and image denoising. Robust Principal Component Analysis (RPCA) has emerged for handling such kinds of problems; however, the existing RPCA approaches are usually computationally expensive, due to the fact that they need to obtain the singular value decomposition (SVD) of large matrices. In this paper, we propose a novel RPCA approach that eliminates the need for SVD of large matrices. Scalable algorithms are designed for several variants of our approach, which are crucial for real world applications on large scale data. Extensive experimental results confirm the effectiveness of our approach both quantitatively and visually.

Keywords

RPCA, fixed rank, scalability, low-rank recovery

1. INTRODUCTION

High-dimensional data have been ubiquitously used in numerous tasks of machine learning and data mining, which is often challenging in real world applications. To alleviate this challenge and to explore putative, low-dimensional structures of such high-dimensional data, dimension reduction techniques have been studied extensively, among which principal component analysis (PCA) is widely used for unsupervised learning [6, 9, 10]. Classic PCA seeks the best rank- r estimate Z of a matrix M by solving $\min_{\text{rank}(Z) \leq r} \|M - Z\|$, with $\|\cdot\|$ denoting the spectral norm of a matrix. However, it suffers from poor estimation of Z when there are grossly corrupted entries in the data. As real world applications often encounter gross errors, it is highly desirable to design more robust PCA (RPCA) approaches. In the past decade, a number of methods have been proposed including, for example, alternating minimization [13], random sampling techniques [15], multivariate trimming, and so on [4].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'16, October 24 - 28, 2016, Indianapolis, IN, USA

© 2016 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-4073-1/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2983323.2983651>

More recently, a new type of RPCA method has emerged and become popular for many tasks [3, 20] with a general form:

$$\min_{Z, S} \text{rank}(Z) + \lambda \|S\|_0, \quad \text{s.t.} \quad X = Z + S, \quad (1)$$

where $\|\cdot\|_0$ is the ℓ_0 (pseudo) norm that counts the number of nonzero elements, and λ is a trade-off parameter that balances the rank of Z and the sparsity of S . It is assumed that the data matrix $X \in \mathcal{R}^{d \times n}$ can be decomposed into a low-rank component Z and a sparse one S , which is natural for many scenarios. For example, for images containing the same scene taken under different illuminating conditions, the scene is the low-rank component while shadows on these images represent the sparse components [1]. However, the optimization problem in (1) is NP-hard due to the minimization of the rank function and the ℓ_0 norm. In practice, their convex relaxations are often used with which (1) becomes

$$\min_{Z, S} \|Z\|_* + \lambda \|S\|_1, \quad \text{s.t.} \quad X = Z + S, \quad (2)$$

where $\|\cdot\|_*$ represents the nuclear norm defined as the summation of all singular values of a matrix, and $\|\cdot\|_1$ is the ℓ_1 norm that adds the absolute values of all elements of a matrix. It has been theoretically shown that Z and S can be exactly recovered by (2) with high probability, when nonzero entries of S are uniformly distributed in locations while Z and S satisfy some mild conditions [3]. Despite the convexity and ease of optimization, (2) may suffer from inaccurately recovering Z and S if the data are grossly corrupted or the underlying matrix does not have the incoherence property [3]. Another drawback of (2) [11] is that the nuclear norm is a biased estimator of the rank due to its similar behavior to ℓ_1 norm of a vector [7, 23]. It has been shown that some newly developed non-convex approximations to ℓ_0 norm [7, 21] can provide better estimation accuracy and variable selection consistency [19], and thus these approximations have drawn increasing attention in such applications as subspace clustering [12, 17, 18]. Recently, non-convex RPCA has been proposed by using a non-convex rank approximation [11]. However, it still has two potential limitations: First, to accurately separate the low-rank and sparse components, its model parameters need to be carefully tuned, which considers little prior knowledge on the desired rank of Z . Second, the singular value decomposition (SVD) of large matrices needs to be computed, which hinders the applications of RPCA to large-scale data due to the associated high complexity.

In this paper, to alleviate the above mentioned limitations,

we propose a novel RPCA approach, named Robust scalable (Ascendable) PCA (RAP), based on a factorization model which fixes the rank of Z by exploiting prior knowledge and avoids computing SVD of large matrices. Essentially important for real world applications, our method is scalable.

2. RELATED WORK

The convex approach to RPCA in (2) has been studied thoroughly in the literature. To incorporate the spatial connection information of the sparse elements, $\ell_{2,1}$ norm is introduced in (2) as follows [22]:

$$\min_{Z, S} \|Z\|_* + \lambda \|S\|_{2,1}, \quad s.t. \quad X = Z + S, \quad (3)$$

where $\|\cdot\|_{2,1}$ is defined to be the summation of ℓ_2 norms of all column vectors of a matrix. To overcome the above mentioned drawback of the nuclear norm, a non-convex RPCA model is proposed by replacing the nuclear norm in (3) with $\|Z\|_\gamma = \sum_i \frac{(1+\gamma)\sigma_i(Z)}{\gamma+\sigma_i(Z)}$ with $\gamma > 0$ and $\sigma_i(Z)$ being the i -th largest singular value of Z [11]. Both the non-convex and convex models in (2) and (3) need to tune some balancing parameters to accurately recover the low-rank component, which is time consuming and tedious, making these methods less systematic and automatic. Although an optimal choice of this parameter has been theoretically found for (2) [3], the conditions are not often satisfied in reality, and as a result Z with an exceedingly high rank may be obtained.

3. PROPOSED METHOD

Assume that X can be separated into a low-rank component Z and a sparse one S , i.e., $X = Z + S \in \mathcal{R}^{d \times n}$, then Z can be represented as $Z = LW$, where $L \in \mathcal{R}^{d \times r}$ is a basis matrix, $W \in \mathcal{R}^{r \times n}$ is a weight matrix, and r is the desired rank of Z . Therefore, each column of Z can be constructed as a linear combination of the basis vectors. Inspired by (1), we formulate the following model:

$$\min_{L, S, W} \|S\|_1 \quad s.t. \quad X = LW + S. \quad (4)$$

With this formulation, we can see that the rank of the low-rank component is at most r by matrix factorization and no balancing parameter is needed. (4) may lead to different variants for various applications. To derive these variants, we consider a specific application of RPCA, background-foreground separation. In general, there are three cases: single background, multiple suddenly changing backgrounds, and multiple gradually changing backgrounds. For these cases, we discuss the corresponding variants of (4) in the following.

In the case of a single background, we have $r = 1$, and (4) can be reformulated as

$$\min_{L, S} \|S\|_1, \quad s.t. \quad X_i = L + S_i, \quad (5)$$

where M_i is the i -th column of M with $M = X$ or S . With (5) we assume that all images can be separated as combinations of one common background and the varying foregrounds.

In the case of multiple backgrounds with sudden changes,

we have $r > 1$, and (4) can be expressed as

$$\begin{aligned} \min_{L, S, W} \|S\|_1 \\ s.t. \quad X = LW + S, W \in \{0, 1\}^{n \times r}, \sum_{j=1}^r W_{ij} = 1, \end{aligned} \quad (6)$$

by which we assume that each frame can be separated as a combination of one background and one foreground. It is noted that (5) is a special case of (6), where W in the former is an all-1 row vector of length n .

In the case that there are gradual changes in the backgrounds, (4) can be formulated as

$$\min_{L, S, W} \|S\|_1 \quad s.t. \quad X = LW + S, W_{ij} \geq 0, \sum_{i=1}^r W_{ij} = 1, \quad (7)$$

where W takes continuous rather than binary values as in (6). The optimization of (5), (6), and (7) will be developed in Section 4. It is noted that models (5) and (6) are special cases of (7) by restricting W values.

4. OPTIMIZATION

Because (5) and (6) are special cases of (7), for clarity of presentation, we will firstly derive the optimization of the general model (7), then we will specialize the optimization procedures of the other models based on that of (7).

4.1 Optimization of model (7)

The augmented Lagrange function of (7) is

$$\begin{aligned} \mathcal{L}(L, S, W, \Theta, \rho) = & \|S\|_1 + Tr(\Theta^T(X - LW - S)) \\ & + \frac{\rho}{2} \|X - LW - S\|_F^2, \quad s.t. \quad W \geq 0, \sum_{i=1}^r W_{ij} = 1. \end{aligned} \quad (8)$$

First, the subproblem for optimizing W is

$$\begin{aligned} \min_W Tr(\Theta^T(X - LW - S)) + \frac{\rho}{2} \|X - LW - S\|_F^2, \\ s.t. \quad W \geq 0, \sum_{i=1}^r W_{ij} = 1, \end{aligned} \quad (9)$$

The above optimization problem is equivalent to solving the following optimization problems for each column of W :

$$\min_w \frac{1}{2} w^T Q w - y^T w, \quad s.t. \quad w_j \geq 0, \sum_{j=1}^r w_j = 1, \quad (10)$$

where $Q := L^T L$, $y := L^T(X_i - S_i + \Theta_i/\rho)$, and $w := W_i$. (10) is a standard quadratic programming (QP) problem and has been studied extensively. We directly make use of the ‘quadprog’ function in Matlab with the default interior point method to solve for w . Then L , S , Θ , and ρ are updated as follows:

$$L = (X - S + \Theta/\rho)W(WW^T)^{-1}, \quad (11)$$

$$S = \mathcal{T}_{\frac{1}{\rho}}(X - LW - S + \Theta/\rho), \quad (12)$$

$$\Theta = \Theta + \rho(X - LW - S), \quad \rho = \rho\kappa, \quad (13)$$

where $[\mathcal{T}_\alpha(D)]_{ij} = \max(|D_{ij}| - \alpha, 0)\text{sign}(D_{ij})$, and $\kappa > 1$ controls the increasing speed of ρ .

It is noted that, due to the nonnegative constraints on W in (9), there is no analytic solution of W for this sub-

problem, which requires higher computational cost because inner iterations are needed. To speed it up, we propose a relaxed formulation of (7), where we drop the nonnegative constraints on W as follows:

$$\min_{L, S, W} \|S\|_1 \quad s.t. \quad X = LW + S, \quad \sum_{i=1}^r W_{ij} = 1. \quad (14)$$

We express (14) in a column manner as follows:

$$\min_w \frac{1}{2} w^T Q w - y^T w, \quad s.t. \quad \sum_{j=1}^r w_j = 1. \quad (15)$$

The Lagrange function of (15) is

$$J(w, \lambda) = \frac{1}{2} w^T Q w - y^T w + \lambda \left(\sum_{j=1}^r w_j - 1 \right), \quad (16)$$

where λ is the Lagrange multiplier. The optimal solution is obtained by setting $\nabla J(w, \lambda) = 0$. Therefore, we have

$$\begin{bmatrix} w \\ \lambda \end{bmatrix} = \begin{bmatrix} 2Q & \mathbf{1}_r \\ \mathbf{1}_r^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} y \\ 1 \end{bmatrix}, \quad (17)$$

where $\mathbf{1}_r$ is an all-1 column vector of length r . Therefore, the optimal W can be obtained as

$$W = \left[\begin{bmatrix} 2Q & \mathbf{1}_r \\ \mathbf{1}_r^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} L^T(X - S + \frac{1}{\rho}\Theta) \\ \mathbf{1}_n^T \end{bmatrix} \right]_{1:r}, \quad (18)$$

where $[\cdot]_{1:r}$ denotes the first r rows of a matrix. For L , S , Θ and ρ , the updating rules follow the updating of (7).

4.2 Optimization of model (5) / (6)

As special cases of (7), the optimization of (5) and (6) may follow the above subsection. To be more efficient, we consider the special constraints and update W as follows:

$$[W]_{ij} = \begin{cases} 1, & \text{if } i = \operatorname{argmin}_{1 \leq t \leq r} \| [X - S + \frac{\Theta}{\rho}]_j - L_t \|_2^2 \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

By simple algebra, due to the special structure of W , L is equivalently updated column-wisely as

$$L_i = \frac{1}{\sum_j W_{ij}} \sum_{(i,j) \in \{(i,j) | W_{ij}=1\}} (X_j - S_j + \Theta_j / \rho), \quad (20)$$

for $i = 1, 2, \dots, r$. It is noted that, for (5), calculating W and L is more straightforward with the following updating:

$$W = \mathbf{1}_n, \quad L = \frac{1}{n} \sum_{i=1}^n (X_i - S_i + \Theta_i / \rho). \quad (21)$$

For notational convenience, we call the optimization procedures corresponding to (5), (6), (7), and (14) as RAP-S, RAP-MS, RAP-MG, and RAP-MGR, respectively.

4.3 Complexity

Given that r is fixed, it is straightforward to verify that RAP-S, RAP-MS, and RAP-MGR all have a complexity of $O(dn)$. For RAP-MG, each sub-QP problem has a constant cost; hence, solving (9) has a complexity of $O(n)$. Therefore the total complexity of RAP-MG is $O(dn)$.

5. EXPERIMENTS

Three important practical applications are used to evaluate our method: foreground-background separation on surveillance video, shadow removal from human face images, and anomaly detection. We compare our method with some state-of-the-art methods, including convex RPCA [3], and AltProj [16]¹. Specifically, we use inexact ALM (IALM)² [3, 14] to solve convex RPCA. IALM and AltProj make use of the PROPACK package [5] to solve SVDs.

5.1 Foreground-background separation

It is a popular way to detecting moving objects or interesting activities in a scene by removing background(s) from a video sequence. In this test, we choose a set of video datasets including Escalator Airport, Hall Airport, Bootstrap, Campus, Fountain, Water Surface, Shopping Mall, Curtain, Office, PETS2006, Pedestrian, Highway, Lobby, Camera Parameter, Light Switch-1, Light Switch-2, and Time Of Day. These data sets can be divided into two scenarios: with a single background (the first 12) and with multiple backgrounds (the last five)³. For all these datasets, $X \in \mathcal{R}^{d \times n}$ are constructed by reshaping each frame into a column vector.

In the single-background case, we compare RAP-S with IALM and AltProj on the first 12 datasets. For RAP-S, we fix $\rho = 0.001$, $\kappa = 1.5$, and $r = 1$ for all the datasets. In a way similar to AltProj, we vanish those small values in $(-0.001, 0.001)$ for S . For IALM, we use the default balancing parameter given in the corresponding paper. Besides, we fix $\rho = 10^{-4}$ and $\kappa = 1.5$ for IALM, which results in a good visual quality and faster convergence. For AltProj, we fix $r = 1$ and use the other default parameters. For all these methods, we terminate the iteration when $\frac{\|X - Z - S\|_F}{\|X\|_F} \leq 10^{-3}$ is satisfied as in [16]. We compare these methods in terms of the rank of Z , the density of S defined to be ratio of nonzero elements in S , the relative fitting error $\frac{\|X - Z - S\|_F}{\|X\|_F}$, total number of iterations as well as that of (partial) SVDs⁴, and computational time. The results are shown in Table 1, which are all obtained on a dual-core Intel Xeon E3-1240 V2 3.40 GHz Linux Server with 8 GB memory. It is observed that IALM has the best performance in sparsity of S . However, it results in higher ranks of Z and takes much longer time than both AltProj and RAP-S. AltProj results in Z with the desired rank and has generally smaller fitting errors. However, AltProj needs more iterations than IALM and RAP-S, and the latter two can achieve lower fitting errors given the same number of iterations. Both IALM and AltProj need to solve SVDs at each iteration, which are computationally expensive. For RAP-S, it results in the desired rank for Z ⁵; it has comparable or more sparsity on S than AltProj, and has comparable fitting error to IALM. The most important is that RAP-S does not need to solve

¹ available at <http://www.personal.psu.edu/nsa10/codes.html>.

² available at <http://perception.csl.illinois.edu/matrix-rank/sample-code.html#RPCA>.

³ All datasets used in this subsection are available at: http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html, <http://limu.ait.kyushu-u.ac.jp/dataset/en/>, <http://wordpress-jodoin.dmi.usherb.ca/dataset2012/>, <http://research.microsoft.com/en-us/um/people/jckrumm/wallflower/testimages.htm>.

⁴ The reported number is the number of partial SVDs, because IALM and AltProj solve partial SVDs with PROPACK package.

⁵ For RAP, we use Z to denote LW .

Table 1: Results over Datasets with Single Background

Data	Method	Rank(Z)	$\ S\ _0/(d \times n)$	$\frac{\ X-Z-S\ _F}{\ X\ _F}$	Iterations	# of SVDs	Time
Hall Airport	AltProj	1	0.9806	5.16e-5	46	47	113.98
	IALM	556	0.8500	8.15e-4	11	12	408.03
	RAP-S	1	0.9194	6.93e-4	17	0	28.54
Highway	AltProj	1	0.9331	2.96e-4	37	38	49.65
	IALM	539	0.8175	6.02e-4	12	13	269.10
	RAP-S	1	0.9242	8.34e-4	17	0	15.70
Office	AltProj	1	0.8018	9.40e-4	51	52	84.43
	IALM	374	0.7582	9.46e-4	11	12	230.53
	RAP-S	1	0.8702	1.00e-3	17	0	20.89
PETS2006	AltProj	1	0.8590	5.20e-4	35	36	44.64
	IALM	293	0.8649	5.63e-4	12	13	144.26
	RAP-S	1	0.8655	6.06e-4	18	0	16.11
Shopping Mall	AltProj	1	0.9853	3.91e-5	45	46	45.35
	IALM	328	0.8158	9.37e-4	11	12	123.99
	RAP-S	1	0.9084	9.24e-4	17	0	12.80
Pedestrian	AltProj	1	0.5869	9.32e-4	41	42	37.90
	IALM	35	0.8910	5.69e-4	11	12	36.18
	RAP-S	1	0.6490	7.01e-4	17	0	11.36
Bootstrap	AltProj	1	0.9747	1.17e-4	44	45	107.15
	IALM	1146	0.8095	6.27e-4	12	13	1182.92
	RAP-S	1	0.9284	9.09e-4	17	0	27.82
Water Surface	AltProj	1	0.8890	3.97e-4	47	48	27.27
	IALM	224	0.7861	5.32e-4	12	13	51.00
	RAP-S	1	0.9021	4.60e-4	18	0	6.71
Campus	AltProj	1	0.9790	9.50e-5	41	42	54.1
	IALM	488	0.8136	9.30e-4	11	12	242.59
	RAP-S	1	0.9353	7.35e-4	17	0	14.30
Curtain	AltProj	1	0.8280	7.46e-4	40	41	102.41
	IALM	834	0.7398	6.84e-4	12	13	747.36
	RAP-S	1	0.8968	5.9e-4	18	0	30.57
Fountain	AltProj	1	0.9113	2.91e-4	50	51	23.90
	IALM	102	0.8272	4.91e-4	12	13	25.62
	RAP-S	1	0.8589	5.19e-4	18	0	5.44
Escalator Airport	AltProj	1	0.9152	2.29e-4	40	41	110.75
	IALM	957	0.7744	7.76e-4	11	12	1,040.91
	RAP-S	1	0.8817	6.82e-4	17	0	33.94

Figure 1: Foreground-background separation in the Highway video. The top left is the original frame and the rest are extracted background (top) and foreground (bottom).

any SVD and the speed is around 4 times faster than AltProj and at least 10 times faster than IALM on most of the datasets. At the same time, RAP-S also has good visual quality. To illustrate this, we show some sample frames and resultant Z and S in Fig. 1 of Highway. It is observed that AltProj and RAP-S have comparable performance visually, where the foregrounds are well separated from the background. However, the low rank component by IALM still contains artifacts from the sparse component. For example, in Fig. 1, two cars are not completely removed from the background in the upper part of the image.

In the case of multiple backgrounds, we compare RAP-MS, RAP-MG, and RAP-MGR with AltProj and IALM, following the same parameter settings discussed above. We show the quantitative results and sample visual results of Light Switch-2 dataset in Table 2 and Fig. 2, respectively, which are all obtained on a six-core Intel Xeon E5-2620 V3 2.40 GHz Windows Server with 32 GB memory. We have similar observations to the single-background case. For RAP-MG, we parallel the computation of W at each iteration using 4 cores; however, RAP-MGR has almost the same

Table 2: Results over Datasets with Multiple Backgrounds

Data	Method	Rank(Z)	$\ S\ _0/(d \times n)$	$\frac{\ X-Z-S\ _F}{\ X\ _F}$	Iterations	# of SVDs	Time
Lobby	AltProj	2	0.9243	1.88e-4	39	41	68.53
	IALM	223	0.8346	6.19e-4	12	13	205.83
	RAP-MS	2	0.8597	6.54e-4	18	0	30.27
	RAP-MG	2	0.8523	7.28e-4	18	0	48.03
	RAP-MGR	2	0.8521	7.28e-4	18	0	27.77
Light Switch-1	AltProj	2	0.9600	1.12e-4	55	57	180.28
	IALM	499	0.6737	9.99e-4	11	12	497.41
	RAP-MS	2	0.9649	6.21e-4	17	0	43.93
	RAP-MG	2	0.9156	9.35e-4	17	0	77.07
	RAP-MGR	2	0.9158	9.31e-4	17	0	43.88
Light Switch-2	AltProj	2	0.9050	2.24e-4	47	49	129.31
	IALM	591	0.7921	7.93e-4	12	13	926.97
	RAP-MS	2	0.8319	7.81e-4	18	0	49.19
	RAP-MG	2	0.8213	9.21e-4	18	0	80.73
	RAP-MGR	2	0.8216	9.22e-4	18	0	45.86
Camera Parameter	AltProj	2	0.8806	5.34e-4	47	49	129.20
	IALM	607	0.7750	6.86e-4	12	13	647.85
	RAP-MS	2	0.8170	5.48e-4	18	0	45.77
	RAP-MG	2	0.8460	7.61e-4	18	0	73.39
	RAP-MGR	2	0.8460	7.61e-4	18	0	42.05
Time Of Day	AltProj	2	0.8646	4.72e-4	44	46	90.02
	IALM	351	0.6990	6.12e-4	13	14	384.61
	RAP-MS	2	0.8132	7.40e-4	18	0	35.04
	RAP-MG	2	0.8383	6.47e-4	19	0	61.12
	RAP-MGR	2	0.8383	6.47e-4	19	0	34.92

RAP-MG needs to solve SVD (matrix inverse) of size $(r+1) \times (r+1)$, which in our experiments is 3×3 and can be ignored compared to AltProj and IALM.

Table 3: Recovery Results of Face Data

Data	Method	Rank(Z)	$\ S\ _0/(d \times n)$	$\frac{\ X-Z-S\ _F}{\ X\ _F}$	Iterations	# of SVDs	Time
Subject 1	AltProj	1	0.9553	8.18e-4	50	51	4.20
	IALM	32	0.7745	6.28e-4	25	26	2.39
	RAP-S	1	0.9898	9.78e-4	8	0	0.40
Subject 2	AltProj	1	0.9755	2.34e-4	49	50	5.00
	IALM	31	0.7656	6.47e-4	25	26	2.66
	RAP-S	1	0.9918	5.72e-4	9	0	0.45

performance as RAP-MG only with a much faster speed. Additionally, to investigate the recovery performance with the gradually changing backgrounds, we show two original frames of Time Of Day video and the recovery results in Fig. 3. It is observed that AltProj and RAP-MS fail to recover the correct background for the first frame while RAP-MG and -MGR succeed. All methods produce visually good results on the second frame.

5.2 Shadow removal from face images

Face recognition is an important task in pattern recognition and computer vision, but it is usually challenging when face images are taken under varying lighting conditions [2]. It is crucial to handle shadows, peculiarities and saturations on face images, which can be achieved by applying RPCA. Extended Yale B (EYaleB) dataset [8] contains face images of 38 persons, each of whom has 64 heavily corrupted images taken under different lighting conditions. We choose two of the subjects (subject 1 and subject 2) and convert all face images of size 192×168 to 32,256-dimensional vectors. For each subject, a data matrix $X \in \mathcal{R}^{32,256 \times 64}$ is constructed. It is natural to assume that the face images reside in a rank one space, hence we set $r = 1$. The quantitative and visual results are given in Table 3 and Fig. 4, respectively. Fig. 4 demonstrates that the low-rank component recovered by IALM still has heavy shadows, whereas those by AltProj and RAP-S do not. $\rho = 10$ and $\kappa = 1.5$ are used for RAP-S.

5.3 Anomaly Detection

Another important application of RPCA is anomaly detection. If we inject several images of one subject into a dominating set of images of another subject, then the newly

Figure 2: Foreground-background separation in the Light Switch-2 video. From top to bottom are two frames and extracted results. For each method, from left to right are the extracted background and dynamic foreground, respectively.

Figure 3: Foreground-background separation in the Time Of Day video. From top to bottom are two frames and extracted results. For each method, from left to right are the extracted background and dynamic foreground, respectively.

Figure 4: Shadow removal results on EYaleB data. For each of the two parts, from left to right are original data, results of (1) IALM, (2) AltProj, and (3) RAP-S, respectively. The top panel are the original image and low rank components, while the bottom are the sparse components.

Figure 5: Selected '1's and '7's from USPS dataset.

Figure 6: Anomaly identified by RAP-S, including written '1's and outliers '7's.

added images are outliers, or anomaly, and the dominant images form a low-dimensional subspace. The task is to detect all anomalies. USPS dataset contains 9,298 images of hand-written digits 0-9 with size 16×16 . We choose the first 190 images of '1' and the last 10 images of '7' as shown in Fig. 5 and align the vectorized images into the data matrix. In this case, '1' is the dominant digit and '7's are outliers. Anomaly detection is not only to detect the '7's, but also the anomaly of hand-written '1's. For the anomaly in '1' images or outliers '7', the corresponding columns in S contain relatively larger values. We simply use ℓ_2 norm as a measure of these values, then identify the outliers by finding the columns of S with the largest values of ℓ_2 norms [11]. Fig. 7 shows the ℓ_2 norms of the columns of S . For ease of visualization, values that are smaller than 4 are made zeros. Totally, seven anomalies in '1's and all '7's are detected. The identified anomalies or outliers are shown in Fig. 6. $r = 1$, $\rho = 0.001$ and $\kappa = 1.5$ are used for RAP-S.

Figure 7: ℓ_2 -norms of each row of S .

5.4 Scalability

To numerically illustrate the scalability of RAP algorithms⁶, we show the time cost versus the values of n and d , respectively, in Fig. 8. For all these algorithms, we temporarily ignore the terminating tolerance and terminate them within 50 iterations with $\rho = 0.001$ and $\kappa = 1.5$. The figures empirically verify that RAP algorithms have a linear complexity in n or d and desirable scalability.

6. CONCLUSION

This paper proposes a scalable RPCA method (RAP) as well as its several variants for specific applications. The proposed method and all variants have linear complexity in both data size and dimension, which renders it possible to have RPCA applications on large scale data. Besides, the desired low rank of Z can be guaranteed, which is more systematic than other approaches that need to carefully tune balancing parameters. Quantitatively, RAP not only has the fastest speed but also comparable performance in other measures. Visually, RAP demonstrates better quality than other state-of-the-art methods under various conditions.

Acknowledgment

Qiang Cheng is the corresponding author. This work is supported by National Science Foundation under grant IIS-1218712, National Natural Science Foundation of China, under grant 11241005, and Shanxi Scholarship Council of China 2015-093, Fund Program for the Scientific Activi-

⁶It is suggested to use RAP-MGR instead of RAP-MG in Section 5.1, therefore, we only show the results of RAP-S, RAP-MS, and RAP-MGR.

Figure 8: Plots of time cost of RAP-S, RAP-MS, and RAP-MGR as functions of data size in (a)-(c) and dimension in (d)-(f), respectively.

ties of Selected Returned Overseas Professionals in Shanxi Province.

References

- [1] N. S. Aybat, D. Goldfarb, and G. Iyengar. Fast first-order methods for stable principal component pursuit. *arXiv preprint arXiv:1105.2126*, 2011.
- [2] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.
- [3] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.
- [4] C. Croux and G. Haesbroeck. Principal component analysis based on robust estimators of the covariance or correlation matrix: influence functions and efficiencies. *Biometrika*, 87(3):603–618, 2000.
- [5] X. Ding, L. He, and L. Carin. Bayesian robust principal component analysis. *Image Processing, IEEE Transactions on*, 20(12):3419–3430, 2011.
- [6] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [7] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.
- [8] A. S. Georgiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):643–660, 2001.
- [9] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- [10] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [11] Z. Kang, C. Peng, and Q. Cheng. Robust pca via nonconvex rank approximation. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 211–220. IEEE, 2015.
- [12] Z. Kang, C. Peng, and Q. Cheng. Robust subspace clustering via tighter rank approximation. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 393–401. ACM, 2015.
- [13] Q. Ke and T. Kanade. Robust l_1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 739–746. IEEE, 2005.
- [14] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.
- [15] R. Maronna, D. Martin, and V. Yohai. *Robust statistics*. John Wiley & Sons, Chichester. ISBN, 2006.
- [16] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain. Non-convex robust pca. In *Advances in Neural Information Processing Systems*, pages 1107–1115, 2014.
- [17] C. Peng, Z. Kang, H. Li, and Q. Cheng. Subspace clustering using log-determinant rank approximation. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 925–934. ACM, 2015.
- [18] C. Peng, Z. Kang, M. Yang, and Q. Cheng. Feature selection embedded subspace clustering. *IEEE Signal Processing Letters*, 23(7):1018–1022, July 2016.
- [19] Z. Wang, H. Liu, and T. Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *Annals of statistics*, 42(6):2164, 2014.
- [20] 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*, pages 2080–2088, 2009.
- [21] S. Xiang, X. Tong, and J. Ye. Efficient sparse group feature selection via nonconvex optimization. In *ICML (1)*, pages 284–292, 2013.
- [22] H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.
- [23] C.-H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, pages 894–942, 2010.