EFFICIENT LOW RANK MATRIX APPROXIMATION VIA ORTHOGONALITY PURSUIT AND ℓ^2 REGULARIZATION

Siyuan Li[‡] *Jiawan Zhang*[‡§] *Xiaojie Guo*[†*]

[‡] School of Computer Software, Tianjin University [†] State Key Lab of Information Security, IIE, CAS
[§] School of Computer Science, Qinghai Normal University

ABSTRACT

Low rank matrix approximation, in the presence of missing data and outliers, has previously shown its significance as a theoretic foundation in a wide spectrum of tabulated information processing applications. To fit low rank models, minimizing the nuclear norm of matrices is a popular scheme, the computational load of which, however, is heavy. While bilinear factorization can largely mitigate the computational complexity. Unfortunately, without a known or precisely estimated target rank, this strategy often performs vulnerably when the given data is dirty. This paper attempts to simultaneously achieve the computational efficiency as well as the robustness to mild rank initialization and gross corruptions. Moreover, several Augmented Lagrange Multiplier based solvers and a heuristic rank estimator are customized to seek the optimal solution. Theoretical analysis on convergence and complexity, and experiments on both synthetic and real data are provided to reveal the efficacy of our method and show its superiority over the state-of-the-art alternatives.

***Index Terms*—** Low Rank Matrix Approximation, Orthogonality Pursuit, ℓ^2 Regularization

1. INTRODUCTION

It has been recognized that even very high-dimensional observations should have a low-dimensional structure. In real-world tasks, however, we often have to face handling dirty observations, say incomplete and/or noisy data, that likely destroy the intrinsic low-dimensional structure. In last decades, Low Rank Matrix Approximation (LRMA), which aims to learn a low-dimensional model from given observations in the presence of missing data and noises, has been focus of research in various fields. Many tasks can be understood as its examples, like background modeling [1] and denoising [2].

Mathematically, LRMA has the following general shape:

$$\textit{General LRMA} \quad \min_{\mathbf{L}} \operatorname{rank}(\mathbf{L}) + \lambda f(\mathbf{X}-\mathbf{L}), \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ denotes the observation matrix that contains n samples, each of which is an m-dimensional measure, and $\mathbf{L} \in \mathbb{R}^{m\times n}$ the desired low rank recovery. In addition, $f(\cdot)$ is a penalty on the residual between the observed and recovered signals, $\operatorname{rank}(\cdot)$ is the rank function and λ is a non-negative parameter that provides a trade-off between the recovery fidelity and the low-rank promoting regularization.

The loss function $f(\cdot)$ is usually in the form of $\|\mathbf{W}\odot(\mathbf{X}-\mathbf{L})\|$, where $\|\cdot\|$ designates a certain matrix norm to measure the error, and $\mathbf{W} \in \{0,1\}^{m\times n}$ is an indicator matrix. The square loss (*a.k.a.* ℓ^2 loss) is arguably one of the most commonly used penalties, which is optimal to Gaussian noises. But, the square loss lacks robustness to outliers that are not unusual to find in real data. From the definition of ℓ^0 norm, it is the "ideal" option for being robust against gross corruptions, as it ignores the scale of the outliers. However, the associated model becomes impractical because of its non-convexity and discontinuity. A common solution is to adopt its convex proxy, say the ℓ^1 norm. The choice of the error model is critical to different tasks, but not the main focus of this work. For the sake of robustness to outliers, in this paper, we will merely consider $f(\mathbf{X}-\mathbf{L}) := \|\mathbf{W}\odot(\mathbf{X}-\mathbf{L})\|_1$.

The rank constraint, due to its intractability, is typically relaxed by its convex surrogate, *i.e.* the nuclear norm. As a consequence, the corresponding problem becomes:

$$\textit{NNM Model} \quad \min_{\mathbf{L}} \|\mathbf{L}\|_* + \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{L})\|_1. \tag{2}$$

Nuclear Norm Minimization (NNM) methods can perform stably without knowing the target rank of recovery in advance. The computational bottleneck of NNM approaches comes from the necessity of executing expensive SVD for multiple times. Alternatively, at (much) less expense, Bilinear Factorization (BF) can achieve the goal by solving:

$$\textit{BF Model} \quad \min_{\mathbf{U},\mathbf{V}} \|\mathbf{W}\odot(\mathbf{X}-\mathbf{U}\mathbf{V}^T)\|_1, \tag{3}$$

where the product of two smaller factor matrices $\mathbf{U} \in \mathbb{R}^{m\times r}$ (left factor) and $\mathbf{V} \in \mathbb{R}^{n\times r}$ (right factor) implicitly guarantees that the rank of $\mathbf{U}\mathbf{V}^T$ is never over r. This factorization strategy can greatly release the pressure of computation and provide accurate results when the target rank is given. But, in many tasks, the target rank is unknown beforehand. In such a situation, the performance of BF would sharply degrade due to its sensitivity to the guess of target rank.

* Corresponding Author. This work was supported by grants (15XMZ057, 2015-ZJ-718, 61402467).

978-1-5090-6067-2/17/$31.00 ©2017 IEEE

This paper proposes a novel method to solve LRMA (1) in a matrix factorization fashion, which considers the efficiency, as well as the robustness to rough rank initialization and gross corruptions. Specifically, we first show the bridge between NNM (2) and BF (3) to see the potential of efficiency improvement. Next, an orthogonal prior on the left factor matrix is introduced to simplify the model and shrink the solution space. The robustness to rank initialization is achieved by an adaptive thresholding strategy on (the update of) the right factor. Furthermore, several algorithms are customized to effectively find the optimal solution of the proposed model for the cases with and without target rank known. Theoretical analysis on convergence and complexity is provided. To reveal the efficacy and the superior performance of the proposed model over the state-of-the-arts, experiments on both synthetic and real data are conducted.

2. METHODOLOGY

As aforementioned, the model (2) involves expensive SVDs on the entire data, which are desired to be replaced by cheaper ones. Theorem 1 bridges the models (2) and (4), and thus makes such a replacement possible.

Theorem 1. *For any matrix* $\mathbf{L} \in \mathbb{R}^{m\times n}$*, the following relationship holds [3]:*

$$\|\mathbf{L}\|_* = \min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 \ \ \mathrm{s.t.}\ \ \mathbf{L} = \mathbf{U}\mathbf{V}^T.$$

If $\mathrm{rank}(\mathbf{L}) = r \leq \min(m,n)$*, then the minimum solution above is attained at a factor decomposition* $\mathbf{L} = \mathbf{U}\mathbf{V}^T$*, where* $\mathbf{U} \in \mathbb{R}^{m\times r}$ *and* $\mathbf{V} \in \mathbb{R}^{n\times r}$*.*

As a consequence, applying Theorem 1 on (2) reads:

$$\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{U}\mathbf{V}^T)\|_1, \quad (4)$$

which is the model proposed by Cabral *et al.* in [4]. We call this model Unifying for simplicity. Compared the Unifying model (4) with the BF (3), there are two extra terms, namely $\frac{1}{2}\|\mathbf{U}\|_F^2$ and $\frac{1}{2}\|\mathbf{V}\|_F^2$, which make the solution more identifiable and avoid over-fitting. For further shrinking the solution space, it is reasonable to constrain $\mathbf{U}$ to be column-orthogonal, *i.e.* $\mathbf{U}^T\mathbf{U} = \mathbf{I}$.[1] Incorporating the orthogonality of $\mathbf{U}$ leads to the optimization problem as:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} \frac{1}{2}\|\mathbf{U}\|_F^2 + \frac{1}{2}\|\mathbf{V}\|_F^2 + \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{U}\mathbf{V}^T)\|_1 \\ \mathrm{s.t.}\ \ \mathbf{U}^T\mathbf{U} = \mathbf{I}.\end{aligned} \quad (5)$$

From (5), we can observe that, owning to the constraint $\mathbf{U}^T\mathbf{U} = \mathbf{I}$ and the relationship of $\|\mathbf{U}\|_F^2 = \mathrm{trace}(\mathbf{U}^T\mathbf{U})$, the first term in the objective function (5) becomes a constant. Hence the model can be further simplified as follows:

$$\min_{\mathbf{U}\in\{\mathbf{U}^T\mathbf{U}=\mathbf{I}\},\mathbf{V}} \frac{1}{2}\|\mathbf{V}\|_F^2 + \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{U}\mathbf{V}^T)\|_1. \quad (6)$$

The above formulation can be interpreted as seeking a low-dimensional subspace from the given data by imposing ℓ^2 regularization on the right factor matrix and orthogonality pursuit on the left, such that the fidelity between the observed and recovered signals is minimized with respect to the sparsity-inducing optimality criterion.

2.1. Solver with Rank Known

In past years, it has been shown in the literature that ALM-ADM [5] is very efficient for some convex or non-convex optimization problems. To apply ALM-ADM on our problem, we need to make the objective function separable. Therefore, an auxiliary variable $\mathbf{K} \in \mathbb{R}^{m\times n}$ is introduced to replace $\mathbf{U}\mathbf{V}^T$ in the fidelity term of (6). Accordingly, $\mathbf{K} = \mathbf{U}\mathbf{V}^T$ performs as the additional constraint. Hence, we have:

$$\min_{\mathbf{U}^T\mathbf{U}=\mathbf{I}} \frac{1}{2}\|\mathbf{V}\|_F^2 + \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{K})\|_1 \ \mathrm{s.t.}\ \mathbf{K} = \mathbf{U}\mathbf{V}^T. \quad (7)$$

The augmented Lagrangian function of (7) can be naturally written as follows:

$$\begin{aligned}\mathcal{L}_{\{\mathbf{U}^T\mathbf{U}=\mathbf{I}\}}(\mathbf{U},\mathbf{V},\mathbf{K}) := \frac{1}{2}\|\mathbf{V}\|_F^2 + \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{K})\|_1 \\ +\frac{\mu}{2}\|\mathbf{K}-\mathbf{U}\mathbf{V}^T\|_F^2 + \langle\mathbf{Z},\mathbf{K}-\mathbf{U}\mathbf{V}^T\rangle,\end{aligned}$$

where μ is a positive penalty scalar and $\mathbf{Z} \in \mathbb{R}^{m\times n}$ is a Lagrangian multiplier. The designed solver iteratively updates one variable at a time by fixing the others.

Updating $\mathbf{U}$. Dropping and adding some proper terms unrelated to $\mathbf{U}$, the $\mathbf{U}$ sub-problem is as below:

$$\mathbf{U}^{(t+1)} = \operatorname*{argmin}_{\mathbf{U}\in\{\mathbf{U}^T\mathbf{U}=\mathbf{I}\}} \|\mathbf{U}\mathbf{V}^{(t)T} - \mathbf{D}^{(t)}\|_F^2, \quad (8)$$

with the definition $\mathbf{D}^{(t)} := \mathbf{K}^{(t)} + \mathbf{Z}^{(t)}/\mu^{(t)}$. The optimal solution can be given by the SVD of $\mathbf{D}^{(t)}\mathbf{V}^{(t)}$. To avoid heavy computational load, we adopt the idea in [6] that alternatively resorts to the QR decomposition. The equivalence between using SVD and QR in such an iterative scheme can be easily established as Theorem 7 given in [7] with a simple modification on $\mathbf{V}$ sub-problem. So, updating $\mathbf{U}$ can be done by:

$$[\mathbf{Q},\mathbf{R}] \leftarrow \mathrm{qr}\left(\mathbf{D}^{(t)}\mathbf{V}^{(t)}\right), \quad \mathbf{U}^{(t+1)} \leftarrow \mathbf{Q}, \quad (9)$$

where $\mathbf{U}^{(t+1)}$ is an orthogonal basis for the range space of $\mathbf{D}^{(t)}\mathbf{V}^{(t)}$.

Updating $\mathbf{V}$. The associated problem turns out to be like:

$$\mathbf{V}^{(t+1)} = \operatorname*{argmin}_{\mathbf{V}} \|\mathbf{V}\|_F^2 + \mu^{(t)}\|\mathbf{U}^{(t+1)}\mathbf{V}^T - \mathbf{D}^{(t)}\|_F^2, \quad (10)$$

[1] With a much smaller solution space, the computational cost would be remarkably cut. But, in some cases, the optimal solution to problem (4) may be excluded in the presence of heavy outliers. So a remedy to this issue is desirable, which will be discussed later.

Algorithm 1: Exact Solver with Rank Known

```
Input: Data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, positive integer $r$ and
        positive real value $\lambda$.
Initial.: $t \leftarrow 0$; $\mathbf{U}^{(0)} \in \mathbb{R}^{m\times r} \leftarrow \mathrm{eye}(m, r)$;
        $\mathbf{V}^{(0)} \in \mathbb{R}^{n\times r} \leftarrow \mathbf{0}$; $\mathbf{K}^{(0)} \in \mathbb{R}^{m\times n} \leftarrow \mathbf{0}$;
        $\mathbf{Z}^{(0)} \in \mathbb{R}^{m\times n} \leftarrow \mathbf{0}$; $\mu^{(0)} \leftarrow 1$; $\rho \leftarrow 1.5$;
while not converged do
    Update $\mathbf{U}^{(t+1)}$, $\mathbf{V}^{(t+1)}$, $\mathbf{K}^{(t+1)}$ and $\mathbf{Z}^{(t+1)}$ via Eq.
     (9), (11), (13) and (14);
    $\mu^{(t+1)} \leftarrow \min(\rho\mu^{(t)}, 1e20)$; $t \leftarrow t + 1$;
end
Output: $\mathbf{U}^t$,$\mathbf{V}^t$,$\mathbf{K}^t$
```

Algorithm 2: Heuristic Rank Estimator

```
Input: Right factor $\mathbf{V} \in \mathbb{R}^{n\times d}$, single and batch
        contribution thresholds $\tau_s \geq 0$ and $\tau_b > 0$.
$[\mathbf{c}, \mathbf{o}] \leftarrow \mathrm{sort} \downarrow ([\|\mathbf{V}_1\|_F, \|\mathbf{V}_2\|_F, ..., \|\mathbf{V}_d\|_F])$;
$\mathbf{c} \leftarrow \mathbf{c}/\sum_{i=1}^{d} \|\mathbf{V}_i\|_F$; $c_b \leftarrow 0$; $r_{est} \leftarrow d$;
for $i$ from 1 to $d$ do
    if $c_b > \tau_b$ & $\mathbf{c}(i) < \tau_s$ then
        $\mathbf{V}_{\mathbf{o}(i)} \leftarrow \mathbf{0}$; $r_{est} \leftarrow r_{est} - 1$;
    end
    $c_b \leftarrow c_b + \mathbf{c}(i)$;
end
Output: Truncated $\mathbf{V}$ and estimated $r_{est}$
```

Algorithm 3: Exact Solver with Rank Unknown

```
Input: Data $\mathbf{X} \in \mathbb{R}^{m\times n}$, relatively large $r_{est}$.
while not converged do
    $r \leftarrow r_{est}$;
    Obtain $(\mathbf{U}_r^*, \mathbf{V}_r^*, \mathbf{K}_r^*)$ via Algorithm 1 with $r$;
    Estimate $r_{est}$ via Algorithm 2;
end
Output: $r$, $\mathbf{U}_r^*$,$\mathbf{V}_r^*$,$\mathbf{K}_r^*$
```

where $\{\mu^{(t)}\}$ is a monotonically increasing positive sequence. As can be seen from (10), it is a classic Ridge regression problem. Its the closed-form solution can be easily calculated by:

$$\mathbf{V}^{(t+1)} \leftarrow \mu^{(t)}\mathbf{D}^{(t)T}\mathbf{U}^{(t+1)}/(1+\mu^{(t)}). \tag{11}$$

Updating K. For known **U**, **V** and **Z**, **K** can be computed through optimizing the following problem:

$$\begin{aligned}\mathbf{K}^{(t+1)} = &\underset{\mathbf{K}}{\mathrm{argmin}}\, \lambda\|\mathbf{W}\odot(\mathbf{X}-\mathbf{K})\|_1 + \\ &\frac{\mu^{(t)}}{2}\|\mathbf{K}-(\mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)T}-\frac{\mathbf{Z}^{(t)}}{\mu^{(t)}})\|_F^2.\end{aligned} \tag{12}$$

This sub-problem can be efficiently solved in closed form by the shrinkage operator, the definition of which on scalars is $\mathcal{S}_{\epsilon>0}[x] := \mathrm{sgn}(x)\max(|x|-\epsilon, 0)$. The extension of the shrinkage operator to vectors and matrices is simply applied element-wise. In the sequel, the solution to (12) is:

$$\begin{aligned}\mathbf{K}^{(t+1)} \leftarrow &\mathbf{W}\odot(\mathbf{X}-\mathcal{S}_{\frac{\lambda}{\mu^{(t)}}}[\mathbf{X}-\mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)T}+\frac{\mathbf{Z}^{(t)}}{\mu^{(t)}}]) \\ &+\bar{\mathbf{W}}\odot(\mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)T}-\frac{\mathbf{Z}^{(t)}}{\mu^{(t)}}),\end{aligned} \tag{13}$$

where $\bar{\mathbf{W}}$ is the complementary support of **W**.

Updating Multiplier Z. Besides, there is one multiplier to update, which is simply given by:

$$\mathbf{Z}^{(t+1)} \leftarrow \mathbf{Z}^{(t)} + \mu^{(t)}(\mathbf{K}^{(t+1)} - \mathbf{U}^{(t+1)}\mathbf{V}^{(t+1)T}). \tag{14}$$

For completeness, the procedure of solving (6) is outlined in Algorithm 1. The algorithm should not be terminated until the equality constraint $\mathbf{K}^{(t)} - \mathbf{U}^{(t)}\mathbf{V}^{(t)T}$ is satisfied up to a given tolerance, that is $\|\mathbf{K}^{(t)} - \mathbf{U}^{(t)}\mathbf{V}^{(t)T}\|_F \leq \varsigma\|\mathbf{X}\|_F$, or the maximal number of iterations is reached. In all our experiments, the tolerance factor ς is chosen as $1e{-}10$.

2.2. Solver with Rank Unknown

As previously analyzed, the model (6) can perform reliably in a simpler shape than (4) and (3) when the target rank is known *a priori*. But it would still suffer from degraded performance when the intrinsic rank is unknown. The reason is that, when the guess of rank d is (much) larger than the intrinsic rank r, some errors might survive. So it is expected to have an effective and efficient rank estimator to address this issue.

Prior to detailing our strategy, we here revisit the model shown in (6). Owning to the orthogonality of **U**, its rank is fixed to be r, so the rank of recovery $\mathbf{U}\mathbf{V}^T$ depends on that of **V**. In the sequel, suppressing the upper bound of the rank of **V** is key. From a dictionary learning perspective, **U** acts as an orthogonal dictionary while $\mathbf{V}^T$ a representation matrix. If the response of data to the ith item in **U**, say $\mathbf{V}_i$, is small, then it means the contribution of $\mathbf{U}_i$ to data reconstruction is weak and the noise highly likely hides in the weak component.

Inspired by the above analysis, we design a simple and effective strategy to estimate rank. The Frobenius norms of columns of **V** are first sorted in descending order. With the ordered and normalized value vector **c** and the corresponding index **o**, the procedure performs sequentially. A response vector $\mathbf{V}_{\mathbf{o}(i)}$ is truncated (set to **0**) and the upper-bound of rank is accordingly decreased by 1 if the batch contribution of reserved responses c_b succeeds a pre-defined threshold τ_b and the single contribution of the current response $\mathbf{c}(i)$ is less than another threshold τ_s, otherwise the response vector and the upper-bound of rank remain unchanged. This two-threshold principle ensures that the dominant information is kept while no important information is inexorably neglected. For clarity, we summarize the procedure in Algorithm 2.

Although a weak component contributes less to data reconstruction than a strong one, it absolutely does not mean blindly discarding the weak component is proper. We here give an extreme example to simply show the reason. Suppose we have two components, the response vectors of which are $\mathbf{v}_s$ and $\mathbf{v}_w$, respectively, $\|\mathbf{v}_s\|_2^2$ is larger than $\|\mathbf{v}_w\|_2^2$, and $\mathbf{v}_w = a\mathbf{v}_s$. It is easy to see that the two components can be merged into one by somehow rectifying the directions of orthogonal basis with *e.g.* $\tilde{\mathbf{v}}_s \leftarrow \sqrt{1+a^2}\mathbf{v}_s$ and $\tilde{\mathbf{v}}_w \leftarrow \mathbf{0}$. Without any loss of information, the objective value is unchanged while the upper-bound of rank is suppressed. To avoid manslaughter, one possible solution is iterating rank estimation and Algorithm 1 until converged as shown in Algorithm 3, which is called Exact Solver with Rank Unknown. The algorithm demands only several rounds (in our experiments, within 4 iterations) to converge, but it would be interesting and attractive to eliminate the outer iteration. To this end, we bring Alg. 2 into Alg. 1, *i.e.* immediately truncating $\mathbf{V}$ after each update of $\mathbf{V}$, as an inexact version of Alg. 3 (Inexact Solver with Rank Unknown).

3. CONVERGENCE AND COMPLEXITY ANALYSIS

Convergence Analysis. Note that the estimated rank sequence $\{r_{est}\}$ by Alg. 3 is non-increasing, and the convergence of Alg. 3 depends on that of Alg. 1. For Alg. 1 and 3, the convergence is established by Theorem 2.

Theorem 2. *The proposed Algorithm 1 converges to at least a critical point of the optimization problem* (6).

Proof. Due to space limitation, we put the detailed proof in the supplementary material. □

As regards the Inexact Solver with Rank Unknown, we are not aware of any solid proof on its convergence. In spite of this imperfection, empirical evidence on both synthesized and real data presented in the next section suggests that it has very strong and stable convergence behavior.

Complexity Analysis. Regarding the $\mathbf{U}$ sub-problem, it requires $\mathcal{O}(mnd + md^2)$ for a matrix multiplication and a QR decomposition. As for the $\mathbf{V}$, $\mathbf{E}$ and $\mathbf{Z}$, each of them takes $\mathcal{O}(mnd)$. Thus the total time complexity of Alg. 1 is $\mathcal{O}(t(mnd + md^2))$, where t is the number of (inner) iterations. Typically $d \ll \min(m, n)$, hence we can say that the complexity is $\mathcal{O}(tmnd)$. The dominant operation of Alg. 2 comes from computing and sorting the Frobenius norms of d columns of $\mathbf{V}$, which has the complexity $\mathcal{O}(nd + d\log d)$. Therefore the time complexity of Alg. 3 takes $\mathcal{O}(qtmnd)$ where q is the number of outer iterations (in our experiments, $q \leq 4$) while that of the inexact version only $\mathcal{O}(tmnd)$.

4. RELATED WORK

Here, we briefly review classic and recent LRMA achievements closely related with ours, which are basically derived from the NNM (2) and BF (3) models. PCA [8] follows the NNM with the ℓ^2 loss by assuming the residual existing in the observation satisfies a Gaussian distribution, while Principal Component Pursuit (PCP) [9] takes care of arbitrary outliers by adopting the ℓ^1 penalty. To accelerate PCP, Zhou and Tao developed GoDec [10] by using bilateral random projections (BRP) based approximation. More recently, Oh *et al.* proposed an approximate SVT method that exploits the property of iterative NNM procedures by range propagation and adaptive rank prediction [11]. Since conventional NNM based approaches do not fully utilize a priori target rank information about the problems when the exact rank of clean data is given, PSVT [12] attempts to minimize partial sum of singular values in PCP, which behaves better than PCP when the number of samples is deficient. Cabral *et al.* proposed a method Unifying [4] that unifies nuclear norm and bilinear factorization. To further improve the stability of Unifying when highly corrupted data are presented, factEN [13] employs the Elastic-Net regularization. As a hybrid of NNM and BF, RegL1 [14] solves a similar problem to ours by replacing $\frac{1}{2}\|\mathbf{V}\|_F^2$ with $\|\mathbf{V}\|_*$, which reduces the cost of PCP by computing SVDs on a smaller matrix $\mathbf{V}$ instead of $\mathbf{U}\mathbf{V}^T$. RBF [7] shares the same model with RegL1 with different solving details.

Table 1: Performance comparison of state-of-the-art methods

Matrix		PSVT [12]		Unifying [4]		factEN [13]		RBF [7]		Ours	
Dim.	Rank	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
500	50	**2e−10**	6.02	$9e{-9}$	2.89	$4e{-10}$	1.58	$6.5e{-3}$	4.02	**2e−10**	**0.52**
1000	50	**1e−10**	28.89	$6e{-10}$	13.56	$3e{-10}$	8.65	**1e−10**	11.15	$2e{-10}$	**2.17**
2000	200	**2e−10**	310.81	**2e−10**	135.57	$4e{-10}$	55.95	$22.6e{-3}$	141.79	**2e−10**	**18.79**
5000	300	$1e{-10}$	3389.07	**7e−11**	1154.76	$4e{-10}$	434.71	$1e{-10}$	845.08	$1e{-10}$	**137.48**

5. EXPERIMENTAL VERIFICATION

The parameters of the competitors adopt those suggested by the authors. As for our algorithms, we empirically set $\lambda = \sqrt{n}$ and, $\tau_b = 0.7$ and $\tau_s = 0.01$ throughout this section.

5.1. Synthetic Data

A – Data Preparation and Quantitative Metrics. Similar with [9, 4], we generate a rank-r matrix $\mathbf{L}_0$ as a product $\mathbf{L}_0 = \mathbf{U}_0\mathbf{V}_0^T$ where $\mathbf{U}_0$ and $\mathbf{V}_0$ are $m \times r$ and $n \times r$ matrices with entries independently sampled from a $\mathcal{N}(0,1)$ distribution. Then we corrupt the entries by replacing a fraction δ_s of $\mathbf{L}_0$ with large errors drawn from a uniform distribution over $[-50, 50]$ at random. To quantitatively reveal the recovery performance, we employ error (defined as $\|\mathbf{L}_0 - \hat{\mathbf{L}}\|_F / \|\mathbf{L}_0\|_F$) and elapsed time as our metrics.

B – Performance Evaluation on Recovery with Rank Known. We compare our method with state-of-the-art alternatives including PSVT , Unifying, factEN and RBF.[2] Table 1 reports the results on four cases in terms of error and time, where the outlier ratio is fixed to 0.2. From the table, we see

[2] The code of RBF is unavailable when this paper is prepared, so we implement it by strictly following the pseudo-code given in [7]. The codes of all other competitors are from the authors. All the codes are in Matlab.

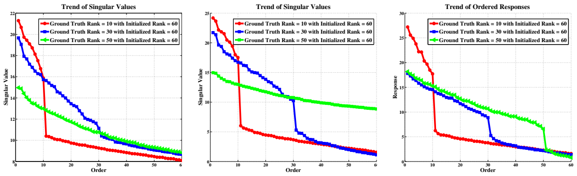

Fig. 1: **Left:** singular values of original data. **Middle** and **Right:** singular values and our ordered responses of output of Algorithm 1, respectively.

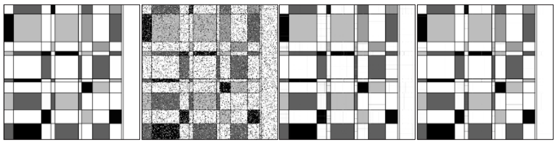

Fig. 2: From **Left** to **Right**: Ground truth, dirty input, results by our inexact and exact algorithms, respectively.

that the gain of our method in time becomes conspicuous as the data scale increases. The recovery errors of PSVT, Unifying, factEN and our method are close and low, while RBF performs poorly when the ratio r/n is relatively large.

C – Efficacy of Heuristic Rank Estimator. One may have a question: *Can singular values (SVs) be applied to estimate rank?* Our answer is: *Yes but inferior to our strategy.* To verify this, we generate three square matrices of dimension $n = 400$ with different ground truth (GT) ranks $(10, 30, 50)$, fixed outlier ratio 0.2 and initialized rank 60. The first picture in Fig. 1 shows that, by using singular values of the original matrices, only the rank-10 case has a clear division at GT rank. This implies that directly estimating the target rank on the SVs of original dirty data is not so reliable. The middle and right plots respectively display the results of the SV based and our strategies after executing Alg. 1 on the original data, from which we can see that the SV scheme recalls the boundary of the rank-30 case but still fails for the rank-50, while our strategy provides clear divisions for all three cases. In addition to the effectiveness, the SV strategy is less efficient than ours especially applied to the inexact procedure, which again introduces SVD operations that our work attempts to avoid.

D – Performance Evaluation on Recovery with Rank Unknown. For better illustration, we employ a 256×256 image with GT rank 9. Table 2 contains the results obtained by PCP-IALM, Unifying, factEN, RBF, our exact and inexact solvers with rank unknown for the case of outlier ratio 0.25 and initialized rank 100. From the numbers, we can see that our exact solver achieves the best accuracy and the second lowest time cost, while the inexact one reaches the second best accuracy and the lowest time. PCP-IALM, Unifying and RBF perform reasonably well in accuracy, as these methods in nature are NNM based, which are superior to factEN in this scenario. Our exact algorithm takes 3 outer iterations, the (estimated) input ranks for the iterations are 100, 27 and 9, respectively. That is why it spends less than 3 times time that the inexact one spends. The rank of $\hat{\mathbf{L}}$ obtained by the inexact solver is also 9. Please see Fig. 2 for visual results.

Table 2: Performance comparison. Error has a factor 10^{-2}.

PCP-IALM		Unifying		factEN		RBF		Our Inexact		Our Exact	
Error	Time	Error	Time	Error	Time	Error	Time	Error	Time	Error	Time
4.47	0.786	4.33	2.168	28.65	0.588	4.32	2.19	*3.34*	**0.187**	**1.98**	*0.399*

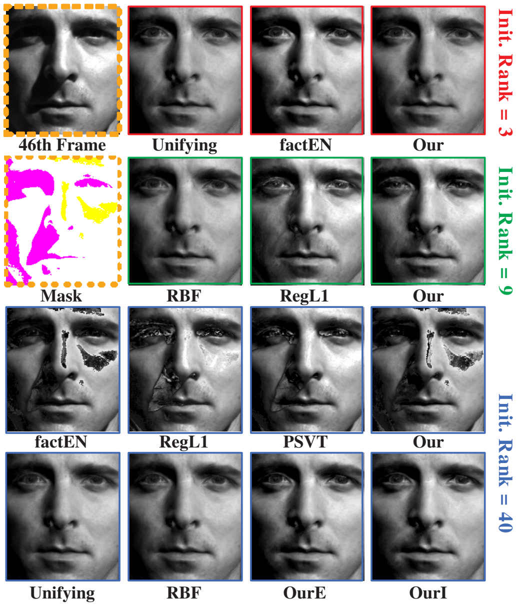

Fig. 3: The 1st and 2nd rows correspond to the cases with rank initialized to 3 and 9, respectively, while the rest two rows to rank initialized to 40.

5.2. Real Data

A – Photometric Stereo. Images of a static Lambertian object sensed by a fixed camera under a varying but distant point lighting source lie in a rank-3 subspace. While extending it to general conditions, first and second order spherical harmonics corresponding to rank-4 and rank-9 factorization are able to capture at least 75% and 98% of the reflectance, respectively. We assess the performance of LRMA techniques on the cropped Extended YaleB-10 sequence. As in [14, 4], we treat as missing all pixels with intensity greater than 235, for 8-bit depth images, or lower than 20. A sample frame and its corresponding mask are displayed on the top-left of Fig. 3. We can see that all the competitors give promising results with initialized rank 3 and 9. In time, Unifying takes (rank-3: 41.66s, rank-9: 41.68s), while RegL1 (23.04s, 51.52s), RBF (27.11s, 27.03s), PSVT (14.65s, 14.22s) and factEN (10.41s, 13.54s), respectively to accomplish the task. Our method needs noticeably less computational cost than the others, *i.e.* (3.98s, 4.57s). Next, we test the ability of the competitors with a relatively large initial of rank, say 40. As shown in Fig. 3, the results by Unifying and RBF reflect their stability thanks to the connection to NNM, while the others including factEN, RegL1, PSVT and our solver with initialized rank-40 fail to

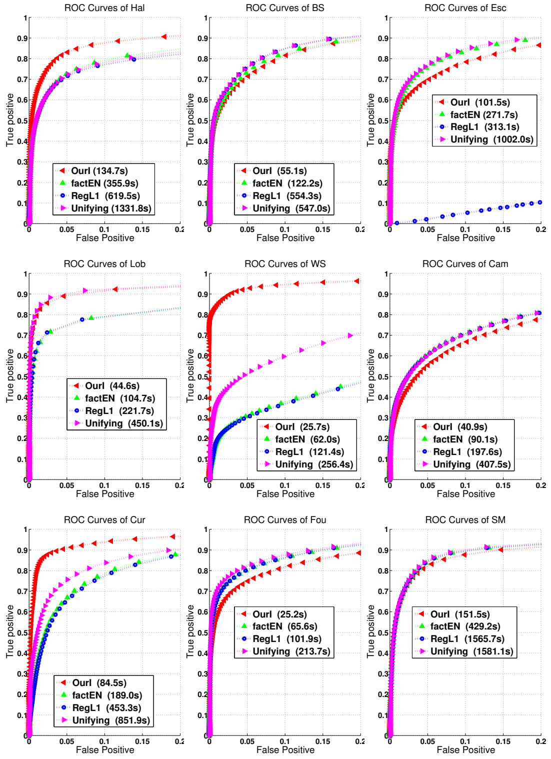

Fig. 4: Quantitative results on the Star dataset.

generate good results. Our solvers for rank unknown (exact and inexact versions denoted as OurE and OurI, respectively), can improve the performance and give pleasing results. Both OurE and OurI converge to rank 4, which indicates the key information can be contained by a 4-dimensional subspace.

B – Foreground Detection. The aim of this task is to separate foregrounds from surveillance videos of the Star dataset, which consists of 9 real world videos of varying scenarios. Since the background of surveillance videos typically lies in a relatively low-dimensional subspace, we empirically set the initial rank to 5 for all the 9 sequences. In this experiment, we employ our inexact solver with rank unknown to compare with Unifying, factEN and RegL1. Figure 4 summarizes the ROC plots. From the curves, it can be viewed that our approach significantly outperforms the others on the Hal, WS, Lob and Cur sequences, and competes very favorably with the others on the cases of BS and SM, but is slightly inferior to the others on Esc, Cam and Fou, in terms of accuracy. Please notice that Unifying performs well on these real videos but at very high time cost. Our method shows its superiority with regard to time over all of Unifying, factEN and RegL1. The time costs of the competitors for each sequence can be found in the legends of Fig. 4, please.

6. CONCLUSION

This paper has shown a simple factorization method for solving the LRMA problem, which imposes the orthogonality pursuit on one factor and ℓ^2 regularization on the other to shrink the solution space and thus accelerate the optimization procedure with sufficient accuracy. The theoretical analysis on convergence and complexity of the proposed algorithms, and the experimental results compared to the state-of-the-arts, have demonstrated their advantages. It is positive that our framework is ready to embrace various domain knowledge for further boosting the performance on different specific tasks.

7. REFERENCES

[1] X. Guo, X. Wang, L. Yang, X. Cao, and Y. Ma, "Robust foreground detection using smoothness and arbitrariness constraints," in *ECCV*, 2014, pp. 535–550.

[2] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with applications to image denoising," in *CVPR*, 2014, pp. 2862–2869.

[3] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *Journal of Machine Learning Research*, vol. 99, pp. 2287–2322, 2010.

[4] R. Cabral, F. De la Torre, J. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *ICCV*, 2013, pp. 2488–2495.

[5] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low rank representation," in *NIPS*, 2011, pp. 695–704.

[6] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, pp. 333–361, 2012.

[7] F. Shang, Y. Liu, H. Tong, J. Cheng, and H. Cheng, "Robust bilinear factorization with missing and grossly corrupted observations," *Information Sciences*, vol. 307, pp. 53–72, 2015.

[8] K. Pearson, "On lines and planes of closet fit to systems of points in space," *Philosophical Magazine*, 1901.

[9] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[10] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *ICML*, 2011, pp. 33–40.

[11] T. Oh, Y. Matsushita, Y. Tai, and I. Kweon, "Fast randomized singular value thresholding for nuclear norm minimization," in *CVPR*, 2015, pp. 4484–4493.

[12] T. Oh, Y. Tai, J. Bazin, H. Kim, and I. Kweon, "Partial sum minimization of singular values in robust pca: Algortihm and applications," *IEEE TPAMI*, 2015.

[13] E. Kim, M. Lee, and S. Oh, "Elastic-net regularization of singular values for robust subspace learning," in *CVPR*, 2015, pp. 915–923.

[14] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust l1-norm," in *CVPR*, 2012, pp. 1410–1417.