

Robust Matrix Factorization by Majorization Minimization

Zhouchen Lin¹, Senior Member, IEEE, Chen Xu, and Hongbin Zha¹, Member, IEEE

Abstract— L_1 -norm based low rank matrix factorization in the presence of missing data and outliers remains a hot topic in computer vision. Due to non-convexity and non-smoothness, all the existing methods either lack scalability or robustness, or have no theoretical guarantee on convergence. In this paper, we apply the Majorization Minimization technique to solve this problem. At each iteration, we upper bound the original function with a strongly convex surrogate. By minimizing the surrogate and updating the iterates accordingly, the objective function has sufficient decrease, which is stronger than just being non-increasing that other methods could offer. As a consequence, without extra assumptions, we prove that any limit point of the iterates is a stationary point of the objective function. In comparison, other methods either do not have such a convergence guarantee or require extra critical assumptions. Extensive experiments on both synthetic and real data sets testify to the effectiveness of our algorithm. The speed of our method is also highly competitive.

Index Terms—Matrix factorization, majorization minimization, alternating direction method of multipliers (ADMM)

1 INTRODUCTION

LOW rank matrix factorization has been successfully applied to a variety of computer vision tasks, such as rigid [1] and nonrigid [2] Structure from Motion (SfM), photometric stereo [3], [4], layer extraction [5], face recognition [6] and image recovery [7]. Quite often, some entries of the observed data matrix are missing. This problem can be modelled as follows:

$$\min_{U,V} \|W \odot (M - UV^T)\|, \quad (1)$$

where $M \in \mathbb{R}^{m \times n}$ is the measurement matrix with rank r known apriori ($r \ll \min(m, n)$) and $\|\cdot\|$ is some matrix norm. $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{n \times r}$ are the unknown low rank factorization. W is a 0-1 binary mask with the same size as M . The entry value of W being 0 means that the component at the same position in M is missing, and 1 otherwise. The operator \odot is the Hadamard element-wise product.

Alternative to bilinear factorization as (1), the nuclear norm, or sum of singular value, has been proposed and shown to be effective [8], [9], [10]. Under certain assumptions [11], the low rank product matrix UV^T can be exactly recovered. However, such assumptions may not hold in practice. In addition, the rank of target matrix in some computer vision problems, e.g., rigid and nonrigid SfM and image recovery, is known or could be estimated apriori,

while the nuclear norm cannot easily incorporate this prior information. So in this paper, we only focus on the bilinear factorization as (1).

When there are no missing data and the norm is the L_2 -norm, namely Frobenius norm, SVD gives the optimal solution to (1). To tackle the case of missing data, many optimization methods [12], [13], [14], [15], [16] have been proposed. However, the L_2 -norm is only optimal to Gaussian noise and is fragile to outliers. For robustness, [17] proposed to adopt the L_1 -norm, the sum of absolute values of entries, in (1). Although having been extensively investigated in the following years, it remains a hot yet challenging topic [18], [19], [20], [21], [22]. Due to its non-convexity and non-smoothness, all the existing methods either lack scalability or robustness, or have no theoretical guarantee on convergence.

In this paper, we propose a new algorithm for robust matrix factorization (RMF), i.e., problem (1) with the L_1 -norm. Our method is based on the Majorization Minimization (MM) technique and can overcome all the weakness of the existing algorithms. At each iteration, we upper bound the original objective function with a strongly convex surrogate, which is a function of the increments in U and V . By minimizing the surrogate, U and V can be updated accordingly. The contributions of this paper are as follows:

- (a) We propose to use MM to solve the L_1 -norm based bilinear matrix factorization. By constructing a strongly convex surrogate at each iteration, the corresponding convex subproblem can be efficiently solved, despite its non-smoothness. Hence our method could handle relatively large scale problems.
- (b) Although in general MM, as well as most existing methods for RMF, can only ensure the non-increment of the original objective function and hence has no convergence guarantee, we prove that our MM for RMF (RMF-MM for short) has sufficient

• The authors are with the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China and the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, China.
E-mail: {zlin, xuen}@pku.edu.cn, zha@cis.pku.edu.cn.

Manuscript received 9 Apr. 2015; revised 20 Apr. 2016; accepted 3 Jan. 2017.
Date of publication 10 Jan. 2017; date of current version 12 Dec. 2017.

Recommended for acceptance by F. de la Torre.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2017.2651816

decrease in the objective function value. As a consequence, we can further prove that any limit point of RMF-MM is a stationary point of (1). To the best of our knowledge, this is the first work for RMF with convergence guarantee and without extra critical assumptions.

- (c) We conduct extensive experiments to demonstrate the effectiveness of RMF-MM by comparing it with the state-of-the-art methods. In particular, when the measurement matrix has a high missing rate or/and is severely corrupted, RMF-MM can still attain plausible results while all others fail. Its speed is also highly competitive.

2 RELATED WORK

In this section, we review some representative algorithms for the problem of bilinear matrix factorization with missing data.

When the norm is the L_2 -norm, Buchanan et al. [12] gave a comprehensive survey on many factorization-based methods. They also proposed the Damped Newton algorithm to optimize for U and V jointly. Okatani et al. [13] showed that a Wiberg marginalization strategy on U or V is insensitive to initialization. Mitra et al. [14] formulated the problem as a low rank semidefinite program (LRSDP) and showed its ability to handle large scale data and incorporate additional constraints. Bue et al. [16] utilized Augmented Lagrangian Multiplier (ALM) and projected variables onto the constraint manifold at each step. They presented appealing visual results for some popular factorization problems in computer vision. Wen et al. [15] constructed a nonlinear successive over-relaxation (SOR) algorithm and showed that its speed is several times faster than many other methods.

Unfortunately, all of the above L_2 -norm based methods are only optimal to Gaussian noise and sensitive to outliers. An intuitive extension of L_2 -norm to handle outliers is to assign a proper weight to each element by iterative re-weighted least squares [23]. However, a good initialization is very critical for success. For robustness, Ke and Kanade [17] suggested replacing the L_2 -norm with the L_1 -norm. From then on, many algorithms have been proposed to tackle this problem. They mainly fall into two categories: Alternate Minimization (AM) algorithms and Alternating Direction Method of Multipliers (ADMM).

Alternate Minimization algorithms update U and V alternately, by fixing the other matrix. When one of the matrix is fixed, the corresponding subproblem becomes convex and the subproblem is of smaller size, which makes this kind of methods very attractive. For example, Ke and Kanade [17] converted the subproblem to convex Linear Programs (LP) (or Quadratic Programs (QP) for Huber loss). However, Eriksson and Hengel [18] pointed out that such an updating scheme tends to flatline, i.e., converge slowly after initial rapid update. Actually, for non-smooth problems, alternate minimization can easily get stuck at non-stationary points [24]. So Eriksson and Hengel represented V implicitly with U and extended the Wiberg Algorithm to L_1 -norm (named as L_1 -Wib) [21]. They only proved the convergence of the objective function value, not the sequence $\{(U_k, V_k)\}$ itself. Moreover, they had to assume that the dependence of V on

U is differentiable, which is unlikely to hold everywhere. Additionally, as L_1 -Wib unfolds matrix U into a vector its memory requirement is very high, which prevents it from large scale computation. To overcome this issue, Meng et al. [19] changed to update only one entry of U and V in each iteration. By adopting the cyclic weighted median (CWM) method, each sub-problem can be solved efficiently. Although shown to be more scalable, this coordinate decent algorithm can only converge to coordinate-wise minimum points [19], which are cusps of the iso-surfaces of the objective function, not stationary points. Kim et al. [25] also proposed two versions of algorithms (namely ARG-A and ARG-D) by finding rectified gradients of U and V cyclically. They were shown to be less computationally expensive than computing the exact updating direction. However, the intrinsic alternate minimization scheme restricted their convergence guarantee. And they could only prove the convergence of one version of the algorithms, i.e., ARG-D, to subspace-wise local minimum [25], which is similar to that of CWM [19].

Recently, ADMM has also been applied to tackle RMF. By introducing an auxiliary variable E and a constraint $E = UV^T$, the original unconstrained problem is reformulated as a constrained one

$$\min_{E,U,V} \|W \odot (M - E)\|, \quad s.t. \quad E = UV^T. \quad (2)$$

Then by ADMM each sub-problem can be solved in closed form. As there are more than two blocks of variables (E, U , and V), the original convergence theory of ADMM may not be applied to this non-convex problem directly. By assuming that the variables are bounded and convergent, Shen et al. [20] proved that any accumulation point of their LMA-Fit algorithm is the Karush-Kuhn-Tucker (KKT) point. It was shown that the method was only able to handle outliers with magnitudes comparable to the low rank matrix [20]. Moreover, the penalty parameter was fixed, which was not easy to tune for fast convergence. Zheng et al. [21] thus modified with an adaptive penalty. To reduce the solution space, they added a nuclear norm regularization to V and enforced U to be column orthogonal (named as $\text{Reg}L_1$). Cabral et al. [22] replaced the nuclear norm regularization with a sum of squared Frobenius norms on both U and V (named as UNuBi). Both algorithms, $\text{Reg}L_1$ [21] and UNuBi [22], scaled well and achieved plausible performance in practice. However, as the experiments in Section 6 show, they are still not robust when the observed matrix is ill-conditioned. Since the convergence analysis in [20] cannot be directly extended to the case with additional regularization and adaptive penalty, it remains unknown whether the iterations of $\text{Reg}L_1$ and UNuBi converge to KKT points.

3 PROBLEM FORMULATION

3.1 Optimization with MM

For completeness, we first give a brief introduction to optimization with MM, which will be used later for RMF.

When an objective function is not easy to optimize directly, e.g., it is non-convex or/and non-smooth, iteratively minimizing a majorizing surrogate function is preferred. Such a method is called Majorization Minimization [26], [27], [28]. The key idea of MM is to construct an easy-

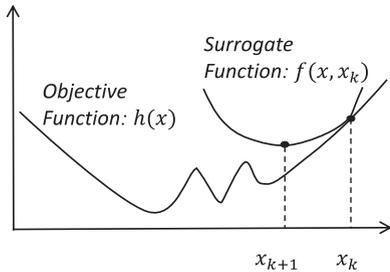


Fig. 1. Illustration of MM. The curve of the surrogate function $f(x, x_k)$ is above that of the objective function $h(x)$ and equal to $h(x)$ at x_k . By minimizing the surrogate function $f(x, x_k)$, we get the next iterate x_{k+1} . The values of objective function $h(x)$ at $\{x_k\}$ are non-increasing.

to-tackle surrogate function $f(x, x_k)$ based on the current iterate x_k . Denote the objective function as $h(x)$. Then the surrogate function should satisfy

$$h(x) \leq f(x, x_k), \quad \forall x, \quad \text{and} \quad (3)$$

$$x_k = \arg \min_x f(x, x_k) - h(x). \quad (4)$$

Typically, we can further choose $f(x, x_k)$ such that $f(x_k, x_k) = h(x_k)$. By minimizing $f(x, x_k)$, we get the next iterate as

$$x_{k+1} = \arg \min_x f(x, x_k). \quad (5)$$

Then

$$h(x_{k+1}) \leq f(x_{k+1}, x_k) \leq f(x_k, x_k) = h(x_k). \quad (6)$$

Namely, the objective function value is non-increasing. Fig. 1 gives an illustration. Note that in general MM does not have any guarantee, e.g., on the convergence of the iteration, beyond the non-increment of the objective function value.

A variety of algorithms can be regarded as MM, such as proximal methods [29], [30], [31], DC programming [32], boosting [33], [34] and some variational Bayes techniques [35], [36]. The concept of surrogate has also been introduced in the literature of sparse optimization [37], [38] and matrix factorization [39], [40].

3.2 New Surrogate Function

Choosing L_1 -norm, we can explicitly write (1) as follows:

$$\min_{U, V} \|W \odot (M - UV^T)\|_1. \quad (7)$$

Note that there is a gauge freedom: for any invertible $r \times r$ matrix G , $\tilde{U}\tilde{V}^T = (UG)(VG^{-1})^T = UV^T$. Therefore, for any minimizer (U^*, V^*) , there are a family of infinitely many equivalent solutions. In addition, not all mask matrices admit a unique (even up to the gauge) solution [12]. To reduce the degrees of freedom, we target on a modified version of (7) as [22]:

$$\begin{aligned} \min_{U, V} H(U, V) &= \min_{U, V} \|W \odot (M - UV^T)\|_1 \\ &+ \frac{\lambda}{2} \|U\|_F^2 + \frac{\lambda}{2} \|V\|_F^2, \end{aligned} \quad (8)$$

where $\lambda \in \mathbb{R}^+$ is a regularization parameter. In addition, for any matrix X with rank less than or equal to r , there exists an identity [22]:

$$\|X\|_* = \min_{X=UV^T} \frac{1}{2} \|U\|_F^2 + \frac{1}{2} \|V\|_F^2, \quad (9)$$

where $\|\cdot\|_*$ denotes the nuclear norm. Thus model (8) can also be regarded as a nuclear norm regularized one to some extent.

Suppose that we have obtained (U_k, V_k) at the k th iteration. We split (U, V) as the sum of (U_k, V_k) and the unknown increment $(\Delta U, \Delta V)$

$$(U, V) = (U_k, V_k) + (\Delta U, \Delta V). \quad (10)$$

Then (8) can be equivalently written as

$$\begin{aligned} \min_{\Delta U, \Delta V} H_k(\Delta U, \Delta V), \quad \text{where} \\ H_k(\Delta U, \Delta V) &= \|W \odot (M - (U_k + \Delta U)(V_k^T + \Delta V)^T)\|_1 \\ &+ \frac{\lambda}{2} \|U_k + \Delta U\|_F^2 + \frac{\lambda}{2} \|V_k + \Delta V\|_F^2. \end{aligned} \quad (11)$$

Now the key step is to find an increment $(\Delta U, \Delta V)$ such that the objective function keeps decreasing. However, problem (11) is not easier than the original problem (8). Inspired by MM, we try to relax (11) to a convex surrogate.

By the triangular inequality of norms, we have the following inequality:

$$\begin{aligned} H_k(\Delta U, \Delta V) &\leq \|W \odot (M - U_k V_k^T - \Delta U V_k^T - U_k \Delta V^T)\|_1 \\ &+ \|W \odot (\Delta U \Delta V^T)\|_1 + \frac{\lambda}{2} \|U_k + \Delta U\|_F^2 + \frac{\lambda}{2} \|V_k \\ &+ \Delta V\|_F^2. \end{aligned} \quad (12)$$

Denoting Δu_i^T and Δv_i^T as the i th rows of ΔU and ΔV , respectively, we can further relax

$$\begin{aligned} \|W \odot (\Delta U \Delta V^T)\|_1 &= \left\| W \odot \begin{pmatrix} \Delta u_1^T \Delta v_1 & & \Delta u_1^T \Delta v_n \\ \vdots & \ddots & \vdots \\ \Delta u_m^T \Delta v_1 & & \Delta u_m^T \Delta v_n \end{pmatrix} \right\|_1 \\ &\leq \frac{1}{2} \|\Lambda_u \Delta U\|_F^2 + \frac{1}{2} \|\Lambda_v \Delta V\|_F^2, \end{aligned} \quad (13)$$

where the inequality is derived from the Cauchy-Schwartz inequality. Λ_u and Λ_v are diagonal matrices. The i th diagonal entry of Λ_u is chosen to be $\sqrt{\#W_{(i, \cdot)}} + \epsilon$, where $\#W_{(i, \cdot)}$ represents the number of non-zero entries in the i th row of W and $\epsilon > 0$ is any positive scalar. Similarly, the j th diagonal entry of Λ_v is chosen to be $\sqrt{\#W_{(\cdot, j)}} + \epsilon$, where $\#W_{(\cdot, j)}$ represents the number of non-zero entries in the j th column of W . The equality holds if and only if $(\Delta U, \Delta V) = (\mathbf{0}, \mathbf{0})$.

For simplicity, we define $J_k(\Delta U, \Delta V)$ as follows:

$$\begin{aligned} J_k(\Delta U, \Delta V) &= \|W \odot (M - U_k V_k^T - \Delta U V_k^T - U_k \Delta V^T)\|_1 \\ &+ \frac{\lambda}{2} \|U_k + \Delta U\|_F^2 + \frac{\lambda}{2} \|V_k + \Delta V\|_F^2. \end{aligned} \quad (14)$$

Then we have a relaxed function of $H_k(\Delta U, \Delta V)$

$$F_k(\Delta U, \Delta V) = J_k(\Delta U, \Delta V) + \frac{1}{2} \|\Lambda_u \Delta U\|_F^2 + \frac{1}{2} \|\Lambda_v \Delta V\|_F^2. \quad (15)$$

Combining (12), (13) and (14), it is obvious that $F_k(\Delta U, \Delta V)$ satisfies conditions (3) and (4). Thus it can be a surrogate of $H_k(\Delta U, \Delta V)$. Moreover, $F_k(\Delta U, \Delta V)$ is strongly convex with a unique optimal solution, denoted as $(\Delta U_k, \Delta V_k)$. By updating (U_k, V_k) with this optimal increment according to (10), the original function $H(U, V)$ will be non-increasing. The iteration stops when the improvement on the objective function $H(U, V)$ in (8) is below a threshold.

Note that the traditional MM only ensures non-increment of the objective function, hence has no convergence guarantee. By contrast, for RMF we can prove that by adopting this new surrogate function, the objective function will have sufficient decrease. Consequently, we can prove that any limit point is a stationary point. More details can be found in Section 5. Also note that Mairal [28] gave comprehensive analysis on the convergence of MM using the first-order surrogate. However, he had to assume that the difference between the objective function and the surrogate is differentiable. As $F_k(\Delta U, \Delta V) - H_k(\Delta U, \Delta V)$ is *not* differentiable, his result does not apply to our problem.

4 MINIMIZING THE SURROGATE FUNCTION

Now we show how to find the minimizer of the convex $F_k(\Delta U, \Delta V)$ in Eq.(15). This can be easily done by using the Linearized Alternating Direction Method with Parallel Splitting and Adaptive Penalty (LADMPSAP) method [41], whose computation cost and memory requirement are relatively low.

4.1 Sketch of LADMPSAP

LADMPSAP fits for solving the following linearly constrained separable convex programs

$$\min_{\mathbf{x}_1, \dots, \mathbf{x}_n} \sum_{j=1}^n f_j(\mathbf{x}_j), \quad \text{s.t.} \quad \sum_{j=1}^n \mathcal{A}_j(\mathbf{x}_j) = \mathbf{b}, \quad (16)$$

where \mathbf{x}_j and \mathbf{b} could be either vectors or matrices, f_j is a proper convex function and \mathcal{A}_j is a linear mapping. Very often, there are multiple blocks of variables ($n \geq 3$).

We denote the iteration index by superscript i . The LADMPSAP algorithm consists of the following steps [41]:

- (a) Update \mathbf{x}_j 's ($j = 1, \dots, n$) in parallel:

$$\mathbf{x}_j^{i+1} = \underset{\mathbf{x}_j}{\operatorname{argmin}} f_j(\mathbf{x}_j) + \frac{\sigma_j^{(i)}}{2} \left\| \mathbf{x}_j - \mathbf{x}_j^i + \mathcal{A}_j^\dagger(\hat{\mathbf{y}}^i) / \sigma_j^{(i)} \right\|^2. \quad (17)$$

- (b) Update \mathbf{y} :

$$\mathbf{y}^{i+1} = \mathbf{y}^i + \beta_i \left(\sum_{j=1}^n \mathcal{A}_j(\mathbf{x}_j^{i+1}) - \mathbf{b} \right). \quad (18)$$

- (c) Update β :

$$\beta^{(i+1)} = \min(\beta^{\max}, \rho \beta^{(i)}), \quad (19)$$

where \mathbf{y} is the Lagrange multiplier, $\beta^{(i)}$ is the penalty parameter, $\beta^{\max} \gg 1$ is an upper bound of $\beta^{(i)}$, $\sigma_j^{(i)} = \eta_j \beta^{(i)}$ with $\eta_j > n \|\mathcal{A}_j\|^2$ ($\|\mathcal{A}_j\|$ is the operator norm of \mathcal{A}_j), \mathcal{A}_j^\dagger is the adjoint operator of \mathcal{A}_j ,

$$\hat{\mathbf{y}}^i = \mathbf{y}^i + \beta^{(i)} \left(\sum_{j=1}^n \mathcal{A}_j(\mathbf{x}_j^i) - \mathbf{b} \right), \quad (20)$$

and

$$\rho = \begin{cases} \rho_0, & \text{if } \beta^{(i)} \max \left(\left\{ \sqrt{\eta_j} \left\| \mathbf{x}_j^{i+1} - \mathbf{x}_j^i \right\| \right\} \right) / \|\mathbf{b}\| < \varepsilon_1, \\ 1, & \text{otherwise,} \end{cases} \quad (21)$$

with $\rho_0 \geq 1$ being a constant and $0 < \varepsilon_1 \ll 1$ being a threshold. For more details of LADMPSAP, please refer to [41].

4.2 Optimization Using LADMPSAP

To apply LADMPSAP, we first introduce an auxiliary matrix E such that $E = M - U_k V_k^T - \Delta U V_k^T - U_k \Delta V^T$. Then minimizing $F_k(\Delta U, \Delta V)$ in Eq. (15) can be transformed into the following linearly constrained separable convex program

$$\begin{aligned} \min_{E, \Delta U, \Delta V} & \|W \odot E\|_1 + \left(\frac{\lambda}{2} \|U_k + \Delta U\|_F^2 + \frac{1}{2} \|\Lambda_u \Delta U\|_F^2 \right) \\ & + \left(\frac{\lambda}{2} \|V_k + \Delta V\|_F^2 + \frac{1}{2} \|\Lambda_v \Delta V\|_F^2 \right), \end{aligned} \quad (22)$$

$$\text{s.t. } E + \Delta U V_k^T + U_k \Delta V^T = M - U_k V_k^T,$$

which naturally fits into the model problem (16). By injecting the corresponding variables to the above steps of LADMPSAP, it is easy to update them accordingly. The details can be found in the supplementary material, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TPAMI.2017.2651816>. The stopping criteria of LADMPSAP are derived from the KKT condition [41], [42], [43]. We terminate the iteration when the following two conditions are met

$$\begin{aligned} \beta^{(i)} \max(\sqrt{\eta_e} \|E^{i+1} - E^i\|_F, \sqrt{\eta_u} \|\Delta U^{i+1} - \Delta U^i\|_F, \\ \sqrt{\eta_v} \|\Delta V^{i+1} - \Delta V^i\|_F) / \|M - U_k V_k^T\|_F < \varepsilon_1, \end{aligned} \quad (23)$$

$$\|E^{i+1} - \Delta U^{i+1} V_k^T - U_k \Delta V^{(i+1)T}\|_F / \|M - U_k V_k^T\|_F < \varepsilon_2. \quad (24)$$

Once we obtain the optimal $(\Delta U_k, \Delta V_k)$, we update (U_k, V_k) in the main iteration according to (10). When first executing LADMPSAP, we initialize $E^0 = M - U_0 V_0^T$, $\Delta U^0 = \mathbf{0}$ and $\Delta V^0 = \mathbf{0}$. In the subsequent iterations, we adopt the warm start strategy. Namely, we initialize E^0 , ΔU^0 and ΔV^0 with their respective optimal values in last main iteration.

Denote K and I as the total number of main and inner iterations, respectively, then the overall computation cost of minimizing $H(U, V)$ is $\mathcal{O}(KImnr)$. We also adopt the continuation technique [44] for setting Λ_u and Λ_v in $F_k(\Delta U, \Delta V)$ (see Eq. (15)), which controls the step sizes of updating U and V . Namely, we initialize the diagonal entries of Λ_u and Λ_v with relatively small values and then increase them gradually along with the main iteration until they touch the upper bounds in (13). Such a simple trick can greatly cut the number K of main iterations. As shown in Fig. 2, our RMF-MM converges quickly after only a few iterations.

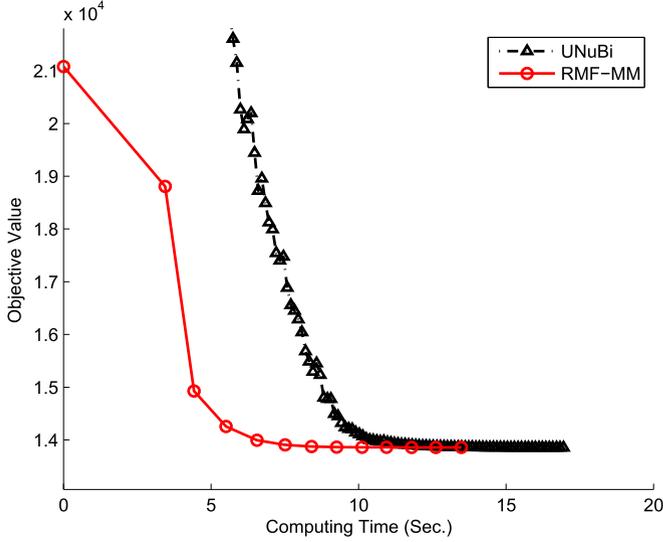


Fig. 2. Objective value (8) versus computing time of UNuBi [22] and our RMF-MM on large scale synthetic data, where each marker represents one iteration. The left part of the curve by UNuBi is not shown as the values are greater than 2.1×10^4 .

5 CONVERGENCE ANALYSIS

In this section, we analyze the convergence of our algorithm. We will show that the objective function has sufficient decrease, and accordingly any limit point of the iterates is a stationary point of problem (8). We first recall the definitions of directional derivative, stationary point and sufficient decrease [45], [46].

Definition 1. Let $f: \mathcal{D} \rightarrow \mathbb{R}$ be a function, where $\mathcal{D} \in \mathbb{R}^m$ is an open set. The directional derivative of f at point x in the feasible direction d is defined as

$$\nabla f(x; d) = \liminf_{\theta \downarrow 0} \frac{f(x + \theta d) - f(x)}{\theta}. \quad (25)$$

Definition 2. A point x is a (minimizing) stationary point of f if $\nabla f(x; d) \geq 0$ for all d such that $x + d \in \mathcal{D}$.

Definition 3. A function f is said to have sufficient decrease on the sequence $\{x_k\}$ if there exists a constant $\alpha > 0$ such that

$$f(x_k) - f(x_{k+1}) \geq \alpha \|x_k - x_{k+1}\|^2, \quad \forall k. \quad (26)$$

Next, we have the following proposition.

Proposition 1. $\nabla F_k(\mathbf{0}, \mathbf{0}; D_u, D_v) = \nabla H_k(\mathbf{0}, \mathbf{0}; D_u, D_v)$ holds for any feasible direction (D_u, D_v) .

Proof. Consider minimizing $F_k(\Delta U, \Delta V) - H_k(\Delta U, \Delta V)$ w.r.t. $(\Delta U, \Delta V)$. It reaches the minimum 0 at $(\Delta U, \Delta V) = (\mathbf{0}, \mathbf{0})$. We then have

$$\nabla F_k(\mathbf{0}, \mathbf{0}; D_u, D_v) \geq \nabla H_k(\mathbf{0}, \mathbf{0}; D_u, D_v), \quad (27)$$

where (D_u, D_v) is any feasible direction.

Define $R_k(\Delta U, \Delta V)$ as

$$R_k(\Delta U, \Delta V) = J_k(\Delta U, \Delta V) - \left(\frac{1}{2} \|\Lambda_u \Delta U\|_F^2 + \frac{1}{2} \|\Lambda_v \Delta V\|_F^2 \right). \quad (28)$$

Similar to $F_k(\Delta U, \Delta V)$, we have

$$H_k(\Delta U, \Delta V) \geq R_k(\Delta U, \Delta V), \quad \forall (\Delta U, \Delta V). \quad (29)$$

The equality holds if and only if $(\Delta U, \Delta V) = (\mathbf{0}, \mathbf{0})$. In other words, the function $H_k(\Delta U, \Delta V) - R_k(\Delta U, \Delta V)$ reaches the minimum 0 at $(\Delta U, \Delta V) = (\mathbf{0}, \mathbf{0})$. We then have

$$\nabla H_k(\mathbf{0}, \mathbf{0}; D_u, D_v) \geq \nabla R_k(\mathbf{0}, \mathbf{0}; D_u, D_v), \quad \forall (D_u, D_v). \quad (30)$$

Compare $F_k(\Delta U, \Delta V)$ with $R_k(\Delta U, \Delta V)$. They are combined with two parts, the common part $J_k(\Delta U, \Delta V)$ and the continuously differentiable part $\pm(\frac{1}{2} \|\Lambda_u \Delta U\|_F^2 + \frac{1}{2} \|\Lambda_v \Delta V\|_F^2)$. And the function $F_k(\Delta U, \Delta V) - R_k(\Delta U, \Delta V) = \|\Lambda_u \Delta U\|_F^2 + \|\Lambda_v \Delta V\|_F^2$ achieves its global minimum at $(\Delta U, \Delta V) = (\mathbf{0}, \mathbf{0})$. Hence the first order optimality condition [46, Proposition 1] implies

$$\nabla F_k(\mathbf{0}, \mathbf{0}; D_u, D_v) = \nabla R_k(\mathbf{0}, \mathbf{0}; D_u, D_v). \quad (31)$$

Combining (27), (30) and (31), we have $\nabla F_k(\mathbf{0}, \mathbf{0}; D_u, D_v) = \nabla H_k(\mathbf{0}, \mathbf{0}; D_u, D_v)$. \square

Now we are ready to prove our convergence theorem.

Theorem 1. The sequence $\{(U_k, V_k)\}$ generated by our algorithm satisfies the following properties:

(a) $H(\cdot, \cdot)$ has sufficient decrease on the sequence $\{(U_k, V_k)\}$

$$H(U_k, V_k) - H(U_{k+1}, V_{k+1}) \geq \frac{1}{2} \|\Lambda_u (U_{k+1} - U_k)\|_F^2 + \frac{1}{2} \|\Lambda_v (V_{k+1} - V_k)\|_F^2.$$

(b) $\lim_{k \rightarrow \infty} (U_{k+1} - U_k) = \mathbf{0}$ and $\lim_{k \rightarrow \infty} (V_{k+1} - V_k) = \mathbf{0}$.

(c) The sequence $\{(U_k, V_k)\}$ is bounded.

Proof. As the zero vector is contained in the subgradient of $F_k(\Delta U, \Delta V)$ at the optimal solution $(\Delta U_k, \Delta V_k)$, there exists $(G_{U_k}, G_{V_k}) \in \partial J_k(\Delta U_k, \Delta V_k)$ such that

$$(G_{U_k}, G_{V_k}) + (\Lambda_u^2 \Delta U_k, \Lambda_v^2 \Delta V_k) = (\mathbf{0}, \mathbf{0}). \quad (32)$$

An inner product with $(\Delta U_k, \Delta V_k)$ on both sides of (32) gives

$$\langle G_{U_k}, \Delta U_k \rangle + \langle G_{V_k}, \Delta V_k \rangle + \|\Lambda_u \Delta U_k\|_F^2 + \|\Lambda_v \Delta V_k\|_F^2 = 0. \quad (33)$$

By the definition of subgradient of convex function, we have

$$J_k(\mathbf{0}, \mathbf{0}) \geq J_k(\Delta U_k, \Delta V_k) - \langle G_{U_k}, \Delta U_k \rangle - \langle G_{V_k}, \Delta V_k \rangle. \quad (34)$$

Combining (33) and (34), we can get

$$\begin{aligned} & H(U_k, V_k) - H(U_{k+1}, V_{k+1}) \\ & \geq F_k(\mathbf{0}, \mathbf{0}) - F_k(\Delta U_k, \Delta V_k) \\ & \geq \frac{1}{2} \|\Lambda_u (U_{k+1} - U_k)\|_F^2 + \frac{1}{2} \|\Lambda_v (V_{k+1} - V_k)\|_F^2, \end{aligned} \quad (35)$$

where the first inequality is derived from the property of surrogate function in (4). Thus $H(U_k, V_k)$ has sufficient decrease. Summing all the inequalities in (35) for $k \geq 1$, it follows that

$$H(U_1, V_1) \geq \frac{1}{2} \sum_{k=1}^{\infty} \|\Lambda_u(U_{k+1} - U_k)\|_F^2 + \frac{1}{2} \sum_{k=1}^{\infty} \|\Lambda_v(V_{k+1} - V_k)\|_F^2. \quad (36)$$

By the positive definiteness of Λ_u and Λ_v , we can infer that $\lim_{k \rightarrow \infty} (U_{k+1} - U_k) = \mathbf{0}$ and $\lim_{k \rightarrow \infty} (V_{k+1} - V_k) = \mathbf{0}$.

As $H(U, V) \rightarrow \infty$ when $\|U\| \rightarrow \infty$ or $\|V\| \rightarrow \infty$, we can conclude that the sequence $\{U_k, V_k\}$ is bounded. \square

Theorem 2. *Let $\{(U_k, V_k)\}$ be the sequence generated by our algorithm. Then any accumulation point of $\{(U_k, V_k)\}$ is a stationary point (U^*, V^*) of problem (8).*

Proof. By Theorem 1 the sequence $\{(U_k, V_k)\}$ is bounded. Hence the sequence $\{(U_k, V_k)\}$ has accumulation points. For any accumulation point (U^*, V^*) , there exists a subsequence $\{(U_{k_j}, V_{k_j})\}$ such that $\lim_{j \rightarrow \infty} (U_{k_j}, V_{k_j}) = (U^*, V^*)$. From the fact that $\lim_{j \rightarrow \infty} \Delta U_j = \lim_{j \rightarrow \infty} (U_{j+1} - U_j) = \mathbf{0}$ and $\lim_{j \rightarrow \infty} \Delta V_j = \lim_{j \rightarrow \infty} (V_{j+1} - V_j) = \mathbf{0}$, we have $\lim_{j \rightarrow \infty} U_{k_j+1} = U^*$ and $\lim_{j \rightarrow \infty} V_{k_j+1} = V^*$.

Since $(\Delta U_{k_j}, \Delta V_{k_j})$ minimizes $F_{k_j}(\Delta U, \Delta V)$, we have

$$\nabla F_{k_j}(\Delta U_{k_j}, \Delta V_{k_j}; D_u, D_v) \geq 0, \quad \forall (D_u, D_v). \quad (37)$$

As $j \rightarrow \infty$, $(\Delta U_{k_j}, \Delta V_{k_j})$ approaches to $(\mathbf{0}, \mathbf{0})$. By Proposition 1, we have

$$\nabla H_{k_j}(\mathbf{0}, \mathbf{0}; D_u, D_v) = \nabla F_{k_j}(\mathbf{0}, \mathbf{0}; D_u, D_v) \geq 0, \quad \forall (D_u, D_v). \quad (38)$$

As $H_k(\Delta U, \Delta V) = H_k(U - U_k, V - V_k) = H(U, V)$, we have

$$\nabla H(U^*, V^*; D_u, D_v) \geq 0, \quad \forall (D_u, D_v). \quad (39)$$

By Definition 2, we can conclude that (U^*, V^*) is a stationary point of (8). \square

6 EXPERIMENTAL RESULTS

In this section, we compare our RMF-MM with several state-of-the-art algorithms for RMF, including L_1 -Wib¹ [18], CWM² [19], ARG-D [25], LMAFit³ [20], Reg L_1 ⁴ [21] and UNuBi [22]. The code of ARG-D and UNuBi are kindly provided by their authors. We modify the original code of LMAFit so that it fits for missing data with a fixed rank. L_1 -ALP/AQP [17] are not included here as they have been shown to be much inferior to the above methods [18], [19], [21]. All the codes are run in Matlab on a desktop PC with a 3.4 GHz CPU and 8 GB RAM.

For the monotonically decreasing algorithms, L_1 -Wib, CWM, ARG-D and our RMF-MM, we stop them when the relative change of the objective function is less than 10^{-4} . For the ADMM based algorithms, LMAFit, Reg L_1 and UNuBi, we terminate them when

$$\max(\|E_k - E_{k-1}\|_F, \|U_k - U_{k-1}\|_F, \|V_k - V_{k-1}\|_F) / \|M\|_F < 10^{-4}, \quad (40)$$

$$\|E_k - U_k V_k^T\|_F / \|M\|_F < 10^{-4}, \quad (41)$$

where E_k is the auxiliary variable. As LMAFit is hard to reach such a precision using a fixed penalty parameter when the data are noisy, we tune the penalty parameter and the maximum iteration number in each experiment.

For the inner iterations in our RMF-MM algorithm, we set $\varepsilon_1 = 10^{-5}$, $\varepsilon_2 = 10^{-4}$, $\rho_0 = 1.5$, and $\beta^{\max} = 10^{10}$ as default values. For algorithms for regularized models, Reg L_1 , UNuBi and RMF-MM, we simply set the regularization parameter $\lambda = 20/(m+n)$. In the following, unless explicitly mentioned all the algorithms are initialized with the rank- r truncation of the singular-value decomposition of $W \odot M$ [18].

6.1 Synthetic Data

We first compare all the methods on synthetic data. We generate data matrices $M = U_0 V_0^T$, where $U_0 \in \mathbb{R}^{m \times r}$ and $V_0 \in \mathbb{R}^{n \times r}$. The entries of U_0 and V_0 are sampled i.i.d. from a Gaussian distribution $\mathcal{N}(0, 1)$. Then Gaussian noise $\mathcal{N}(0, 0.1)$ is added independently to every entry of M . We additionally corrupt a portion of entries with outliers uniformly distributed in $[-\sigma, \sigma]$, where $\sigma > 0$ represents the magnitude of outliers. The positions of both outliers and missing data are chosen uniformly at random. We denote the missing data ratio as s percent and the outliers ratio as o percent, respectively. Two metrics are used to measure the performance. One is the optimized L_1 -norm error between the reconstructed matrix M_{est} and the corrupted matrix M

$$\text{Err}_1 = \|W \odot (M_{est} - M)\|_1 / \#W, \quad (42)$$

where we normalize the error with the number $\#W$ of observed entries. The other metric is the L_1 -norm error between M_{est} and the ground truth matrix M_0 ,

$$\text{Err}_2 = \|M_{est} - M_0\|_1 / (mn). \quad (43)$$

Experiments are conducted at two sizes: $(m = 20, n = 30, r = 4)$ and $(m = 200, n = 300, r = 4)$. Due to the significant memory requirement of the L_1 -Wib algorithm [21], [22], we do not include it in the large scale experiments. At each size, we test the algorithms by varying one of the three hyper parameters (missing data ratio s percent, outliers ratio o percent and outliers magnitude σ) and fixing the other two. At each combination of these hyper parameters, we repeat the experiments 50 times. All the compared algorithms use the same data in each trial. The results are summarized at Table 1. Ave₁ and Std₁ denote the average and the standard derivation of Err₁, respectively. Ave₂ and Std₂ are defined similarly for Err₂. The least Ave₁, Std₁, Ave₂ and Std₂ among all algorithms are presented in bold-face. To account for the inconsistency in the termination criteria, the four measurements within a difference of 10^{-3} from the least ones are also shown in bold fonts.

In the small scale data experiments, our RMF-MM consistently gives the smallest Ave₁ and Std₁ except when s, o percent or σ is very large. L_1 -Wib and LMAFit are a little inferior to our algorithm in Ave₁ and Std₁. However, the two algorithms seldom achieve the least Ave₂ and Std₂. We attribute this to the Frobenius norm regularizations in (8). In general, the regularized model, e.g., our RMF-MM, is less

1. <http://cs.adelaide.edu.au/~anders/code/cvpr2010.html>
2. <http://gr.xjtu.edu.cn/web/dymeng/3>
3. <http://lmafit.blogs.rice.edu/>
4. <https://sites.google.com/site/yinqiangzheng/>

TABLE 1
Synthetic Experiments on Two Sizes of Data with Varying Missing Data Ratio s Percent,
Outliers Ratio o Percent and Outliers Magnitude σ

| $m = 20, n = 30$ $r = 4$ | | $s\%$ varies, $o\% = 15\%$ and $\sigma = 9$ | | | | $s\% = 20\%$, $o\%$ varies and $\sigma = 9$ | | | | $s\% = 20\%$, $o\% = 15\%$ and σ varies | | | |
|-------------------------------|------------------|---|--------------|--------------|--------------|--|--------------|--------------|--------------|---|--------------|--------------|--------------|
| | | 10% | 15% | 20% | 25% | 5% | 10% | 15% | 20% | 5 | 7 | 9 | 11 |
| L ₁ -Wib [18] | Ave ₁ | 0.713 | 0.701 | 0.716 | 0.687 | 0.271 | 0.483 | 0.716 | 0.906 | 0.413 | 0.564 | 0.716 | 0.859 |
| | Std ₁ | 0.089 | 0.095 | 0.090 | 0.103 | 0.056 | 0.072 | 0.090 | 0.082 | 0.049 | 0.069 | 0.090 | 0.104 |
| | Ave ₂ | 0.115 | 0.784 | 0.662 | 1.722 | 0.072 | 0.136 | 0.662 | 1.313 | 0.126 | 0.253 | 0.662 | 1.002 |
| | Std ₂ | 0.086 | 3.950 | 1.873 | 4.705 | 0.007 | 0.239 | 1.873 | 1.773 | 0.041 | 0.516 | 1.873 | 2.504 |
| CWM [19] | Ave ₁ | 0.753 | 0.763 | 0.796 | 0.788 | 0.290 | 0.540 | 0.796 | 1.005 | 0.441 | 0.617 | 0.796 | 0.961 |
| | Std ₁ | 0.105 | 0.109 | 0.106 | 0.117 | 0.064 | 0.094 | 0.106 | 0.090 | 0.055 | 0.082 | 0.106 | 0.126 |
| | Ave ₂ | 0.287 | 0.386 | 0.508 | 0.656 | 0.126 | 0.301 | 0.508 | 0.806 | 0.246 | 0.373 | 0.508 | 0.656 |
| | Std ₂ | 0.171 | 0.191 | 0.207 | 0.247 | 0.070 | 0.162 | 0.207 | 0.193 | 0.077 | 0.151 | 0.207 | 0.279 |
| ARG-D [25] | Ave ₁ | 0.715 | 0.704 | 0.719 | 0.700 | 0.272 | 0.485 | 0.719 | 0.927 | 0.416 | 0.569 | 0.719 | 0.876 |
| | Std ₁ | 0.090 | 0.098 | 0.090 | 0.103 | 0.056 | 0.073 | 0.090 | 0.084 | 0.050 | 0.071 | 0.090 | 0.110 |
| | Ave ₂ | 0.130 | 0.151 | 0.237 | 0.384 | 0.076 | 0.110 | 0.237 | 0.575 | 0.143 | 0.168 | 0.237 | 0.355 |
| | Std ₂ | 0.080 | 0.071 | 0.181 | 0.259 | 0.009 | 0.061 | 0.181 | 0.270 | 0.043 | 0.069 | 0.181 | 0.275 |
| LMaFit [20] | Ave ₁ | 0.713 | 0.701 | 0.718 | 0.680 | 0.271 | 0.484 | 0.718 | 0.902 | 0.414 | 0.567 | 0.718 | 0.853 |
| | Std ₁ | 0.089 | 0.095 | 0.088 | 0.097 | 0.056 | 0.072 | 0.088 | 0.079 | 0.049 | 0.069 | 0.088 | 0.097 |
| | Ave ₂ | 0.160 | 0.141 | 0.285 | 0.543 | 0.072 | 0.120 | 0.285 | 0.752 | 0.126 | 0.181 | 0.285 | 0.615 |
| | Std ₂ | 0.202 | 0.110 | 0.360 | 0.492 | 0.007 | 0.127 | 0.360 | 0.506 | 0.042 | 0.197 | 0.360 | 0.617 |
| RegL ₁ [21] | Ave ₁ | 0.719 | 0.705 | 0.718 | 0.687 | 0.272 | 0.485 | 0.718 | 0.911 | 0.414 | 0.565 | 0.718 | 0.872 |
| | Std ₁ | 0.096 | 0.096 | 0.090 | 0.101 | 0.060 | 0.075 | 0.090 | 0.080 | 0.049 | 0.069 | 0.090 | 0.103 |
| | Ave ₂ | 0.174 | 0.192 | 0.283 | 0.374 | 0.085 | 0.118 | 0.283 | 0.589 | 0.117 | 0.160 | 0.283 | 0.551 |
| | Std ₂ | 0.188 | 0.231 | 0.259 | 0.317 | 0.070 | 0.087 | 0.259 | 0.352 | 0.027 | 0.128 | 0.259 | 0.333 |
| UNuBi [22] | Ave ₁ | 0.713 | 0.703 | 0.717 | 0.688 | 0.272 | 0.484 | 0.717 | 0.913 | 0.414 | 0.567 | 0.717 | 0.863 |
| | Std ₁ | 0.089 | 0.096 | 0.087 | 0.102 | 0.060 | 0.072 | 0.087 | 0.088 | 0.049 | 0.069 | 0.087 | 0.105 |
| | Ave ₂ | 0.118 | 0.147 | 0.208 | 0.322 | 0.080 | 0.105 | 0.208 | 0.536 | 0.116 | 0.148 | 0.208 | 0.403 |
| | Std ₂ | 0.077 | 0.121 | 0.188 | 0.301 | 0.059 | 0.054 | 0.188 | 0.316 | 0.027 | 0.098 | 0.188 | 0.344 |
| RMF-MM | Ave ₁ | 0.713 | 0.701 | 0.714 | 0.685 | 0.271 | 0.483 | 0.714 | 0.909 | 0.414 | 0.565 | 0.714 | 0.861 |
| | Std ₁ | 0.089 | 0.095 | 0.088 | 0.098 | 0.056 | 0.072 | 0.088 | 0.080 | 0.049 | 0.069 | 0.088 | 0.104 |
| | Ave ₂ | 0.113 | 0.118 | 0.178 | 0.292 | 0.071 | 0.105 | 0.178 | 0.427 | 0.117 | 0.136 | 0.178 | 0.248 |
| | Std ₂ | 0.076 | 0.042 | 0.153 | 0.266 | 0.006 | 0.054 | 0.153 | 0.317 | 0.027 | 0.054 | 0.153 | 0.244 |
| $m = 200, n = 300$ $r = 4$ | | $s\%$ varies, $o\% = 35\%$ and $\sigma = 9$ | | | | $s\% = 85\%$, $o\%$ varies and $\sigma = 9$ | | | | $s\% = 85\%$, $o\% = 35\%$ and σ varies | | | |
| | | 75% | 80% | 85% | 90% | 25% | 30% | 35% | 40% | 5 | 7 | 9 | 11 |
| CWM [19] | Ave ₁ | 1.619 | 1.614 | 1.641 | 1.659 | 1.173 | 1.404 | 1.641 | 1.862 | 0.915 | 1.274 | 1.641 | 2.036 |
| | Std ₁ | 0.022 | 0.026 | 0.040 | 0.037 | 0.026 | 0.024 | 0.040 | 0.030 | 0.018 | 0.026 | 0.040 | 0.050 |
| | Ave ₂ | 0.105 | 0.185 | 0.618 | 1.767 | 0.195 | 0.338 | 0.618 | 0.932 | 0.313 | 0.435 | 0.618 | 0.989 |
| | Std ₂ | 0.007 | 0.018 | 0.119 | 0.095 | 0.023 | 0.048 | 0.119 | 0.085 | 0.031 | 0.054 | 0.119 | 0.190 |
| ARG-D [25] | Ave ₁ | 1.617 | 1.607 | 1.599 | 1.526 | 1.161 | 1.381 | 1.599 | 1.815 | 0.899 | 1.248 | 1.599 | 1.969 |
| | Std ₁ | 0.022 | 0.025 | 0.033 | 0.035 | 0.025 | 0.023 | 0.033 | 0.030 | 0.017 | 0.026 | 0.033 | 0.052 |
| | Ave ₂ | 0.082 | 0.125 | 0.351 | 1.914 | 0.120 | 0.182 | 0.351 | 0.777 | 0.199 | 0.256 | 0.351 | 0.704 |
| | Std ₂ | 0.005 | 0.010 | 0.074 | 0.136 | 0.010 | 0.024 | 0.074 | 0.123 | 0.020 | 0.036 | 0.074 | 0.291 |
| LMaFit [20] | Ave ₁ | 1.667 | 1.662 | 1.647 | 1.483 | 1.184 | 1.417 | 1.647 | 1.849 | 0.914 | 1.277 | 1.647 | 2.019 |
| | Std ₁ | 0.025 | 0.029 | 0.033 | 0.028 | 0.027 | 0.024 | 0.033 | 0.028 | 0.018 | 0.026 | 0.033 | 0.038 |
| | Ave ₂ | 0.170 | 0.210 | 0.358 | 3.020 | 0.139 | 0.209 | 0.358 | 0.898 | 0.201 | 0.265 | 0.358 | 0.516 |
| | Std ₂ | 0.017 | 0.018 | 0.053 | 0.241 | 0.009 | 0.018 | 0.053 | 0.243 | 0.018 | 0.031 | 0.053 | 0.116 |
| RegL ₁ [21] | Ave ₁ | 1.616 | 1.606 | 1.597 | 1.454 | 1.161 | 1.381 | 1.597 | 1.808 | 0.898 | 1.246 | 1.597 | 1.977 |
| | Std ₁ | 0.022 | 0.025 | 0.031 | 0.028 | 0.025 | 0.022 | 0.031 | 0.028 | 0.017 | 0.024 | 0.031 | 0.042 |
| | Ave ₂ | 0.077 | 0.109 | 0.275 | 3.621 | 0.107 | 0.164 | 0.275 | 1.154 | 0.176 | 0.210 | 0.275 | 1.455 |
| | Std ₂ | 0.003 | 0.008 | 0.085 | 0.202 | 0.023 | 0.060 | 0.085 | 0.393 | 0.017 | 0.029 | 0.085 | 0.618 |
| UNuBi [22] | Ave ₁ | 1.616 | 1.606 | 1.597 | 1.454 | 1.160 | 1.380 | 1.597 | 1.807 | 0.898 | 1.246 | 1.597 | 1.978 |
| | Std ₁ | 0.022 | 0.025 | 0.033 | 0.028 | 0.025 | 0.024 | 0.033 | 0.028 | 0.017 | 0.024 | 0.033 | 0.044 |
| | Ave ₂ | 0.077 | 0.109 | 0.285 | 3.620 | 0.104 | 0.150 | 0.285 | 1.121 | 0.176 | 0.210 | 0.285 | 1.512 |
| | Std ₂ | 0.003 | 0.008 | 0.130 | 0.218 | 0.007 | 0.040 | 0.130 | 0.399 | 0.017 | 0.029 | 0.130 | 0.683 |
| RMF-MM | Ave ₁ | 1.616 | 1.606 | 1.595 | 1.435 | 1.160 | 1.380 | 1.595 | 1.803 | 0.898 | 1.246 | 1.595 | 1.948 |
| | Std ₁ | 0.022 | 0.025 | 0.031 | 0.029 | 0.025 | 0.023 | 0.031 | 0.028 | 0.017 | 0.024 | 0.031 | 0.042 |
| | Ave ₂ | 0.077 | 0.108 | 0.247 | 2.617 | 0.104 | 0.146 | 0.247 | 0.672 | 0.176 | 0.207 | 0.247 | 0.444 |
| | Std ₂ | 0.003 | 0.007 | 0.040 | 0.234 | 0.008 | 0.017 | 0.040 | 0.175 | 0.016 | 0.025 | 0.040 | 0.135 |

Ave₁ and Std₁ (Ave₂ and Std₂) represent the average and the standard derivation of Err₁ (Err₂) across 50 repeats. The best values within a difference of 10^{-3} are in bold fonts.

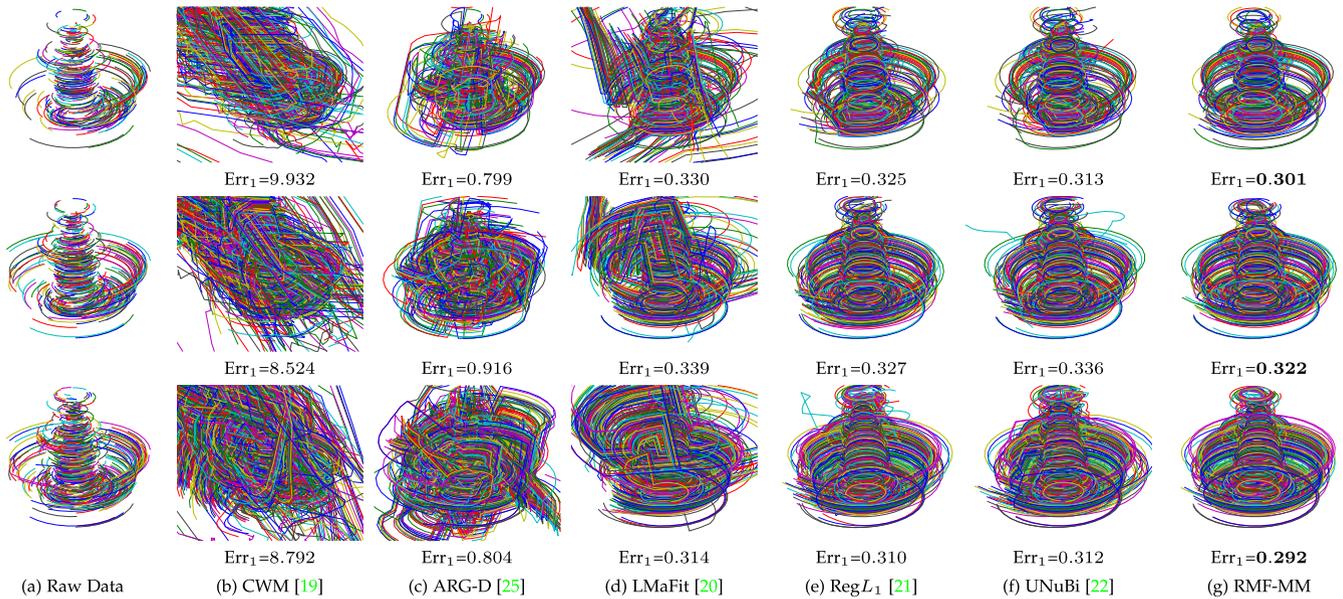


Fig. 3. Original incomplete and recovered data of the Dinosaur sequence with at least 7, 6 and 5 views, respectively. (a) Raw input tracks. (b-g) Full tracks reconstructed by ALP [17], CWM [19], LMaFit [20], $\text{Reg}L_1$ [21], UNuBi [22] and RMF-MM, respectively. The reconstruction errors Err_1 defined as (42) are presented below the tracks.

likely to overfit the corrupted data than the unregularized ones, e.g., L_1 -Wib and LMaFit. Thus we can give more accurate estimate on the ground truth matrices, i.e., the smallest Ave_2 and Std_2 , across almost all combinations of the three hyper parameters. When s , o percent or σ is relatively small, UNuBi can also achieve reasonably good performance.

In the large scale data experiments, our RMF-MM again reaches the smallest values of the four measurements in most cases. When $s\% = 90\%$, although we get much smaller Ave_1 and Std_1 than CWM, the corresponding Ave_2 and Std_2 are much worse. The reason may be the ill conditioning caused by the very high missing data ratio s percent. When $o\% = 40\%$ or $\sigma = 11$, the achieved standard derivation of our RMF-MM is not the smallest. But combining with the far smaller average value, we may still conclude that our algorithm achieves the best performance in these two cases. When s, m percent, or σ is relatively small, $\text{Reg}L_1$ and UNuBi also recover the matrix in high accuracy. Overall, the algorithms with regularized model, i.e., $\text{Reg}L_1$, UNuBi and RMF-MM, are better than the unregularized ones, i.e., CWM, ARG-D and LMaFit.

To have a better understanding on the convergence rate of our RMF-MM, we compare with UNuBi on large scale data ($s\% = 85\%$, $o\% = 30\%$, and $\sigma = 9$). Both algorithms have the same objective function (8). As Fig. 2 shows, our RMF-MM converges quickly after only a few iterations. By contrast, it takes more time for UNuBi to reach reasonably small objective values. From the curve, we can also see that it costs much more time in the first iteration for our RMF-MM. This is because that we fix the initial small diagonal entries of Λ_u and Λ_v across all experiments, and it takes time for our continuation technique to reach the proper values of Λ_u and Λ_v for this specific task.

6.2 Real Data

In this section, we compare the performance of various algorithms on real data. We do not include L_1 -Wib here again due to its high memory requirement[21], [22].

6.2.1 Affine Rigid SfM

As first proposed in [1], Tomasi and Kanade modelled the orthographic rigid SfM, a special case of affine rigid SfM, as a matrix factorization problem. When there are no missing data and outliers, it can be formulated as a rank-3 factorization problem after registering the image origin to the centroid of feature points in every frame. Unfortunately, such an ideal scenario does not exist in practice. We thus model this problem as a rank-4 matrix factorization [18], [19], [21], [22].

Here we use the famous Oxford Dinosaur sequence,⁵ which consists of 36 images with a resolution of 720×576 . We pick out a portion of the raw features which are observed by at least e views. We conduct three sets of experiments with e chosen as be 7, 6 and 5, respectively. The raw features are shown in Fig. 3a. The corresponding observed matrices are of size 72×336 with $s\% = 76.9\%$, 72×557 with $s\% = 79.5\%$, and 72×932 with $s\% = 82.1\%$, respectively. The observed matrices have a band diagonal pattern with outliers and the missing data ratio increases as e decreases. Thus it is very challenging to conduct SfM, especially when e is small.

We first register the image origin to the image center, (360, 288), by which the intrinsic rank remains to be four under the affine SfM model. Since there is no ground truth, we only measure the reconstruction error Err_1 defined as (42). Figs. 3b, 3c, 3d, 3e, and 3f show the full tracks reconstructed by all algorithms. And the values of Err_1 are shown under the tracks. As the dinosaur sequence is taken on a turntable, all the tracks are expected to be circular. Among them, the tracks reconstructed by CWM are the most inferior. Its reconstruction errors are higher by orders than others. This may be caused by its alternate scheme, which easily gets struck at non-stationary solutions. ARG-D and LMaFit perform better but quite a few tracks also diverge. $\text{Reg}L_1$ and UNuBi give reasonably good results. However,

5. <http://www.robots.ox.ac.uk/~vgg/data1.html>

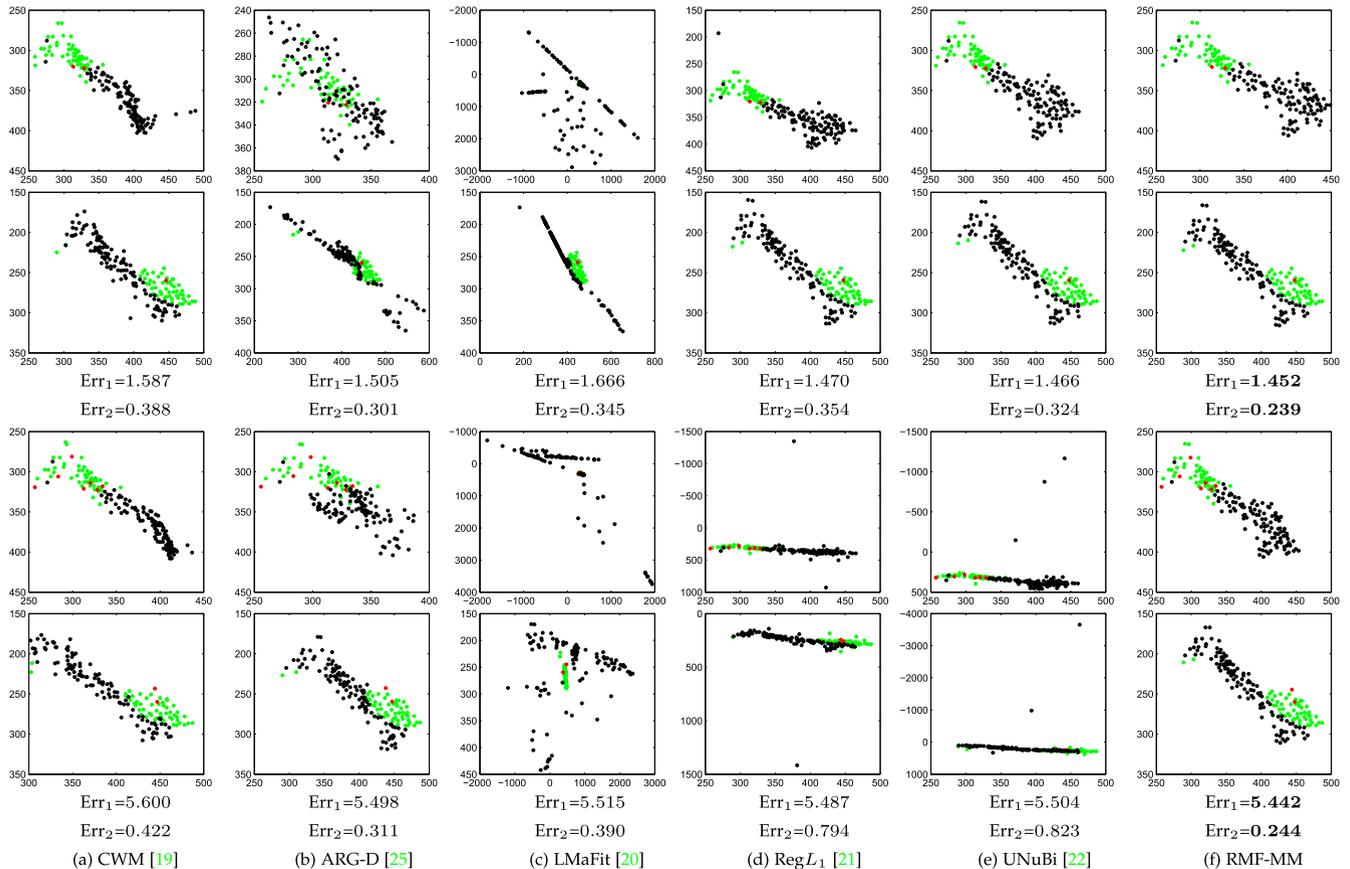


Fig. 4. Recovered points for the first (first and third rows) and the 50th (second and fourth rows) frames of the Giraffe sequence on two levels of outliers ($o\% = 5\%$ and $\sigma = 50$ for the first two rows and $o\% = 10\%$ and $\sigma = 100$ for the last two rows). A red dot represents an observed outliers we randomly generated. Green is an observed entry, and black is a missing entry. Err_1 and Err_2 are presented below the figures (*best viewed on screen*).

most of the reconstructed tracks in large radii are not smooth at their ends, and there are some wild tracks which obviously fail to be reconstructed. By contrast, almost all the tracks reconstructed by our RMF-MM in Fig. 3g are circular, which exhibit the most plausible visual performance. The consistently lowest reconstruction errors also confirm the effectiveness of our RMF-MM.

6.2.2 Nonrigid SfM

As Bregler et al. [2] pointed out, nonrigid motions could be reconstructed from an image sequence via rank-3d matrix factorization, where d is the number of shape basis accounting for nonrigid deformation.

We use the Giraffe sequence,⁶ which contains 166 feature points tracked over 120 successive frames. The measurement matrix M is of size 240×166 with 30.2 percent entries missing. Since it is somehow cleaned [12] for the L_2 -norm based methods, we imitate the experimental setup of [21] by uniformly adding outliers in the range of $[-\sigma, \sigma]$ to o percent of the observed entries. For comparison, we add two levels of outliers with ($o\% = 5\%$, $\sigma = 50$) and ($o\% = 10\%$, $\sigma = 100$), respectively. We set d to 2, leading to a rank-6 matrix factorization problem. We initialize all algorithms with the same random matrix.

Fig. 4 shows the recovered giraffe shape feature points in the first (first and third rows) and the 50th (second and

fourth rows) frames of the sequence. We report Err_1 (42) and Err_2 (43)⁷ below the figures. Among all the competing methods, LMafit gives the worst visual results in the sampled frames, although it reaches reasonably small Err_2 . We may attribute this phenomenon to overfitting of the unregularized model. This may also hold for ARG-D when observing its out-of-shaped frames and its low Err_2 . CWM gives reasonably good reconstruction. However, there are still some wild feature points that obviously fail to recover. When the intensity of outliers increases, the neck of the giraffe in the first frame and the head in the 50th frame are out of shape. RegL1 and UNuBi show a great gap in the two levels of outliers. Both algorithms seem to be more sensitive to outliers. By contrast, our RMF-MM consistently gives the best visual results. The lowest errors also confirm the effectiveness of our algorithm.

6.2.3 Image Recovery

As pointed out in [7], many images could be regarded as low rank matrices, where the top singular values dominate the main information. Thus we can employ low rank approximation to recover corrupted images. Figs. 5a and 6a⁸ show typical ones of size 300×300 , where the unrelated texts to be removed are masked as missing data. To show the robustness of the L_1 -norm based methods, we uniformly

7. As the ground truth is with missing data, similar to Err_1 , we compute Err_2 over the observed entries only.

8. <http://sites.google.com/site/zjyuaohu/>

6. <http://www.robots.ox.ac.uk/~amb/>

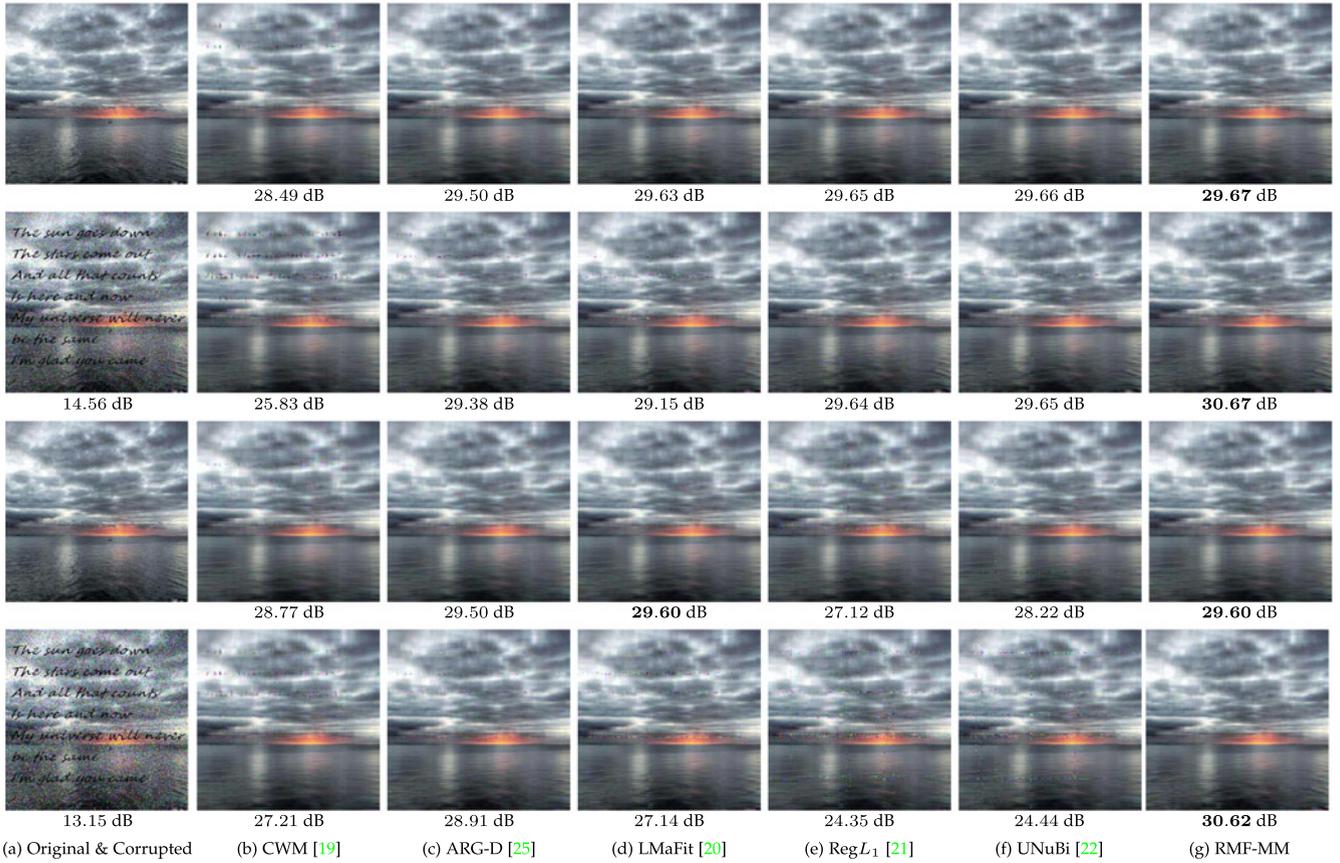


Fig. 5. Comparison of image recovery after adding different magnitudes of outliers and by choosing different ranks. The first to the fourth rows correspond to $(\sigma = 150, r = 15)$, $(\sigma = 150, r = 20)$, $(\sigma = 250, r = 15)$ and $(\sigma = 250, r = 20)$, respectively. The PSNRs are presented below each image (best viewed on screen).

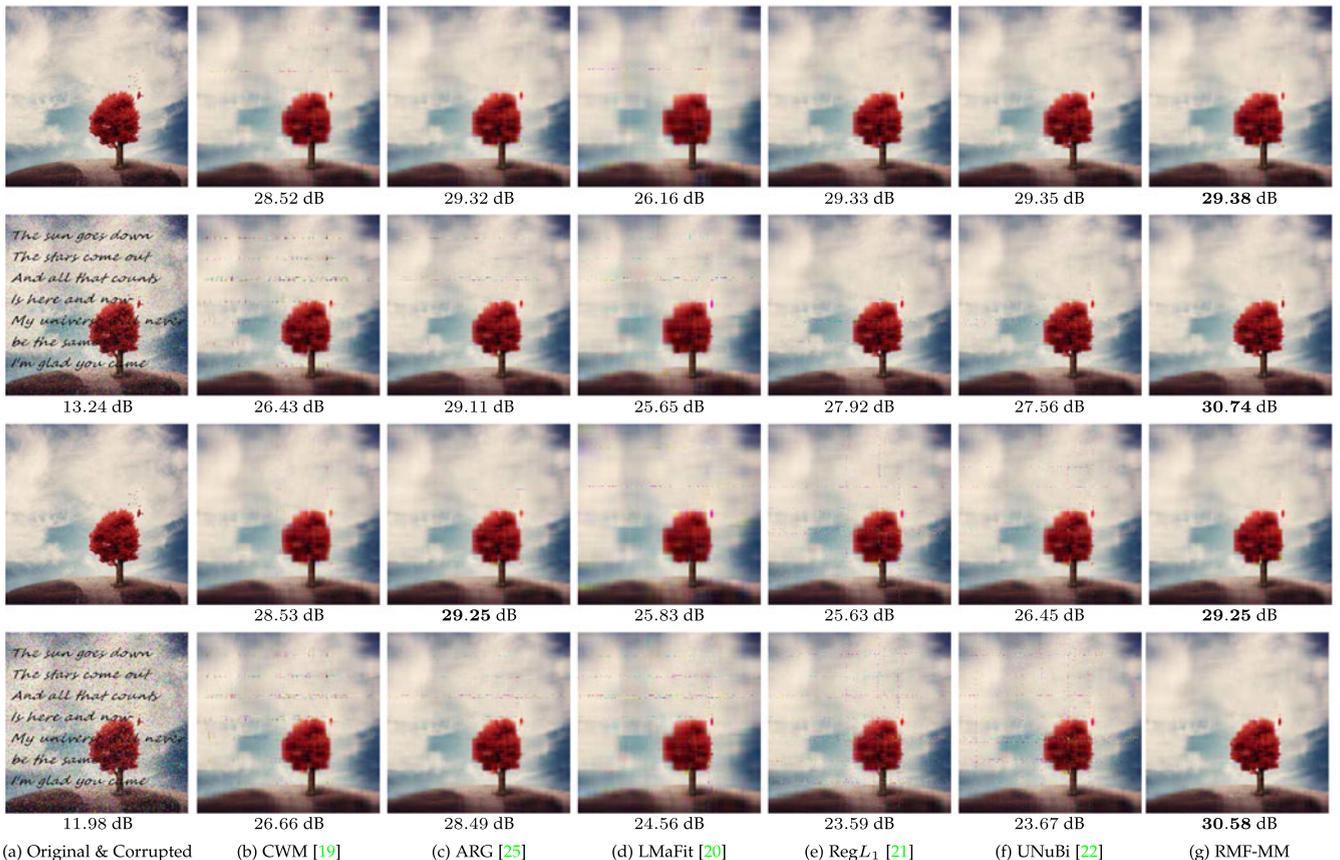


Fig. 6. The description is the same as Fig. 5.

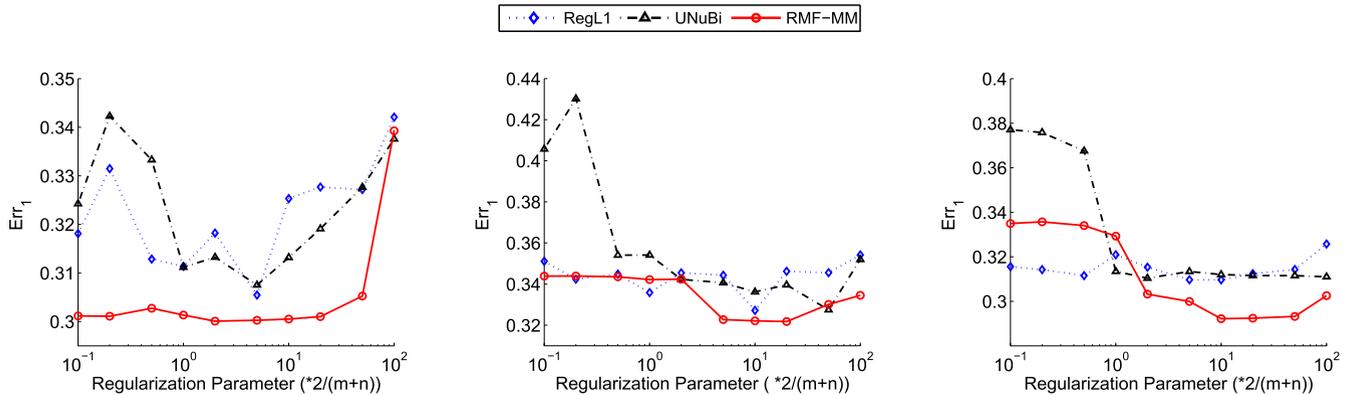


Fig. 7. Err_1 versus the regularization parameter λ for all the regularized methods, e.g., $RegL_1$ [21], UNuBi [22] and RMF-MM, on the Dinosaur data set. The curves from left to right correspond to data sequence with at least 7, 6 and 5 views, respectively.

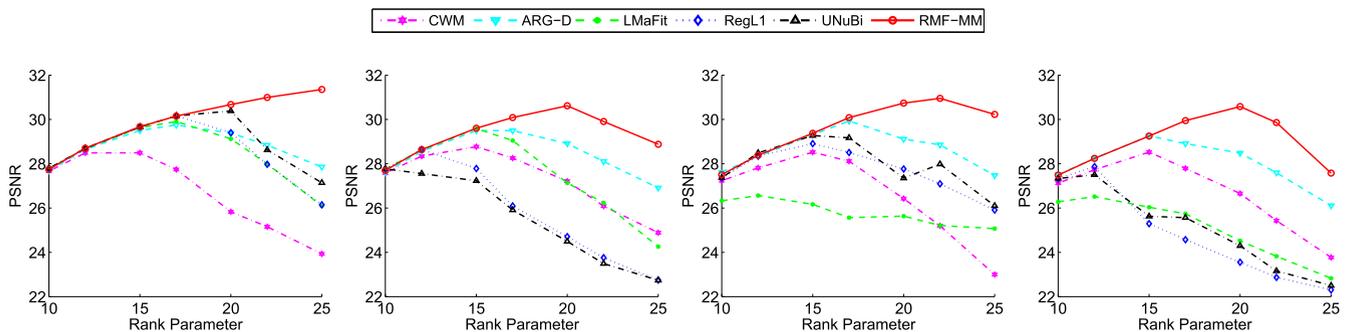


Fig. 8. PSNR versus the rank r on the image recovery task. The first two images correspond to Fig. 5 with outliers magnitude $\sigma = 150$ and 250, respectively. The last two correspond to Fig. 6 similarly.

TABLE 2
Average Computing Time (Seconds) of the Competing Algorithms on Different Data Sets

| Experiments/ Algorithms | L_1 -Wib [18] | CWM [19] | ARG-D [25] | LMaFit [20] | $RegL_1$ [21] | UNuBi [22] | RMF-MM |
|----------------------------|-----------------|------------|-------------|-------------|---------------|-------------|------------|
| Small scale synthetic data | 61.4 | 0.1 | 0.2 | 3.0 | 2.6 | 0.8 | 0.5 |
| Large scale synthetic data | - | 0.9 | 5.9 | 19.9 | 36.4 | 16.8 | 14.3 |
| Affine Rigid SfM (7 views) | - | 17.1 | 138.1 | 270.2 | 20.3 | 10.4 | 47.9 |
| Affine Rigid SfM (6 views) | - | 59.3 | 96.1 | 362.8 | 30.4 | 14.9 | 67.1 |
| Affine Rigid SfM (5 views) | - | 87.2 | 254.5 | 569.3 | 47.2 | 23.7 | 120.1 |
| Nonrigid SfM | - | 13.9 | 10.7 | 79.9 | 32.8 | 19.9 | 9.7 |
| Image Recovery | - | 94.7 | 71.5 | 471.3 | 295.2 | 131.6 | 305.4 |

add outliers in the range of $[-\sigma, \sigma]$ to 20 percent of the observed pixels. For color images which have three channels, we implement matrix factorization on each channel independently.

Figs. 5b, 5c, 5d, 5e, 5f, 5g and 6b, 6c, 6d, 6e, 6f, 6g show the recovered results by all six compared methods on the corrupted images. Here we vary the outliers magnitude σ and the chosen rank r . In the first to the fourth rows, the figures correspond to results of $(\sigma = 150, r = 15)$, $(\sigma = 150, r = 20)$, $(\sigma = 250, r = 15)$ and $(\sigma = 250, r = 20)$, respectively. Overall, our RMF-MM consistently produces good visual quality with few noticeable artifacts. Whereas other methods can only succeed for at most one or two cases. In the remaining cases, their results are more or less with artifacts of some horizontal (and vertical) lines. Besides the visual results, we report the Signal-to-Noise Ratios (PSNRs) below images and highlight the best in boldface. Our RMF-MM consistently achieves the highest PSNRs on the two tested images in all cases.

6.3 Parameter Sensitivity Analysis

As mentioned in the beginning of Section 6, we fix the regularization parameter $\lambda = 20/(m+n)$ for all the regularized methods, i.e., $RegL_1$, UNuBi and RMF-MM, across all experiments. In this section, we first test the sensitivity with respect to λ . We conduct experiments on the Dinosaur sequence by varying λ in the range of $[0.1, 100] * 2/(m+n)$. The results are shown in Fig. 7. In all the three trials, our RMF-MM can produce almost constant Err_1 in a wide range of λ . It also reaches the lowest values among all methods in most cases.

We also test the effect of rank parameter on the performance of Image Recovery. The experiments are conducted on a wider range of ranks, e.g., $[10, 25]$. As Fig. 8 shows, our RMF-MM consistently reaches the highest PSNRs. Compared with other methods, it is less sensitive to the changes of rank. Especially, when the rank is large, all other methods exhibit an obvious degradation as rank increases. By contrast, the PSNRs of our RMF-MM only drop by a small amount or even increase.

6.4 Computing Time

We show in Table 2 the average computing time of all the competing algorithms on all data sets. Note that the computing time in image recovery is the sum of those on three (R, G and B) channels. RMF-MM costs the least in Nonrigid SfM. On other data sets, although RMF-MM is not the fastest, its computing time is at the same scale as those of others. If considering the quality of solutions, RMF-MM should still be highly preferable.

7 CONCLUSION

In this paper, we apply the Majorization Minimization technique to solve the L_1 -norm based low rank matrix factorization problem in the presence of both missing data and outliers. By constructing a novel convex surrogate, the corresponding strongly convex sub-problem can be minimized by LADMPSAP efficiently. Moreover, the objective function has sufficient decrease. Accordingly, we are able to prove that any limit point of RMF-MM is a stationary point, which might be the best possible convergence result for non-convex optimization. To our best knowledge, this is the first convergence guarantee for the RMF problem without extra assumptions. Experiments on both synthetic and real data sets demonstrate that our RMF-MM can outperform the state-of-the-art robust factorization algorithms in robustness and accuracy. Its speed is also highly competitive.

ACKNOWLEDGMENTS

The authors would like to thank Canyi Lu for fruitful discussions and the reviewers for their comments that help improve the manuscript. Z. Lin is supported by National Basic Research Program of China (973 Program) (grant no. 2015CB352502), National Natural Science Foundation (NSF) of China (grant nos. 61625301 and 61231002), and Qualcomm. H. Zha is supported by Beijing Municipal Natural Science Foundation (grant no. 4152006).

REFERENCES

- C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *Int. J. Comput. Vis.*, vol. 9, no. 2, pp. 137–154, 1992.
- C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3D shape from image streams," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2000, vol. 2, pp. 690–696.
- H. Hayakawa, "Photometric stereo under a light source with arbitrary motion," *J. Opt. Soc. America A*, vol. 11, no. 11, pp. 3079–3089, 1994.
- R. Basri, D. Jacobs, and I. Kemelmacher, "Photometric stereo with general, unknown lighting," *Int. J. Comput. Vis.*, vol. 72, no. 3, pp. 239–257, 2007.
- Q. Ke and T. Kanade, "A subspace approach to layer extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 1, pp. 1-255–1-262.
- M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cogn. Neurosci.*, vol. 3, no. 1, pp. 71–86, 1991.
- Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.
- E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, 2011, Art. no. 11.
- B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan, "Multi-task low-rank affinity pursuit for image segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2439–2446.
- F. Xiong, O. I. Camps, and M. Sznajder, "Dynamic context for tracking behind occlusions," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 580–593.
- E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.
- A. M. Buchanan and A. W. Fitzgibbon, "Damped Newton algorithms for matrix factorization with missing data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 316–322.
- T. Okatani and K. Deguchi, "On the Wiberg algorithm for matrix factorization in the presence of missing components," *Int. J. Comput. Vis.*, vol. 72, no. 3, pp. 329–337, 2007.
- K. Mitra, S. Sheorey, and R. Chellappa, "Large-scale matrix factorization with missing data under additional constraints," in *Proc. Advances Neural Inf. Process. Syst.*, 2010, pp. 1651–1659.
- Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Programm. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.
- A. D. Bue, J. Xavier, L. Agapito, and M. Paladini, "Bilinear modeling via augmented Lagrange multipliers (BALM)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1496–1508, Aug. 2012.
- Q. Ke and T. Kanade, "Robust L_1 norm factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 739–746.
- A. Eriksson and A. van den Hengel, "Efficient computation of robust weighted low-rank matrix approximations using the L_1 norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1681–1690, Sep. 2012.
- D. Meng, Z. Xu, L. Zhang, and Ji Zhao, "A cyclic weighted median method for l_1 low-rank matrix factorization with missing entries," in *Proc. 27th AAAI Conf. Artif. Intell.*, 2013, pp. 704–710.
- Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optimization Methods Softw.*, vol. 29, no. 2, pp. 239–263, 2014.
- Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust L_1 -norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1410–1417.
- R. Cabral, F. De La Torre, J. P. Costeira, and A. Bernardino, "Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2488–2495.
- F. De La Torre and M. J. Black, "A framework for robust subspace learning," *Int. J. Comput. Vis.*, vol. 54, no. 1–3, pp. 117–142, 2003.
- D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.
- E. Kim, M. Lee, C.-H. Choi, N. Kwak, and S. Oh, "Efficient L_1 -norm-based low-rank matrix approximations for large-scale problems using alternating rectified gradient method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 237–251, Feb. 2015.
- K. Lange, D. R. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *J. Comput. Graph. Statist.*, vol. 9, no. 1, pp. 1–20, 2000.
- D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, 2004.
- J. Mairal, "Optimization with first-order surrogate functions," *Int. Conf. Mach. Learn.*, 2013, pp. 783–791.
- Y. Nesterov, "Gradient methods for minimizing composite functions," *Mathematical Programming*, vol. 140, no. 1, pp. 125–161, Aug. 2013.
- A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.
- S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.
- R. Horst and N. V. Thoai, "DC programming: Overview," *J. Optimization Theory Appl.*, vol. 103, no. 1, pp. 1–43, 1999.
- M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, AdaBoost and Bregman distances," *Mach. Learn.*, vol. 48, no. 1–3, pp. 253–285, 2002.
- S. D. Pietra, V. D. Pietra, and J. Lafferty, "Duality and auxiliary functions for Bregman distances," DTIC Document, Fort Belvoir, VA, USA, Tech. Rep. CMU-CS-01-109R, 2001.
- M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1/2, pp. 1–305, 2008.

- [36] M. Seeger and D. P. Wipf, "Variational Bayesian inference techniques," *IEEE Signal Process. Mag.*, vol. 27, no. 6, pp. 81–91, Nov. 2010.
- [37] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.
- [38] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and DC programming," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4686–4698, Dec. 2009.
- [39] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Advances Neural Inf. Process. Syst.*, 2001, pp. 556–562.
- [40] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.
- [41] Z. Lin, R. Liu, and H. Li, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning," *Mach. Learn.*, vol. 99, no. 2, pp. 287–325, 2015.
- [42] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends[®] Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [43] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," Tech. Rep. UILU-ENG-09-2215, UIUC, Oct. 2009.
- [44] A. Ganesh, Z. Lin, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast algorithms for recovering a corrupted low-rank matrix," in *Proc. Inf. Workshop Comput. Advances Multi-Sensor Adaptive Process.*, 2009, pp. 213–216.
- [45] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*, vol. 3. Berlin, Germany: Springer, 2010.
- [46] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM J. Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.



Zhouchen Lin (M'00-SM'08) received the PhD degree in applied mathematics from Peking University, in 2000. Currently, he is a professor in the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University. He is also a chair professor with Northeast Normal University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an area chair of CVPR 2014, ICCV 2015, NIPS 2015, and CVPR 2016, and a senior program committee member of AAAI 2016, IJCAI 2016, and AAAI 2017. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and the *International Journal of Computer Vision*. He is a fellow of the IAPR and a senior member of the IEEE.



Chen Xu received the BEng degree in electronic information engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently working toward the PhD degree in the Key Laboratory of Machine Perception (MOE), School of Electronics Engineering and Computer Science, Peking University, Beijing, China. His current research interests include computer vision, machine learning, and numerical optimization.



Hongbin Zha received the BEng degree in electrical engineering from Hefei University of Technology, Anhui, China, in 1983 and the MS and PhD degrees in electrical engineering from Kyushu University, Fukuoka, Japan, in 1987 and 1990, respectively. After working as a research associate in the Kyushu Institute of Technology, he joined Kyushu University, in 1991 as an associate professor. He was also a visiting professor in the Centre for Vision, Speech and Signal Processing, Surrey University, United Kingdom, in 1999. Since 2000, he has been a professor in the Key Laboratory of Machine Perception (MOE), Peking University, Beijing, China. His research interests include computer vision, digital geometry processing, and robotics. He has published more than 250 technical publications in journals, books and international conference proceedings. He received the Franklin V. Taylor Award from the IEEE Systems, Man and Cybernetics Society in 1999. He is a member of the IEEE Computer Society.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.