2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Robust Bilinear Matrix Recovery by Tensor Low-Rank Representation

Zhao Zhang   Shuicheng Yan   Mingbo Zhao   Fan-Zhang Li

***Abstract*— For low-rank recovery and error correction, Low-Rank Representation (LRR) row-reconstructs given data matrix *X* by seeking a low-rank representation, while Inductive Robust Principal Component Analysis (IRPCA) aims to calculate a low-rank projection to column-reconstruct *X*. But either column or row information of *X* is lost by LRR and IRPCA. In addition, the matrix *X* itself is chosen as the dictionary by LRR, but (grossly) corrupted entries may greatly depress its performance. To solve these issues, we propose a simultaneous low-rank representation and dictionary learning framework termed Tensor LRR (TLRR) for robust bilinear recovery. TLRR reconstructs given matrix *X* along both row and column directions by computing a pair of low-rank matrices alternately from a nuclear norm minimization problem for constructing a low-rank tensor subspace. As a result, TLRR in the optimizations can be regarded as enhanced IRPCA with noises removed by low-rank representation, and can also be considered as enhanced LRR with a clean informative dictionary using a low-rank projection. The comparison with other criteria shows that TLRR exhibits certain advantages, for instance strong generalization power and robustness enhancement to the missing values. Simulations verified the validity of TLRR for recovery.**

***Keywords*— Low-rank representation; tensor representation; bilinear recovery; dictionary learning; error correction**

I. INTRODUCTION

High-dimensional observations can be encountered in various emerging real applications attributed to the rapid development of science and technology, such as face recognition [13], [21], and gene selection, leading to the research topics of recovering low-dimensional structures from high-dimensional data. Note that plenty of real data (such as images, videos and documents) can often be characterized by low-rank subspaces [1][2][3][6], thus investigating the low-rank structures of high-dimensional data have attracted explosively increasing attention in the last years. Representative works dedicated to this topic consist of [1-11], [15], [18], [26-27], [29-39], etc.

Z. Zhang is now with the School of Computer Science and Technology, Soochow University, Suzhou 215006, P. R. China; also with the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (email: cszzhang@gmail.com)

S. C. Yan is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore (email: eleyans@nus.edu.sg)

M. B. Zhao is now with the Department of Electronic Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong SAR (email: mzhao4@cityu.edu.hk)

F. Z. Li is now with the School of Computer Science and Technology, Soochow University, Suzhou 215006, P. R. China; also with the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (email: lfzh@suda.edu.cn)

One most representative low-rank recovery criterion is called *Robust Principal Component Analysis* (RPCA) [3], [8], [9], [16]. For a given observed data matrix $X=[x_1,x_2,...,x_N]\in\mathbb{R}^{n\times N}$ corrupted by certain sparse errors E_0, RPCA is able to recover X_0 ($X=X_0+E_0$) from the following nuclear norm problem:

$$\underset{Y,E}{Min}\ \|Y\|_*+\gamma\|E\|_\ell,\ \text{Subj } X=Y+E\,, \tag{1}$$

where $\|\bullet\|_*$ is the nuclear norm of a matrix, i.e., the sum of the singular values of a matrix, $\|\bullet\|_\ell$ is l^1-norm ($\|\bullet\|_1$) or $l^{2,1}$-norm ($\|\bullet\|_{2,1}$) for characterizing the error term E, and γ is a positive parameter. Note that under much broader conditions, as long as the given observations are corrupted by sufficiently sparse errors, RPCA can exactly recover X_0 from the above convex optimization problem [3]. The minimizer Y^* (with respect to the variable Y) corresponds to the principal components of X and is also the low-rank recovery to X_0. It is also worth noting that RPCA can well address the gross corruptions with large magnitude [3] if only a fraction of entries are corrupted [6]. But RPCA is a transductive algorithm, so it cannot embed new data [6]. Besides, the formulation of RPCA implicitly assumes that the underlying data structures lie in or near a single low-rank subspace. But most real-world data are described using a union of multiple subspaces [1], [2], so the recovery of RPCA may be inaccurate in practice. To address these problems, an effective extension to RPCA, referred to as *Inductive Robust Principal Component Analysis* (IRPCA) [6] was proposed to improve RPCA recently via calculating a low-rank projection $U=[u_1,u_2,...,u_n]\in\mathbb{R}^{n\times n}$ to remove the possible corruptions and recover the original data. Given an observed data matrix X, IRPCA calculates the underlying low-rank projection U and the principal components $Y=[y_1,y_2,...,y_N]$ from the following convex nuclear norm problem:

$$\underset{U,E}{Min}\ \|U\|_*+\gamma\|E\|_\ell,\ \text{Subj } X=Y+E,\ Y=UX\,. \tag{2}$$

After the minimizer U^* is achieved, the original data can be recovered through U^*X (or $X-E^*$). Based on the learnt low-rank projection U^*, the given data can be mapped onto the underlying subspaces and the possible corruptions can also be efficiently removed [6]. But note that IRPCA performs error correction along column direction of given matrix. As a result, row information of data is lost in the IRPCA formulation.

To well cope with mixed data with corrupted observations, another effective low-rank criterion, *Low-Rank Representation* (LRR) [1], [2], was also proposed for subspace recovery and segmentation. For subspace recovery, LRR seeks a low-rank representation $V=[v_1,v_2,...,v_N]\in\mathbb{R}^{N\times N}$ among all candidates that represent all data vectors as the linear combination of bases in a given dictionary D. By setting X itself as the dictionary (i.e., $D=X$), the convex criterion of LRR is given as

$$\underset{V,E}{Min}\|V\|_* + \gamma\|E\|_\ell, \text{ Subj } X = DV + E, D = X \ . \quad (3)$$

After calculating the optimal solution (V^*, E^*), the original data is recovered as $X - E^*$ (or XV^*). Different from IRPCA, LRR recovers or segments given data along row direction, but in contrast column information of the given data matrix is lost in the LRR problem. Note that LRR can also be used to handle the issue of error correction, but LRR is also a transductive method as RPCA, so it also cannot handle new data efficiently. As a result, both LRR and RPCA will be inappropriate for the practical applications requiring fast online computation [6]. It is also noted that, for robust subspace recovery, LRR requires that sufficient noiseless data is available in the dictionary (i.e., only a part of D is corrupted). But note that most real data are contaminated by various errors such as outliers, corruptions and noise, so setting X itself as the dictionary directly may be invalid and may depress the robustness performance of LRR greatly for subspace recovery and segmentation [11], [23].

To enhance the robustness of the low-rank recovery to noises, (grossly) corruptions or missing values in data, we propose to incorporate the concept of tensor representation into the low-rank representation and present a bilinear recovery criterion called *Tensor Low-Rank Representation* (TLRR) for enhanced subspace representation and error correction. Compared with RPCA, IRPCA and LRR, the proposed TLRR exhibits certain attractive advantages. First, to enhance the robustness against noise or corruptions, and to well handle the data with missing values, TLRR aims to reconstructs given data along both row and column directions at the same time by embedding the data onto a low-rank tensor subspace $U \otimes V$ spanned applying two low-rank matrices U and V calculated alternately from a nuclear norm minimization problem. Second, TLRR exhibits a strong generalization capability. Note that the formulation of TLRR seamlessly integrates the low-rank representation and dictionary learning into a unified framework, that is, it can perform simultaneous subspace recovery, error correction and dictionary learning. Specifically, when learning a low-rank projection to construct a clean informative dictionary, TLRR is considered as an "enhanced" version of IRPCA based on the noises and corruptions removed data. Similarly, when learning the low-rank representation for subspace recovery, TLRR is regarded as an "enhanced" version of LRR learning with the trained informative dictionary. Thus, the subspace recovery performance and the robustness against noise, corruptions and missing values can be greatly boosted by our TLRR algorithm, compared with other related criteria.

The paper is outlined as follows. Section II briefly reviews other related works. Section III proposes the TLRR algorithm mathematically. Subsequently, we in Section IV describe the simulation settings and evaluate our algorithm. Finally, the paper is concluded in Section V.

II. Related Work

Recent years have witnessed a lot of efforts and increasing interests on the low-rank recovery in the literature. The most related nuclear norm minimization based recovery criteria to ours are RPCA, IRPCA, LRR and *Latent LRR* (LatLRR) [1], [11]. The principles of IRPCA and LRR are illustrated in the *top left* and *top right* of Figure 1, respectively. Obviously, IRPCA and LRR perform recovery and error correction along either column or row direction of the data matrix, thus row or column information of the data matrix is lost by them.

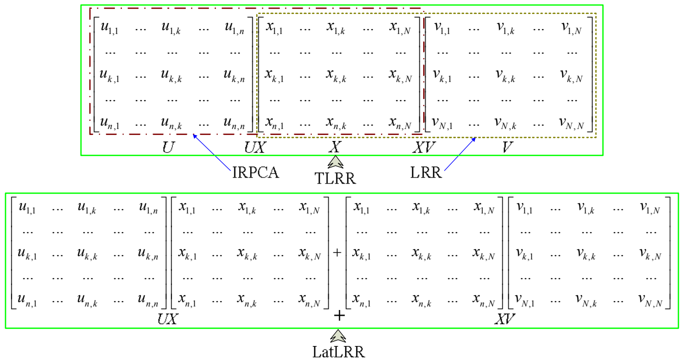

Figure 1: The principles of IRPCA (*top left*), LRR (*top right*), TLRR (*top*) and LatLRR (*bottom*).

Another related recovery criterion to TLRR is LatLRR that reconstructs the data matrix X from two directions as well. To well deal with the issue of insufficient sampling and improve the robustness to noise, LatLRR constructs the dictionary by using both observed and unobserved hidden data, and solves the following convex problem:

$$\underset{V}{Min}\|V\|_*, \text{ Subj } X = DV, D = [X, X_H], \quad (4)$$

where X_H is the hidden data, and the concatenation (along column) of X and X_H is applied as the dictionary D. Finally, the optimization problem of LatLRR is formulated as

$$\underset{U,V,E}{Min}\|U\|_* + \|V\|_* + \lambda\|E\|_\ell, \text{Subj } X = UX + XV + E \quad (5)$$

when corrupted data is included. LatLRR resolves the problem of insufficient sampling and is shown to be more robust than LRR [11], but note that observed and hidden data are sampled from the same collection of low-rank subspaces [11]. Thus, LatLRR may suffer from the same problem as LRR, since one still cannot ensure there are sufficient noiseless data available in $D = [X, X_H]$. The principle of LatLRR is illustrated in the bottom of Figure 1. Intuitively, LatLRR is a combination of LRR and IRPCA. After the solution (U^*, V^*, E^*) is calculated, LatLRR decomposes X into a low-rank XV^*, a low-rank U^*X, and a sparse part E^* fitting noise.

Although column and row information of the data matrix X are reflected in the final reconstructive procedure of LatLRR, i.e., $U^*X + XV^*$, note that the low-rank matrices U and V can be alternately calculated from the following two equivalent convex problems at each iteration:

$$\underset{V,E_U}{Min}\|V\|_* + \lambda\|E_U\|_\ell, \text{Subj } X = XV + E_U, E_U = UX + E\ , \quad 6(a)$$

$$\underset{U,E_V}{Min}\|U\|_* + \lambda\|E_V\|_\ell, \text{Subj } X = UX + E_V, E_V = XV + E\ . \quad 6(b)$$

Note that when solving V for low-rank representation at each iteration, E_U is fixed and the optimization in Eq.6 (a) is equivalent to the LRR formulation by setting the matrix X as the dictionary if E_U is considered as the error matrix E in LRR. Similarly, when solving U for low-rank projection, E_V is fixed and the problem in Eq.6 (b) is equivalent to IRPCA for seeking principal components of X if E_V is regarded as the error matrix E in IRPCA. It is also worth noting that the so-called hidden effects that X_H bring to the problem is still unclear. To boost the robustness to noise and well handle data with missing values, this paper introduces a new mechanism to recover the original data from two directions, as illustrated in the top of Figure 1. Based on this strategy, the proposed TLRR can exhibit certain properties over existing criteria, e.g., strong generalization power and robustness enhancement.

III. Bilinear Low-Rank Recovery Framework

We address the bilinear matrix recovery problem by learning two low-rank factors U and V at the same time to recover the given matrix X from row and column directions in the form of tensor representation (i.e. UXV) of X. In tensor scenarios, X can be regarded as the second-order tensor in tensor space $\mathbb{R}^n \otimes \mathbb{R}^N$ [12], [28]. Denote by $U=(u_1,u_2,...,u_{d_1}) \in \mathbb{R}^{d_1 \times n}$ $(d_1 \le n)$ and $V=(v_1,v_2,...,v_{d_2}) \in \mathbb{R}^{N \times d_2}$ $(d_2 \le N)$ to represent data X, then the tensor product $U \otimes V$ is a subspace of $\mathbb{R}^n \otimes \mathbb{R}^N$ and the projection of X onto the subspace $U \otimes V$ is $UXV \in \mathbb{R}^{d_1 \times d_2}$. This paper considers $d_1 = n$ and $d_2 = N$. We will also elaborate the proposed bilinear recovery criterion can perform simultaneous dictionary learning and low-rank representation.

A. Problem Formulation

For a given data matrix $X \in \mathbb{R}^{n \times N}$ corrupted by certain sparse errors or missing values E_0, we propose to recover the original data X_0 ($X = X_0 + E_0$) along both row and column directions simultaneously. Specifically, we aim at calculating a low-rank representation matrix $V \in \mathbb{R}^{N \times N}$ and a low-rank projection matrix $U \in \mathbb{R}^{n \times n}$ such that $X \approx DV, D = UX$ from the following rank minimization problem:

$$\underset{U,V}{Min}\ \mathrm{rank}(U) + \mathrm{rank}(V) + \gamma \left\| X - \hat{Y} \right\|_\ell, \ \mathrm{Subj}\ \hat{Y} = DV, D = UX, \quad (7)$$

where $\|\bullet\|_*$ is the nuclear norm of a matrix, $X - \hat{Y}$ identifies the sparse errors, $\|\bullet\|_\ell$ (either l^1-norm $\|\bullet\|_1$ or $l^{2,1}$-norm $\|\bullet\|_{2,1}$) is for characterizing the errors E, γ is a positive parameter, and an informative dictionary $D = UX$ is defined by computing a low-rank projection U to project given points onto the underlying subspaces and the entries will be updated at each iteration. Specifically, $\|E\|_1 = \sum_{i,j} |E_{i,j}|$ is designed for handling random corruptions, and $\|E\|_{2,1} = \sum_{j=1}^{n} \sqrt{\sum_{i=1}^{n} [E]_{i,j}}$ can model the sample-specific corruptions and outliers well. If $l^{2,1}$-norm is imposed on E, we can rewrite the above problem as

$$\underset{U,V,E}{Min}\ \mathrm{rank}(U) + \mathrm{rank}(V) + \gamma \|E\|_{2,1}, \ \mathrm{Subj}\ X = \hat{Y} + E, \hat{Y} = DV, D = UX. \quad (8)$$

As a common practice in rank minimization problems [1], [2], [3], [4], we can replace the rank function with the nuclear norm $\|\bullet\|_*$. Then the above problem further becomes

$$\underset{U,V,E}{Min}\ \|U\|_* + \|V\|_* + \gamma \|E\|_{2,1}, \ \mathrm{Subj}\ X = DV + E, D = UX, \quad (9)$$

from which the optimal solution (U^*, V^*, E^*) can be achieved. Therefore, the original data can be reconstructed or recovered as U^*XV^* (or $X - E^*$). Note that both U^* and V^* are required to recover X_0 from two directions, but it is difficult to compute U and V simultaneously. In this paper, we solve U and V alternately and independently, that is, other variables are fixed when optimizing U or V at each iteration. More specifically, one can calculate U and V alternately from the following two equivalent convex problems to Eq.9:

$$\underset{U,E}{Min}\ \|U\|_* + \gamma \|E\|_{2,1}, \ \mathrm{Subj}\ X = U\widetilde{A} + E, \widetilde{A} = XV, \quad 10(a)$$

$$\underset{V,E}{Min}\ \|V\|_* + \gamma \|E\|_{2,1}, \ \mathrm{Subj}\ X = \widetilde{D}V + E, \widetilde{D} = UX, \quad 10(b)$$

where $\widetilde{A} = XV$ denotes the errors corrected and noise removed data by the low-rank representation V, and $\widetilde{D} = UX$ is a clean informative dictionary defined by projecting given data onto the underlying subspaces using the low-rank projection U as IRPCA. In other words, the proposed TLRR framework can perform simultaneous subspace recovery, error correction and dictionary learning. Although the recovery problem in Eq.6 can also learn a low-rank projection, our approach is different from theirs in two main aspects. First, the formulation of Eq.6 is directly built on the LRR criterion by involving the low-rank projection and low-rank representation to estimate for low-rank subspace recovery and projection, while our TLRR criterion is based on a new bilinear model that aims to recover given data matrix from two directions (i.e. row and column) by calculating a pair of low-rank matrix factors to well handle the cases corrupted by noises and missing values in addition to recovering low-rank subspaces. That is, the projection in Eq.6 is updated directly at each iteration, but we instead calculate a low-rank projection U to update the dictionary in TLRR by projecting the given data into the underlying subspaces at each iteration. Second, TLRR is more general than Eq.6, since our TLRR formulation is considered as an enhanced version of LRR in the optimizations.

Note that the above two convex problems can be solved by using various methods, e.g., Augmented Lagrange Multiplier (ALM) [8], [9]. When calculating the low-rank coefficients U to column-reconstruct $\widetilde{A}$, we firstly set U to be an identity matrix, that is, the dictionary is initialized with the given matrix X. After obtaining V from Eq.10 (b), we can update the low-rank projection matrix U from Eq.10 (a) to column-reconstruct $\widetilde{A}$. With U and V obtained alternately, the sparse error matrix E can be obtained from Eq.9. Note that, in the alternating optimizations, the convex problem in Eq.10(a) can be considered as the enhanced IRPCA using errors corrected and noise removed data matrix $\widetilde{A}$ by V. Recall that IRPCA aims at learning a low-rank projection to remove the possible corruptions in given data efficiently through projecting given data onto the underlying subspaces [6]. It is worth noting that the data structures represented by $\widetilde{A} = XV$ will be easier to be projected onto the underlying subspaces than X, because the process of optimizing XV has already corrected the errors in data and segmented points into their respective subspaces by the low-rank representation V. Similarly, the problem in Eq.10 (b) can be considered as enhanced LRR criterion with a clean informative dictionary $\widetilde{D}$ learnt from Eq.10 (a). Therefore, TLRR has the potential to improve the robustness against the noises and missing values, compared with LRR and IRPCA. Based on the relationships among LRR, IRPCA and LatLRR described in Section 2, TLRR can also be considered as the enhanced version of LatLRR.

B. Optimization for Recovering Low-Rank Matrices

In this paper, we use the inexact ALM method [9] to solve our TLRR problem due to efficiency. We first convert the problem in Eq.9 to the following equivalent one:

$$\underset{J,F,U,V,E}{Min}\ \|J\|_* + \|F\|_* + \gamma\|E\|_{2,1},\ \text{Subj}\ X = UXV + E, U = J, V = F\ , \quad (11)$$

from which the variables can be obtained. Note that if l^1-norm is imposed on the sparse error term *E*, that is to solve *E* from $Min_{U,V,E}\ \|U\|_* + \|V\|_* + \gamma\|E\|_1$ with respect to $X = DV + E, D = UX$, the term E_{k+1} can be obtained from $E_{k+1} = \arg\min_E (\gamma/\mu_k)\|E\|_1 + (1/2)\left\|E - \left(\Pi_k^E + (1/\mu_k)Y_1^k\right)\right\|^2$ that can be solved using the shrinkage operator [1], [9]. Motivated by [1],[6], we instead compute U^* and V^* by converting Eq.9 to a simpler problem. With similar argument, the solution U^* to Eq.9 can be factorized into $U^* = \hat{U}^*\left(R^*\right)^{\mathrm{T}}$ with R^* obtained by orthogonalizing the columns of XV. Similarly, the solution V^* to Eq.9 can be factorized into $V^* = Q^*\hat{V}^*$, where Q^* is obtained through orthogonalizing the columns of $X^{\mathrm{T}}U^{\mathrm{T}}$. As a result, the problem in Eq.9 can be equivalent to the following formulation by replacing V and U with $Q^*\hat{V}$ and $\hat{U}\left(R^*\right)^{\mathrm{T}}$ respectively:

$$\begin{aligned}&\underset{J,F,\hat{U},\hat{V},E}{Min}\ \|J\|_* + \|F\|_* + \gamma\|E\|_{2,1} \\ &\text{Subj}\ X = \hat{D}\hat{V} + E, \hat{U} = J, \hat{V} = F, \hat{D} = \hat{U}\mathrm{B}\end{aligned}\ , \quad (12)$$

where $\mathrm{B} = \left(R^*\right)^{\mathrm{T}} XQ^*$. Based on such strategy, the computational burden can be greatly reduced, especially for low-dimensional large-scale problems, that is, N is larger and $n\,(n \le N)$ is relatively smaller, because the above problem is solved with a complexity of $O\left(n^2N + n^3\right)$. Note that when orthogonalizing R^* and Q^*, U and V are defined as the identity matrices. The convergence properties of the inexact ALM are well studied if the number of blocks is at most two and it is able to generally perform well in practice [1], [24]. Since there are more than two blocks in our TLRR, which is similar to IRPCA, LRR and LatLRR, and the objective function is non-convex, it will be hard to prove that the solution of our TLRR converges to the global optimal solution, which is actually a common situation when handling non-convex problems [19], [23], [27]. So, the theoretical study for the convergence analysis of our TLRR method needs to be explored in future. The optimizations of TLRR alternately solve the five blocks at each iteration, which are easily solvable. This paper observes that $\eta = 1.2$ is a good choice for our TLRR. Under this setting, we experimentally observe that TLRR can usually converge with the iteration number *k* within the range of 30~250.

IV. SIMULATION RESULTS AND ANALYSIS

We evaluate the effectiveness of our TLRR algorithm, along with illustrating the comparison results. One benchmark face dataset and one synthetic dataset are involved in this study. The real face database is ORL database [17], and the synthetic dataset is a "Swiss roll" dataset which follows a Swiss-roll distribution. As a common practice, the images of ORL are resized to 32×32 pixels for computational consideration.

A. Baselines and Simulation Settings

In this paper, we mainly examine TLRR for low-rank recovery and error correction. The performance is compared with five most related methods, that is, LRR, LatLRR, RPCA, IRPCA, and *Sparse Representation* (SR) [13], [14], [20], [21]. For fair comparison, l^1-norm or $l^{2,1}$-norm is regularized on the sparse error term *E* of the formulations of SR, RPCA, IRPCA, LRR, LatLRR and TLRR for each simulation.

(a) SR has similar appearance and applications as LRR, e.g., recovery, reconstruction and noise removal. For recovery, SR computes a sparse representation S by solving the following l^1-norm minimization based problem [13], [21]:

$$\underset{S,E}{Min}\ \|S\|_1 + \lambda\|E\|_\ell,\ \text{Subj}\ X = DS + E, D = X, Diag(S) = 0, \quad (13)$$

where SR enforces $Diag(S) = 0$ to avoid the trivial solution $S = I$, and the given matrix *X* is usually set as the dictionary for learning the sparse representations. After the minimizer $S^* = \left[s_1^*, s_2^*, \dots, s_N^*\right] \in \mathbb{R}^{N\times N}$ (with respect to the variable *S*) to the above problem is achieved, the original data can be similarly reconstructed as XS^* (or $X - E^*$), which is analogous to the recovery of LRR, where each column vector s_i^* represents the coefficients for reconstructing the data point x_i and each entry $s_{i,j}^*$ represents the contribution of x_j for reconstructing x_i.

(b) *Parameter Settings*. For the problems of RPCA, IRPCA, LRR, LatLRR and SR, there is a common parameter λ that depends on the actual noise level of datasets to estimate [1]. According to [1], a relatively large λ should be used when the included errors are slight and otherwise one should tune λ to be relatively small. Besides, LPP and NPE need to estimate the neighborhood size $\hat{k}$. In this study, the parameters of each criterion are carefully chosen and the best results over tuned parameters are reported for comparing the performance.

(c) *Evaluation Metrics*. For recovery and error correction, the result of each criterion is evaluated by the reconstruction error $\rho = \left\|\widehat{X_{cl}} - \widehat{X_{co}}\right\|_F / \left\|\widehat{X_{cl}}\right\|_F$, where $\widehat{X_{cl}}$ is the low-rank recovery to the given data matrix which is not corrupted and $\widehat{X_{co}}$ is the recovered result over different percentages of corruptions. We perform all simulations on a PC with Intel (R) Core (TM) i5 CPU 650 @ 3.20 GHz 3.19 GHz 4G.

B. Face Image De-noising via Error Correction

We evaluate the image de-noising capability of the proposed TLRR for handling face images under different levels of pixel corruptions. The recovery performance of our TLRR is mainly compared with those of RPCA, IRPCA, LRR and LatLRR. The images are selected from the ORL database that consists of variation in facial expression (smiling/non smiling), facial details (glasses/no glasses) and poses. In total, the database has 40 persons and consists of 10 images per person. Three faces are selected for the experiments and a data matrix of size 32×96 is created. Note that the gray values of the face images are normalized to [0, 1] for this simulation. To investigate the robustness of various low-rank recovery criteria to corruptions, two settings (i.e., faces are corrupted by random corruptions and sample-specific corruptions) are tested respectively.

Recovery against random pixel corruptions. We firstly test the case that face images are corrupted by random pixel corruptions. In this simulation, we corrupt a percentage of randomly selected pixels from the face images by replacing the gray values with inverted values, i.e., each gray value g is replaced by 1-g. The corrupted pixels are randomly selected from the test faces and the locations are unknown to each method. In this study, we vary the percentage of corrupted pixels from 0 percent to 90 percent, and accordingly increase the values of λ for each method. We apply $\exp(-\rho)$ as a quantitative evaluation criterion of the recovery performance of each algorithm, i.e., the closer $\widehat{X_{cl}}$ and $\widehat{X_{co}}$ are, the bigger the value of $\exp(-\rho)$ is. Figure 2 shows the results of TLRR and its four competitors as a function of the level of pixel corruptions. Figures 2 (b), (c), (d) and (e) mainly evaluate TLRR for recovering the faces with 10 (or 20) percentage of corrupted pixels. Figures 2 (f) and (g) quantitatively evaluate each l^1-norm or $l^{2,1}$-norm based method for error correction. The results are averaged over 15 random pixel selections. From the results, we find that our proposed TLRR works better than other criteria in correcting corruptions, and the performance of TLRR degrades slower than the others with the increasing corruption percents. It is also observed that l^1-norm based criteria are more appropriate choice for recovering the random corruptions than $l^{2,1}$-norm based criteria.

Recovery against sample-specific corruptions. We corrupt a percentage of randomly chosen sample-specific corruptions, i.e., columns of the data matrix. In this study, we add Gaussian noise with zero mean and 0.02 variance to the columns. The corrupted columns are randomly chosen and the locations are also unknown to the users. We also vary the percentage of corrupted columns from 0 percent to 90 percent, and increase the values of λ accordingly for each algorithm. Figure 3 illustrates the result of each method as a function of the level of column corruptions. Figures 3 (b), (c), (d) and (e) examine our presented TLRR for recovering the face images with 20 (or 50) percentage of column corruptions. Illustrations show that our TLRR is able to effectively detect the corruptions and correct them. Figures 3 (f) and (g) illustrate the quantitative evaluation of error correction for l^1-norm or $l^{2,1}$-norm based criterion. We average the results over 15 random column selections. Similar findings can be found here, that is, TLRR can outperform other methods in identifying and correcting the corruptions in most cases. It is also observed that the $l^{2,1}$-norm based error terms are able to well model the sample-specific corruptions than l^1-norm based criteria.

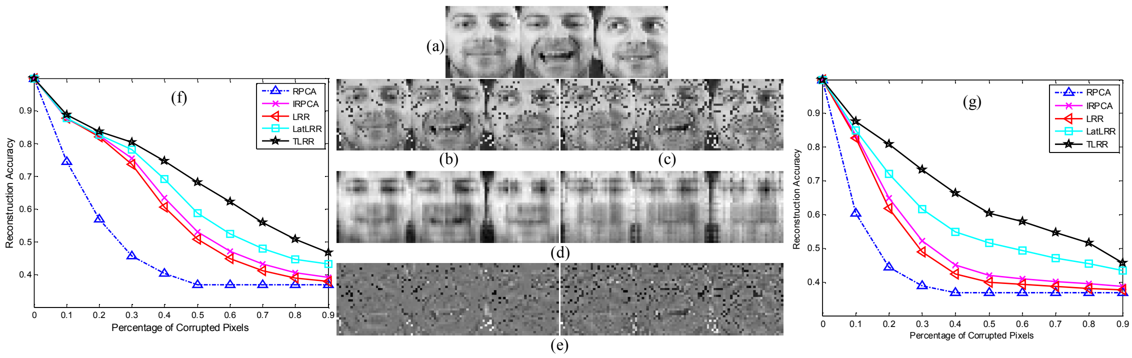

Figure 2: Recovery under random corruptions. (a) Original face images from ORL; (b) 10 percent of pixels are corrupted; (c) 20 percent of pixels are corrupted; (d) Recovered faces by l^1-norm based TLRR; (e) Estimated sparse errors; (f) Reconstruction accuracies across corrupted percents for each l^1-norm based method; (g) Reconstruction accuracies across corrupted percents for each $l^{2,1}$-norm based criterion.

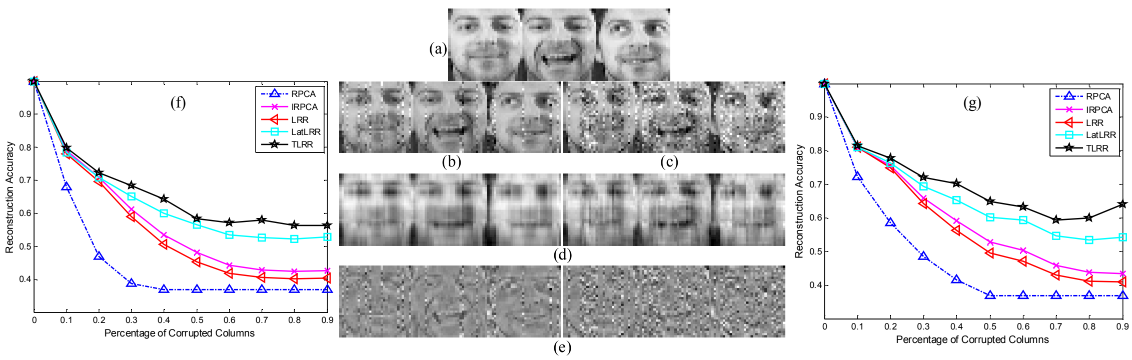

Figure 3: Reconstruction under sample-specific corruptions. (a) Original faces; (b) 20 percent of columns are corrupted; (c) 50 percent of columns are corrupted; (d) Reconstructed faces by the $l^{2,1}$-norm based TLRR; (e) Estimated sparse errors; (f) Reconstruction accuracies across corrupted percents for each l^1-norm based method; (g) Reconstruction accuracies across corrupted percents for each $l^{2,1}$-norm based criterion.

C. Error Correction on Synthetic Data

In this simulation, we address another experiment to evaluate the proposed TLRR method for error correction in noisy case using the synthetic "Swiss roll" dataset. The sampled dataset X contains 2000 column vectors totally and each column vector corresponding to a sample point in a three-dimensional space. The subspace recovery result of our proposed TLRR method is compared with those of IRPCA, LRR, SR and LatLRR. We illustrate the recovery result of each approach in Figure 4. In our experiments, we add Gaussian noise (with zero mean and variance equaling to two) to the x-coordinates of 10 percent of the data vectors (denoted by square symbol) and also add the noise with the same density to the y-coordinates of another 10 percent of vectors (denoted by circle symbol), respectively. The noised "Swiss roll" dataset is illustrated in Figure 4(a). Observing from the recovered data in Figure 4, we see clearly that our TLRR algorithm outperform other compared criteria, which is because of its capability of embedding the given data matrix X into a low-rank tensor representation subspace onto which the included errors can be automatically and effectively corrected by the proposed TLRR criterion from both row and column directions at the same time.

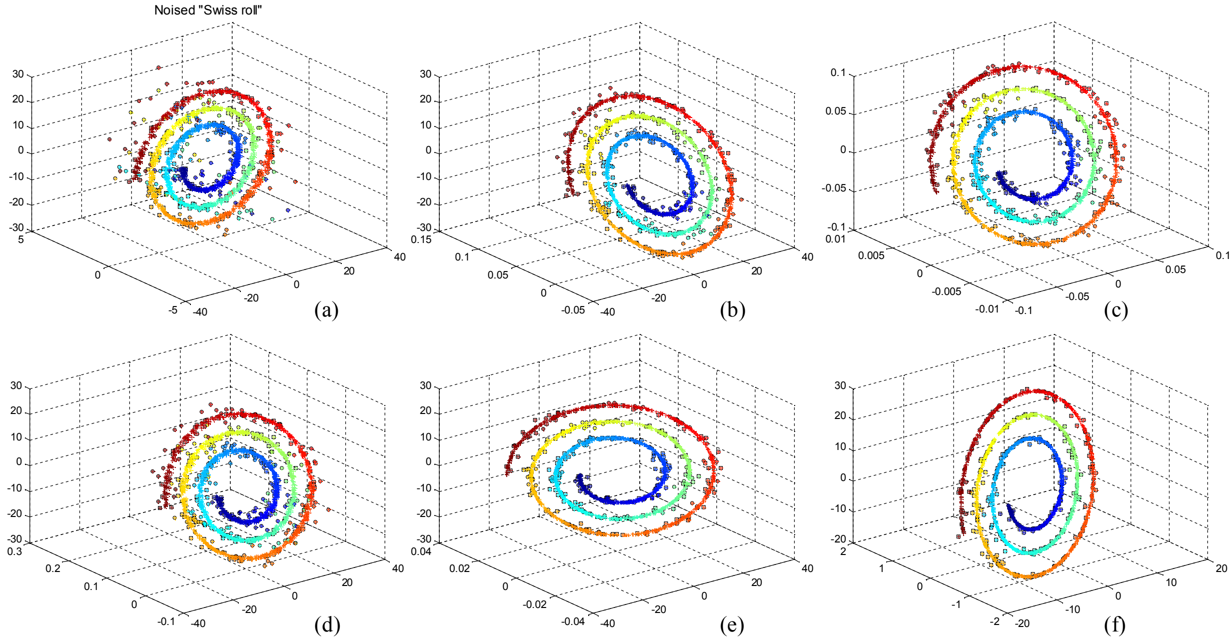

Figure 4: Corrected result of each algorithm on "Swiss roll": (a) Noised dataset; (b) Recovery result (U^*X) of IRPCA; (c) Recovery result (XV^*) of LRR; (d) Recovery result (XS^*) of SR; (e) Recovery result ($U^*X + XV^*$) of LatLRR; (f) Recovery result (U^*XV^*) of our TLRR.

V. Concluding Remarks

In this paper, we have proposed a bilinear subspace recovery framework called TLRR for effective error correction and data repairing. An attractive property of our TLRR criterion is its formulation seamlessly integrates low-rank representation and dictionary learning into a unified framework, which provides us with a new mechanism for recovering low-rank subspaces and simultaneously learning an informative dictionary from a nuclear norm problem. Specifically, TLRR proceeds low-rank recovery through enhancing the robustness against noise and missing values by simultaneously considering column and row information of the given data matrix, thus the shortcomings of LRR and IRPCA can be effectively overcome. In addition, we mathematically elaborate that our TLRR can be regarded as enhanced versions of other existing criteria.

Image recovery and visualization simulations have verified the effectiveness of our TLRR in representing real images and boosting the robustness against to and corruptions. But there are certain future work that still requires to be explored. First, the parameter λ for the error term controls the performance of virtually all low-rank recovery criteria, including our proposed TLRR. In this study, we experimentally observe that TLRR is capable of working well under a wider range of the parameter configurations, but it still remains unclear how to optimally determine the values of λ theoretically for data contaminated by various errors. Second, to date it is still challenging to strictly prove the convergence of inexact ALM based recovery criteria, including IRPCA, LRR, LatLRR and our proposed TLRR, which include more than two blocks to optimize [1] [24]. In this paper, we observe from the experimental results that TLRR can converge with satisfactory iteration numbers, but the theoretical convergence proofs for the presented TLRR are still worth investigating in future.

VI. Acknowledgements

This present work is partially supported by the Major Program of National Natural Science Foundation of China (Grant No. 61033013), and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding

Initiative and administered by the IDM Programme Office.

REFERENCES

[1] G.C Liu, Z.C Lin, S.C Yan, J. Sun, Y. Yu, and Y. Ma, "Robust Recovery of Subspace Structures by Low-Rank Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.35, no.1, pp.171-184, 2013.

[2] G.C. Liu, Z.C. Lin, and Y. Yu, "Robust Subspace Segmentation by Low-Rank Representation," In: *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[3] E. Candes, X.D. Li, Y. Ma, and J. Wright, "Robust Principal Component Analysis?" *Journal of the ACM*, vol.58, no.3, pp.1-37, 2011.

[4] E. Candes, and P. Yaniv, "Matrix completion with noise," In: *Proceedings of the IEEE*, vol.97, no.6, pp., 925-936, 2009.

[5] E. Candµes, and R. Benjamin, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol.9, no.6, pp.717-772, 2009.

[6] B.K. Bao, G.C. Liu, C.S. Xu, and S.C. Yan, "Inductive Robust Principal Component Analysis," *IEEE Trans. on Image Processing (TIP)*, vol.21, no.8, pp.3794-3800, 2012.

[7] Z. Zhang, S. C. Yan, and M. B. Zhao, "Robust Image Representation and Decomposition by Laplacian Regularized Latent Low-Rank Representation," In: *Proceedings of the IEEE International Joint Conference on Neural Networks* (IJCNN), Dallas, Texas, 2013.

[8]Z.C. Lin, A. Ganesh, J. Wright, L.Q. Wu, M.M. Chen, and Y Ma, "Fast Convex Optimization Algorithms for Exact Recovery of a Corrupted Low-Rank Matrix," *UIUC Technical Report UILU-ENG-09-2214*, 2009.

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

[10] Z. Zhang, C. L. Liu, and M. B. Zhao, "Handwriting Representation and Recognition through a Sparse Projection and Low-Rank Recovery Framework," In: *Proceedings of the IEEE International Joint Conference on Neural Networks* (IJCNN), Dallas, Texas, 2013.

[11] G.C. Liu, S.C. Yan, "Latent Low-Rank Representation for subspace segmentation and feature extraction," In: *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[12] Z. Zhang, T. W. S. Chow, "Tensor Locally Linear Discriminative Analysis," *IEEE Signal Processing Letter,* 18(11): 643-646, 2011.

[13] J. Wright, A. Yang, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol.31, no.2, pp.210-227, 2009.

[14] Z. Zhang, M.B. Zhao, and T. W. S. Chow, "Binary- and Multi-Class Group Sparse Canonical Correlation Analysis for Feature Extraction and Classification," *IEEE Trans. Knowledge and Data Engineering*, vol.25, no.10, pp.2192-2205, 2013.

[15] A. Ganesh, J. Wright, X.D. Li, E. Candes, and Y. Ma, "Dense Error Correction for Low-Rank Matrices via Principal Component Pursuit," In: *Proceedings of the International Symposium on Information Theory*, 2010.

[16] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," In: *Advances* in *Neural Information Processing Systems (NIPS)*, 2010.

[17] F. Samaria, and A. Harter, "Parameterisation of a Stochastic Model for Human Face Identification," In: *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota FL, 1994.

[18] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," In: *Advances in Neural Information Processing Systems* (NIPS), pp.612-620, 2011.

[19] A. Del Bue, J. Xavier, L. Agapito, M. Paladini, "Bilinear Modeling via Augmented Lagrange Multipliers (BALM)," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, pp. 1496-1508, 2012.

[20] Z. Zhang, S.C. Yan, and M.B. Zhao, "Pairwise Sparsity Preserving Embedding for Unsupervised Subspace Learning and Classification," *IEEE Trans. on Image Processing*, vol.22, iss.12, pp.4640-4651, 2013.

[21] L. S. Qiao, S. C. Chen, X. Y. Tan, "Sparsity Preserving Projections with Applications to Face Recognition," *Pattern Recognitio*n, vol.43, no.1, pp.331-341, 2010.

[22] A. M. Martinez, A. C. Kak, "PCA versus LDA," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.23, no.2, pp.228-233, 2001.

[23] S. M. Wei, and Z. C. Lin, "Analysis and Improvement of Low Rank Representation for Subspace segmentation," *CoRR abs/1107.1561*, 2011.

[24] Y. Zhang, "Recent advances in alternating direction methods: Practice and theory," *Tutorial*, 2010.

[25] A. Beck, and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol.2, pp. 183-202, 2009.

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

[27] P. Favaro, R. Vidal, and A. Ravichandran, "A Closed Form Solution to Robust Subspace Estimation and Clustering, In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR)," pp. 1801–1807, 2011.

[28] X. F. He, D. Cai, and P. Niyogi, "Tensor Subspace Analysis," In: *Advances in Neural Information Processing Systems 18*, Vancouver, Canada, 2005.

[29] Y. Ni, J. Sun, X. Yuan, S. Yan, L. Cheong, "Robust Low-Rank Subspace Segmentation with Semidefinite Guarantees," In: *Proceedings of the IEEE International Conference on Data Mining*, pp.1179-1188, 2010.

[30] R. S. Liu, Z. C. Lin, F. De la Torre, and Z.X. Su, "Fixed-Rank Representation for Unsupervised Visual Learning," *CoRR abs/1203.2210*, 2012.

[31] L. S. Zhuang, H. Y. Gao, Z. C. Lin, Y. Ma, X. Zhang, and N. H. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp.2328-2335, 2012.

[32] G. C. Liu, and S. C. Yan, "Active subspace: toward scalable low-rank learning," *Neural Computation*, vol.24, no.12, pp.3371-3394, 2012.

[33] Y. Mu, J. Dong, X. Yuan, and S. C. Yan, "Accelerated Low-Rank Visual Recoveryby Random Projection," In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), pp.2609-2616, 2011.

[34] T. Zhou, and D. Tao, "GoDec: Randomized Low-Rank & Sparse Matrix Decomposition in Noisy Case," In: *Proceedings of the International Conference on Machine Learning* , pp.33-40, 2011.

[35] C. Chen, B. He, and X. Yuan, "Matrix completion via an alternating direction method," *IMA Journal of Numerical Analysis*, vol.32, no.1, pp.227-245, 2012.

[36] Y. Y. Liu, L. C. Jiao, F. H. Shang, F. Yin, and F. Liu, "An efficient matrix bi-factorization alternative optimization method for low-rank matrix recovery and completion," *Neural Networks,* vol.48, pp.8-18, 2013.

[37] Y. Y. Liu, L. C. Jiao, and F. H. Shang, "A fast tri-factorization method for low-rank matrix recovery and completion," *Pattern Recognition,* vol.46, no.1, pp.163-173, 2013.

[38] Y. Y. Liu, L. C. Jiao, and F. H. Shang, "An efficient matrix factorization based low-rank representation for subspace clustering" *Pattern Recognition*, vol.46, no.1, pp.284-292, 2013.

[39] X. Ren, and Z. C. Lin, "Linearized Alternating Direction Method with Adaptive Penalty and Warm Starts for Fast Solving Transform Invariant Low-Rank Textures," *International Journal of Computer Vision*, vol.104, no.1, pp.1-14, 2013.