Robust Tensor Factorization Using R_1 Norm

Heng Huang
Computer Science and Engineering
University of Texas at Arlington
heng@uta.edu

Chris Ding
Computer Science and Engineering
University of Texas at Arlington
chqding@uta.edu

Abstract

Over the years, many tensor based algorithms, e.g. two dimensional principle component analysis (2DPCA), two dimensional singular value decomposition (2DSVD), high order SVD, have been proposed for the study of high dimensional data in a large variety of computer vision applications. An intrinsic limitation of previous tensor reduction methods is the sensitivity to the presence of outliers, because they minimize the sum of squares errors (L_2 norm). In this paper, we propose a novel robust tensor factorization method using R_1 norm for error accumulation function using robust covariance matrices, allowing the method to be efficiently implemented instead of resorting to quadratic programming software packages as in other L_1 norm approaches. Experimental results on face representation and reconstruction show that our new robust tensor factorization method can effectively handle outliers compared to previous tensor based PCA methods.

1. Introduction

Principal component analysis (PCA) is a widely used method for dimension reduction and PCA based techniques have been successfully applied into many computer vision application, *e.g.* image reconstruction, object recognition, tracking, detection, appearance, and motion. In face representation and recognition, PCA was first used by Sirovich and kirby to represent human facial images [7]. Later, Turk and Pentland [9] proposed the well-known PCA based eigenface method for face recognition. Because in classical PCA the image matrix is mapped into a one dimensional vector, the spatial correlation within each image are not fully utilized. After realizing this intrinsic problem, many researchers in computer vision and pattern recognition areas have begun to emphasize the image as matrix or tensor to improve the performance of subspace dimension reduction. Based on PCA, some image-based subspace analysis approaches have been developed. Shashua and Levine [12] used rank-1 (one of the three tensor decompositions described by Tucker in [13]) matrices to represent a set of images. Recently, Yang *et al.* [15] proposed a two dimensional PCA (2DPCA) in which image covariance matrices are constructed directly using original image matrices and one-side low-rank approximation is applied. Ye *et al.* [16] proposed a method called Generalized Low Rank Approximation of Matrices (GLRAM) that is a two-side low-rank approximation method and projects the original data onto a two dimensional space L and R such that the project has the largest variance among all two dimensional space. Ding and Ye proposed a non-iterative algorithm in [2] called as two dimensional singular value decomposition (2DSVD). Although several other tensor based PCA research results have been proposed, Inoue and Urahama have shown the equivalence of them in paper [5].

It is commonly known that traditional PCA and novel tensor based PCA (like 2DPCA and 2DSVD) minimize the sum of squared errors, which is prone to the presence of outliers, because large errors squared dominate the sum. In the vision community, some previous attempts to make PCA robust have treated entire images as outliers [14] and several other methods focus on intra-sample outliers [8]. Ke and Kanade [6] proposed another approach using L_1 norm or the least absolute deviance that is less sensitive to outliers compared to L_2 norm (the Euclidean metric). This method need solve the optimization problem through the quadratic programming that is computationally expensive. In order to improve the computational efficiency, Ding *et al.* [3] used the rotational invariant L_1 norm (R_1 norm) as the objective functions of PCA. All those methods were working on robustness improvement of traditional PCA. An assembled matrix distance metric [17] was proposed to measure the distances between features' matrices using L_p norm and work with 2DPCA together to improve the face recognition rate. Since his method didn't apply the new distance function into the objective function of 2DPCA and still used L_2 norm for tensor factorization optimization, the robustness of 2DPCA is still the same. Park and Savvides also proposed a tensor method [11] to improve the robustness of face recognition, but they also used L_2 distance. To our knowledge, so far there is no robust method to improve the performance of tensor based PCA.

In this paper, we propose a novel robust tensor factorization approach using R_1 norm. By projecting the tensor data (2D images) onto the (K_1, K_2)-dimensional space, we expect to obtain optimal approximation of the original tensor with less sensitivity to the outliers. In order to demonstrate the efficiency and effective of the proposed method, experiments are carried out using the well known ORL face database and YALE face database. Experimental results show that, our method can effectively handle outliers compared to previous tensor based PCA methods.

Our contributions are as follows:
1) We propose a novel robust tensor factorization method in which the principal components of subspace are the principle eigenvectors of a robust covariance matrix (re-weighted to soften outliers) with two-side low-rank approximation.
2) We also prove the solutions are rotational invariant. A subspace is not uniquely determined up to an orthogonal transformations
3) An efficient robust tensor factorization algorithm is given out in section 3 and its iterative steps are similar to previous 2DSVD related methods, but we use weighted covariance matrices.
4) There is no extra time and space complexities compared to the general tensor based PCA methods.

2. Subspace analysis

To illustrate the concept, in this section we introduce the relevant preliminary material concerning robust PCA (robust factorization of a 2D tensor consisting of a set of 1D vectors data) and tensor based PCA.

2.1. Rotational invariant L_1 norm

Given n data points in d-dimensional space, we denote them as $X = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$. The Frobenius and L_1 norms are defined as

$$||X||_{_F} = \Big(\sum_{i=1}^{n}\sum_{j=1}^{d} x_{ji}^2\Big)^{1/2},\ ||X||_{L_1} = \sum_{i=1}^{n}\sum_{j=1}^{d} |x_{ji}|. \quad (1)$$

In matrix $X = (x_{ji})$, index i sum over data points, $i = 1, \cdots, n$ and index j sum over spatial dimensions, $j = 1, \cdots, d$. The R_1 norm is defined as [3]

$$||X||_{R_1} = \sum_{i=1}^{n}\Big(\sum_{j=1}^{d} x_{ji}^2\Big)^{1/2}, \quad (2)$$

For some matrices $X_{d\times n}$, the properties of R_1 norm are:

1. Triangle inequality: $||X+Y||_{R_1} \le ||X||_{R_1} + ||Y||_{R_1}$
2. Rotational invariance: $||RX||_{R_1} = ||X||_{R_1}$ for any orthogonal R (a rotation transformation).

Rotational invariance is a fundamental property of Euclidean space with L_2 norm. It has been emphasized in the context of learning algorithms [10]. When we apply PCA into reducing the dimension of high-dimensional data, we hope to project data into a low-dimensional subspace which reduces the noise at same time. A subspace is not uniquely determined up to an orthogonal transformations. Therefore, we prefer to model data with distributions that satisfy rotational invariance.

In standard PCA, the subspace can be estimated by the following matrix factorization:

$$X_{d\times n} = U_{d\times k} V_{k\times n}. \quad (3)$$

where $U = (\mathbf{u}_1, \cdots, \mathbf{u}_k) \in \Re^{d\times k}$ and $V = (\mathbf{v}_1, \cdots, \mathbf{v}_n) \in \Re^{k\times n}$. The standard PCA is formulated as:

$$\min_{U,V} J = ||X - UV||_{_F}^2 = \sum_{i=1}^{n} ||\mathbf{x}_i - U\mathbf{v}_i||^2. \quad (4)$$

The robust version of PCA, R_1-PCA, uses R_1 norm and the cost function is:

$$\min_{U,V} J_{R_1} = ||X - UV||_{R_1} = \sum_{i=1}^{n} ||\mathbf{x}_i - U\mathbf{v}_i||. \quad (5)$$

L_1-PCA has the cost function:

$$\min_{U,V} J_{L_1} = ||X - UV||_{L_1} = \sum_{i=1}^{n} ||\mathbf{x}_i - U\mathbf{v}_i||_{L_1}, \quad (6)$$

This requires quadratic programming which is computationally expensive. It also has a drawback of a skewed isosurface [3].

Both standard PCA of Eq. (4) and R_1-PCA of Eq. (5) can be simplified as:

$$\min_{U^TU=I} J_{PCA} = \sum_{i=1}^{n} (\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T UU^T\mathbf{x}_i). \quad (7)$$

$$\min_{U^TU=I} J_{R_1} = \sum_{i=1}^{n} \sqrt{\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T UU^T\mathbf{x}_i}. \quad (8)$$

From two above equations, we know $J_{PCA}(U)$ and $J_{R_1}(U)$ are convex functions of UU^T. Thus, both PCA and R_1 PCA have a unique global optimal solution. Although UU^T is unique, U is unique up to an orthogonal transformation R.

2.2. Two dimensional PCA and 2DSVD

In 2D approach [15], the image matrix does not need to be previously transformed into a vector, thus a set of N sample images is represented as $\{X_1, X_2, \cdots, X_N\}$. We assume the sample images have been centerized with the mean image $\bar{X} = \sum_{i=1}^{N} X_i = 0$. 2DPCA uses all sample images to construct the image covariance matrix G as:

$$G = \sum_{i=1}^{N} X_i^T X_i, \tag{9}$$

where X_i is the i-th sample image with size of $r \times c$. Only one dimensional column by column correlation is considered.

Ye [16] and later Ding and Ye [2] considered the problem of computing low rank approximations of matrices by minimizing the approximation error and gave out an iterative algorithm 2DSVD. Formally, they consider the following optimization problem:

$$\min_{L,R,\{M_i\}} J_r(L, \{M_i\}, R) = \sum_{i=1}^{n} ||X_i - LM_iR^T||_F^2. \tag{10}$$

where $L \in \Re^{r \times k_1}$, $R \in \Re^{c \times k_2}$, and $M_i \in \Re^{k_1 \times k_2}$. Let's define the row-row and column-column covariance matrices as:

$$F = \sum_{i=1}^{N} X_i R R^T X_i^T, \tag{11}$$

$$G = \sum_{i=1}^{N} X_i^T L L^T X_i. \tag{12}$$

The 2DPCA of Yang *et al.* [15] is a special case of Eq. (10) by setting $L = I$ (*i.e.* ignoring L).

The 2DSVD solutions are calculated as following steps:
(1) Initialize $RR^T = I_{c \times c}$ and an error threshold;
(2) Form a new F according to Eq. (11) and compute the first k_1 eigenvectors $\mathbf{l}_1, \cdots, \mathbf{l}_{k_1}$ to form the updated $L = [\mathbf{l}_1, \cdots, \mathbf{l}_{k_1}]$;
(3) Form a new G according to Eq. (12) and compute the first k_2 eigenvectors $\mathbf{r}_1, \cdots, \mathbf{r}_{k_2}$ to form the updated $R = [\mathbf{r}_1, \cdots, \mathbf{r}_{k_2}]$;
(4) Go back to (2) until the J_r of Eq. (10) is less than the error threshold.

We will formulate a robust version of Eq. (10). From there, a robust version of 2DPCA can also be constructed.

3. Robust tensor factorization

In this section, we formulate the subspace estimation as an R_1 norm problem, and then present a concrete robust tensor factorization algorithm to minimizing the R_1 norm.

3.1. R_1 norm based tensor factorization

In order to improve the robustness of tensor based PCA, we use R_1 norm to replace the L_2 norm as cost function. In 2DSVD, the optimization function is the sum of squared errors in Eq. (10). Similar to Eq. (5), the cost function using the R_1 norm is defined as:

$$\min_{L,R,\{M_i\}} J_r(L, \{M_i\}, R) = \sum_{i=1}^{n} \sqrt{||X_i - LM_iR^T||^2}. \tag{13}$$

A L_1 norm based cost function is:

$$\min_{L,R,\{M_i\}} J_r(L, \{M_i\}, R) = \sum_{i=1}^{n} ||X_i - LM_iR^T||_{L_1}. \tag{14}$$

The L_1 version is computationally expensive and will not be discussed further. The right side of Eq. (13) can be calculated as:

$$\begin{aligned}&\sum_{i=1}^{n} \sqrt{||X_i - LM_iR^T||^2} \\ &= \sum_{i=1}^{n} \sqrt{\mathbf{Tr}(X_i^T X_i - 2LM_iR^T X_i^T + M_i^T M_i)}\end{aligned} \tag{15}$$

First, we solve the optimization problem on M_i. Taking $\partial J_r / \partial M_i = 0$, we have

$$\begin{aligned}\frac{\partial J_r}{\partial M_i} &= \frac{-L^T X_i R + M_i}{\sqrt{\mathbf{Tr}(X_i^T X_i - 2LM_iR^T X_i^T + M_i^T M_i)}} \\ &= 0\end{aligned} \tag{16}$$

Thus, we obtain

$$M_i = L^T X_i R.$$

Using it, Eq. (13) is written as:

$$\begin{aligned}&\min_{L,R} J_r(L, R) \\ &= \sum_{i=1}^{n} \sqrt{\mathbf{Tr}(X_i^T X_i - 2X_i^T L L^T X_i R R^T)}.\end{aligned} \tag{17}$$

Now we solve for L and R. To enforce, the orthogonality: $L^T L = I$ and $R^T R = I$, we follow the standard theory of constrained optimization and introduce the Lagrangian function as follows:

$$\min_{L,R} \mathcal{L} = J_r + \mathbf{Tr}\ \Sigma(L^T L - I) + \mathbf{Tr}\ \Omega(R^T R - I) \tag{18}$$

where the Lagrangian multiples Σ and Ω are symmetric matrices. The KKT condition for optimal solution specifies that the gradient of $\mathcal{L}$ must be zeros. We obtain:

$$\frac{\partial \mathcal{L}}{\partial L} = -2F_r L + 2L\Sigma = 0, \tag{19}$$

where

$$F_r = \sum_i \frac{X_i R R^T X_i^T}{\sqrt{\mathbf{Tr}(X_i^T X_i - 2X_i^T L L^T X_i R R^T)}} \tag{20}$$

and

$$\frac{\partial \mathcal{L}}{\partial R} = -2G_r R + 2R\Omega = 0, \tag{21}$$

where

$$G_r = \sum_i \frac{X_i^T L L^T X_i}{\sqrt{\mathbf{Tr}(X_i^T X_i - 2X_i^T L L^T X_i R R^T)}}. \tag{22}$$

(F_r, G_r) of Eqs. (20,22) are the robust (weighted) version of (F, G) in Eqs. (11,12). We call them as robust covariance matrices.

3.2. Computational algorithm

In paper [3], the problem is solved by the subspace iteration method. Here, we use solve them in a new method by eigen decompositions.

From Eq. 19, we get $F_r L = L\Sigma$ and calculate the first k_1 eigenvectors of F_r with $F_r \mathbf{l}_k = \sigma_k \mathbf{l}_k$ ($k = 1, \cdots, k_1$). L can be formed as $[\mathbf{l}_1, \cdots, \mathbf{l}_{k_1}]$. From Eq. 21, we have $G_r R = R\Omega$ and calculate the first k_2 eigenvectors of G_r with $G_r \mathbf{r}_k = \omega_k \mathbf{r}_k$ ($k = 1, \cdots, k_2$). R can be formed as $[\mathbf{r}_1, \cdots, \mathbf{r}_{k_2}]$.

Theorem 1. The solutions of robust 3D tensor factorization of Eq.(13) are the principal eigenvectors of the rotational invariant covariance matrices (a weighted version of the covariance matrices):

$$F_r = \sum_i w_i X_i R R^T X_i^T, \; G_r = \sum_i w_i X_i^T L L^T X_i \tag{23}$$

where

$$w_i = \frac{1}{\sqrt{\mathbf{Tr}(X_i^T X_i - 2X_i^T L L^T X_i R R^T)}}. \tag{24}$$

In practice, we use the popular Huber's M-estimator:

$$\rho(s) = \begin{cases} s^2 & \text{if} \quad |s| \le c \\ 2c|s| - c^2 & \text{if} \quad |s| > c \end{cases} \tag{25}$$

as the generalized L_1-norm: $\rho(s)$ reduces to L_1 when the cutoff $c \rightarrow 0$. We minimize $J_{r\text{H}} = \sum_{i=1}^n \rho(\sqrt{||X_i - LM_iR^T||^2})$ instead of Eq.(13). The solution is the same as in Theorem 1, with the weights replaced by

$$w_i = \begin{cases} 1 & \text{if} \;\; r_i < c \\ c/r_i & \text{if} \;\; r_i \ge c \end{cases} \tag{26}$$

where $r_i = \sqrt{\mathbf{Tr}(X_i^T X_i - 2X_i^T L L^T X_i R R^T)}$.

We summarize the robust tensor factorization algorithm in Table 1. From 2DSVD solution algorithm in section 2.2, we know it is an iterative algorithm. The first step in our algorithm is the same of the first step of 2DSVD. Starting from 2nd step, our algorithm uses the weighted (see Eq.(26)) form of the same matrices in 2DSVD, thus has the same computational complexity. It is possible that our algorithm may needs slightly more iterations than 2DSVD.

```
1) Compute the first step of 2DSVD to get $L$ and $R$;
2) Calculate residue errors $\{r_i\}$.
   Set cutoff = median of $\{r_i\}$.
3) Using current $L, R$ to compute $F_r$ and its eigenvectors.
   This gives the new $L$.
4) Update $R$. Using current $L, R$ compute $G_r$ and its
   eigenvectors. This gives new $R$.
Repeat 3) and 4) until convergence,
   i.e. $J$ is no longer decrease.
```

Table 1. Robust tensor factorization algorithm.

3.3. 2D rotational invariance

We introduce a new concept of rotational invariance property of tensor factorization. We will show both GLRAM/2DSVD of Eq. 10 and the robust version of Eq.(13) have the nice rotational invariance property. We define the rotational transformation of tensor factorization as:

$$X_i \leftarrow R_1 X_i R_2^T, \; L \leftarrow R_1 L, \; R \leftarrow R_2 R \tag{27}$$

where R_1, R_2 are orthonormal: $R_1^T R_1 = I_r, R_2^T R_2 = I_c$. $R_1 \in \Re^{r \times r}$ performs the rotational transform on the rows of matrix $X \in \Re^{r \times c}$ and $R_2 \in \Re^{c \times c}$ makes the rotational transform on the columns of matrix X. If the projection results (*e.g.* M_i) remain unchanged under the rotation, we say they are invariant. We note pure L_1-PCA of Eq. (6) and L_1-2DSVD of Eq. (14) don't have this property.

Theorem 2. Both GLRAM/2DSVD and robust tensor factorization are rotational invariant.

Proof: For each term in Eq. (10,13), we have:

$$\begin{aligned} & ||X_i - LM_iR^T|| \\ &= ||(R_1^T R_1)(X_i - LM_iR^T)(R_2^T R_2)|| \\ &= ||R_1(X_i - LM_iR^T)R_2^T|| \\ &= ||R_1 X_i R_2^T - (R_1 L) M_i (R_2 R)^T)|| \end{aligned} \tag{28}$$

Therefore, M_i remains unchanged in the rotational transformation.

3.4. Space and time complexities

In Table 2, we summarize the storage and matrix sizes for which we need to compute k largest eigenvectors (suppose we use the same number of largest eigenvectors on both F_r and G_r, $K_1 = K_2 = k$). For a matrix of size $a \times b$, the computational complexity is $O(kab)$. Our robust tensor factorization approach has the same time and storage complexity as 2DSVD. In our method, The robustness is improved without introducing any extra computation.

method	storage	matrix size
PCA	$rck + nk$	$rc \times n$
2DSVD	$rk + nks + sc$	$r \times r$ or $c \times c$
Our method	$rk + nks + sc$	$r \times r$ or $c \times c$

Table 2. Storage and matrix sizes, for n images of size $r \times c$.

4. Robust tensor factorization for higher order tensors (D-1 decomposition)

We can extend our robust tensor factorization from 3D tensor to higher order tensors. Given a D-dimensional tensor, we introduce D-1 subspaces to form a D-1 decomposition [1]. For 2D tensor, D-1 decomposition is PCA. For 3D tensor, D-1 decomposition is GLRAM/2DSVD. Their robust version are discussed in §2.1 and §3.1. Here, we give an example for 4D case. Given a set of 3D tensors $[Y_1, \cdots, Y_n]$, the robust version of D-1 tensor factorization using R_1-norm is:

$$\begin{aligned} \min_{U,V,W,M} J &= \sum_{\ell=1}^{n} ||Y_\ell - U \otimes_1 V \otimes_2 W \otimes_3 M_\ell||_{R_1}, \\ \text{s.t.} \quad & V^T V = I,\ U^T U = I,\ W^T W = I \end{aligned} \tag{29}$$

At first, let's define F, G, H are covariance matrices:

$$\begin{aligned} F_{ii'} &= \sum_\ell w_\ell \sum_{jj'kk'} X^{(\ell)}_{ijk} X^{(\ell)}_{i'j'k'} (VV^T)_{jj'} (WW^T)_{kk'} \\ G_{jj'} &= \sum_\ell w_\ell \sum_{ii'kk'} X^{(\ell)}_{ijk} X^{(\ell)}_{i'j'k'} (UU^T)_{ii'} (WW^T)_{kk'} \\ H_{kk'} &= \sum_\ell w_\ell \sum_{ii'jj'} X^{(\ell)}_{ijk} X^{(\ell)}_{i'j'k'} (UU^T)_{ii'} (VV^T)_{jj'} \end{aligned} \tag{30}$$

The initial U, V, W are the eigenvectors of F, G, H in Eq. 30, respectively. In this step, $w_\ell = 1$. When calculating F, we setup $V^T V = I$ and $W^T W = I$; when calculating G, we setup $W^T W = I$.

We iteratively use current U, V, W and c to update the residue $\{r_\ell\}$ and the weights w_ℓ as Eq. 26, $r_\ell = \sum_{ijk} X^{(\ell)}_{ijk} \Big[X^{(\ell)}_{ijk} - \sum_{i'j'k'} X^{(\ell)}_{i'j'k'} (UU^T)_{ii'} (VV^T)_{jj'} (WW^T)_{kk'} \Big]$ and the cutoff $c =$ median $\{r_\ell\}$. Using the weights w_ℓ, we compute U, V, W as the eigenvectors of F, G, H in Eq. 30.

5. Experimental results

In this section, three benchmark face databases ORL and YALE are used to evaluate the effectiveness of our proposed robust tensor factorization approach. Because Inoue and Urahama have shown the equivalence of tensor based PCA in paper [5], we can compare our method to 2DSVD (GLRAM) without losing generality.

5.1. Experimental results on the ORL database

In the ORL database, there are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expression (open/close eyes, smiling/no-smiling) and facial details (glasses/no glasses). All images were taken against a dark homogeneous background with the subjects in an upright, frontal, position (with tolerance for some side movement). For each subject, we randomly generate an image as outlier using random value between 0 and 255 for every pixel on the noise image. Both 2DSVD and robust tensor factorization algorithms are performed to reconstruct all face images (including the outlier one) under the same subject. Meanwhile we also gradually increase the noise level by adding more outlier images.

5.1.1 Illustration

Fig. 1 illustrates the images reconstruction difference between 2DSVD and robust tensor factorization algorithm with one outlier image (both methods use $K1 = K2 = 30$). For both original and reconstructed images, seven of ten total images plus the outlier image are visualized in Fig. 1. The images on the first, fourth, seventh rows are the original images of subjects. The last images on these three rows are the outlier images that are randomly generated for each subject. The second, fifth, eighth rows show the image reconstruction results using 2DSVD for three subjects; The images on the third, sixth, and ninth rows are the reconstruction results of robust tensor factorization algorithm. Obviously the outlier image influence more on the reconstruction results of 2DSVD and make them fuzzy. Compared to 2DSVD results, robust tensor factorization prevents the image reconstruction from the effect of outlier.

On the other hand, we also can see the outlier image reconstructions get better results on 2DSVD. The robustness factorization method reconstructs the outlier images more fuzzy. The intuition of robust method is to reduce the outlier's effect on other normal images and improve the accuracy on normal images, not outlier image. Our method significantly outperforms 2DSVD on robustness demonstration.

5.1.2 Reconstruction Errors

In Fig. 2, we plot the sum of squared distances from ten original images (the first subject we used in Fig. 1) to both 2DSVD and robust tensor factorization principal subspaces. The values on horizontal axis are images in the sorted order (the first ten are normal images and the 11th is outlier image) and the values on the vertical axis are distances to subspaces using log ratio. In 2DSVD results, the distances from normal images to subspaces are larger than those in

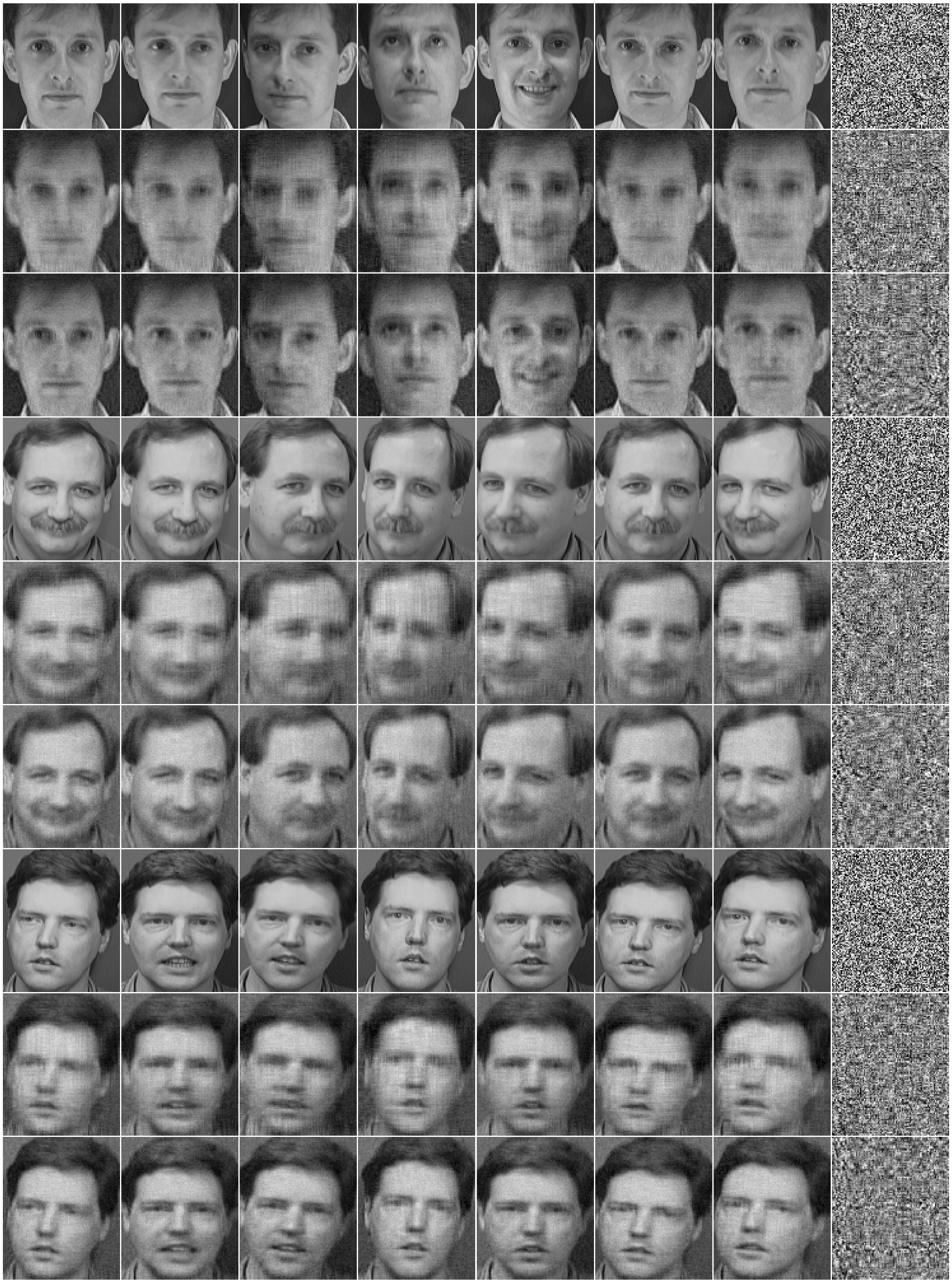


Figure 1. Comparison reconstructed images using 2DSVD and robust tensor factorization with one outlier image (on the right). Original images: 1st, 4th, 7th rows. 2DSVD: 2nd, 5th, 8th rows. Robust factorization: 3rd, 6th, 9th rows.

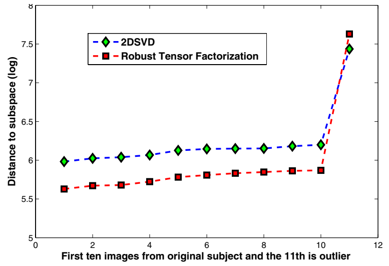


Figure 2. The sum of squared distances (log) from ten original images and outlier image to both 2DSVD and robust tensor factorization principal subspaces in sorted order.

robust tensor factorization. In Fig. 2, outlier image has a lower distance to subspaces calculated by 2DSVD. In other words, 2DSVD gets a worse image representation results for normal images, but obtains a better reconstructed image for outlier. Using L_2 norm as cost function, 2DSVD is confused by outlier.

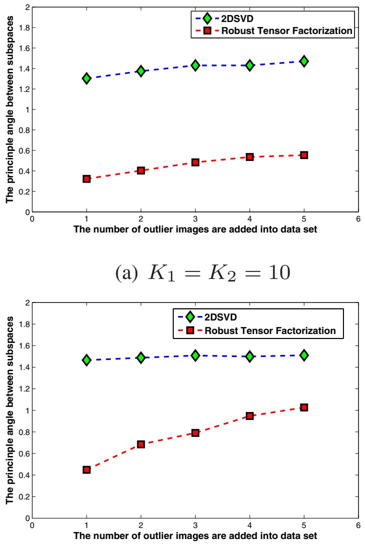
5.1.3 Subspaces

Our intuition is that the subspaces drift due to the presence of outlier images for both 2DSVD and our method. If our method outperform L_2-norm based 2DSVD, than the amount of subspace drift using our method should be smaller than the drift using 2DSVD. The distance between subspaces is calculated by inner products (cosine of angles). The principle angle $\theta \in [0, \pi/2]$ between two subspace L and L' is defined as [4]:

$$\cos\theta = \max_{l \in L} \max_{l' \in L} l^T l', s.t. ||l|| = ||l'|| = 1. \qquad (31)$$

If the angle between two subspace is small, two subspaces are nearly linearly dependent.

The comparison results of every subject in ORL database are calculated and the average results of all 40 subjects are plotted in Fig. 3. In Fig. 3(a), $K_1 = K_2 = 10$ and in Fig. 3(b), $K_1 = K_2 = 20$. The number of outlier images is increasing from 1 to 5 and shown on horizontal axis. The principle angle values on vertical axis are computed by subspaces calculated without outlier and subspaces calculated with different level outliers. When the outliers increase, the principle angles are slowly increasing. It is a natural way, because the new subspace will be further than the previous one as the outlier level increases. After the first outlier image is added, the new subspace of 2DSVD is far away to the original subspace (without outlier). The results of our method are visualized as red dash lines in both Fig. 3(a) and Fig. 3(b). They are still nearby the original subspace before adding outliers. Using the principle angles changes of subspaces, Fig. 3 clearly explain why our method is more robust than 2DSVD.



(a) $K_1 = K_2 = 10$

(b) $K_1 = K_2 = 20$

Figure 3. The principle angles changes between subspaces before adding outlier and subspaces with different level outliers.

5.1.4 Error comparison

In order to evaluate the performance of robust tensor factorization in image representation and reconstruction in quantity, we compare the mean-square errors (MSE) of the 2DSVD and the robust tensor factorization methods.

$$\text{MSE} = \left(\sum_{i=1}^{n} \frac{||X_i - \tilde{X}_i||}{||X_i||} \right) /n \qquad (32)$$

where X_i is the sample image i, $\tilde{X}_i$ is the reconstruction result for image$i$, n is the number of total images in database.

MSE is a natural distance function to measure the reconstruction error of all images through summing the error of each one with normalization item $||X_i||$. There is no bias (none of 2DSVD and our method optimizes this function directly) if we use it to compare both 2DSVD and robust tensor factorization.

All 40 subjects in ORL database are used to calculate MSE and Fig. 4 shows the reconstruction errors. We also compare these two methods to standard PCA and the number of principle components of PCA is selected under the same storage level as 2DSVD which can be calculated from Table 2. When $k = 1, 2, 3$ in 2DSVD, 1 PC is selected for PCA; When $k = 4, 5$ in 2DSVD, 2 PCs are selected for PCA. We have the following observations: 1) robust tensor factorization always achieves the lowest residue error at different k; 2) when k increases, the residue error of 2DSVD and robust tensor factorization decrease; if we add one more outlier image into each subject, the residue error of 2DSVD and robust tensor factorization slightly increase.

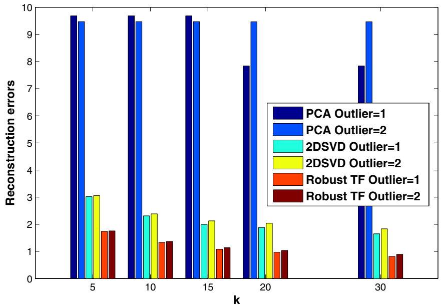


Figure 4. Reconstruction error for ORL dataset using MSE as metric. Three methods are indicated in the insert panel.

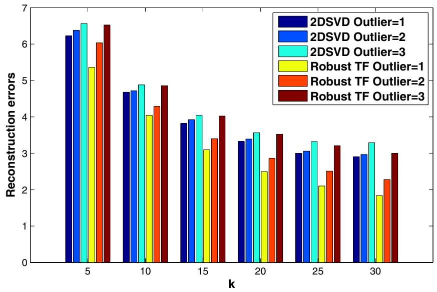


Figure 5. Reconstruction error for Yale dataset using MSE as metric. 2DSVD and our method are indicated in the insert panel.

5.2. Experimental results on the Yale database

The Yale face database contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration. For each subject, we conduct three experiments with one, two, and three random generated outlier images. The image reconstruction results of every subject are calculated using 2DSVD and our method under three outlier conditions. The average result of total 15 subjects are plotted in Fig. 5. Our robust tensor factorization method has a smaller image reconstruction error compared to 2DSVD under all three different outlier conditions. When the number of outlier images increases, the image reconstruction errors of both methods also increase.

6. Conclusion

In this paper, we have proposed to use R_1 norm as cost function of tensor factorization to reduce the effect of outliers in image dataset. A novel robust tensor factorization method is presented with mathematical proof and explicit algorithm. Using this algorithm, we can compute the solutions of robust tensor factorization through the same iterations used by tensor based PCA calculation. There is no extra time and space complexity compared to tensor based PCA. We also experimentally verified that our robust tensor factorization method outperforms the general tensor based PCA on facial image representation and reconstruction through public face database.

References

[1] C. Ding, H. Huang, and D. Luo. Tensor reduction error analysis – applications to video compression and classification. *IEEE Conf. Computer Vision and Pattern Recognition*, 2008. 5

[2] C. Ding and J. Ye. Two-dimensional singular value decomposition (2dsvd) for 2d maps and images. *SIAM Int'l Conf. Data Mining*, pages 32–43, 2005. 1, 3

[3] C. Ding, D. Zhou, X. He, and H. Zha. R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization. *Int'l Conf. Machine Learning*, 2006. 1, 2, 4

[4] G. Golub and C. V. Loan. Matrix computations. *Johns Hopkins, Baltimore*, 1996. 7

[5] K. Inoue and K. Urahama. Equivalence of non-iterative algorithms for simultaneous low rank approximations of matrices. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:154 – 159, 2006. 1, 5

[6] Q. Ke and T. Kanade. Robust l1 norm factorization in the presence of outliers and missing data by alternative convex programming. *IEEE Conf. Computer Vision and Pattern Recognition*, pages 592–599, 2004. 1

[7] Kirby and Sirovich. Application of the kl procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Machine Intell.*, 12:103–108, 1990. 1

[8] F. D. la Torre and M. J. Black. component analysis for computer vision. *International Conference on Computer Vision*, pages 362–369, 2001. 1

[9] T. M. and Pentland. Eigen faces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86, 1991. 1

[10] A. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. *Int'l Conf. Machine Learning*, pages 592–599, 2004. 2

[11] Park and Savvides. Individual kernel tensor-subspaces for robust face recognition: A computationally efficient tensor framework without requiring mode factorization. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 2007. 1

[12] A. Shashua and A. Levin. Linear image coding for regression and classification using the tensor-rank principle. *IEEE Conf. on Computer Vision and Pattern Recognition*, 2001. 1

[13] L. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966. 1

[14] L. Xu and A. Yuille. Robust principal component analysis by self-organizing rules basedon statistical physics approach. *IEEE Transactions on Neural Networks*, 6:131–143, 1995. 1

[15] J. Yang, D. Zhang, A. F. Frangi, and J. Yang. Twodimensional pca: A new approach to appearancebased face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1), 2004. 1, 2, 3

[16] J. Ye. Generalized low rank approximations of matrices. *International Conference on Machine Learning*, 2004. 1, 3

[17] W. Zuo, D. Zhang, and K. Wang. An assembled matrix distance metric for 2dpca-based image recognition. *Pattern Recogn. Lett.*, 27(3):210–216, 2006. 1