Optimal Mean Robust Principal Component Analysis

Feiping Nie FEIPINGNIE@GMAIL.COM
Jianjun Yuan WRIYJJ@GMAIL.COM
Heng Huang HENG@UTA.EDU
Computer Science and Engineering Department, University of Texas at Arlington, Arlington, TX, 76019

Abstract

Principal Component Analysis (PCA) is the most widely used unsupervised dimensionality reduction approach. In recent research, several robust PCA algorithms were presented to enhance the robustness of PCA model. However, the existing robust PCA methods incorrectly center the data using the ℓ_2-norm distance to calculate the mean, which actually is not the optimal mean due to the ℓ_1-norm used in the objective functions. In this paper, we propose novel robust PCA objective functions with removing optimal mean automatically. Both theoretical analysis and empirical studies demonstrate our new methods can more effectively reduce data dimensionality than previous robust PCA methods.

1. Introduction

Machine learning techniques have been widely applied to many scientific domains as diverse as engineering, astronomy, biology, remote sensing, and economics. The dimensionality of scientific data could be more than thousands, such as digital images and videos, gene expressions and DNA copy numbers, documents, and financial time series. As a result, data analysis on such data sets suffers from the curse of dimensionality. To solve this problem, the dimensionality reduction (subspace learning) algorithms have been proposed to project the original high-dimensional feature space to a low-dimensional space, wherein the important statistical properties are well preserved.

Among these methods, the unsupervised dimensionality reduction methods are more useful in the practical applications since labeled data are usually expensive to obtain and we often have no any prior knowledge for new scientific problems. Thus, in this work, we focus on unsupervised dimensionality reduction. Although PCA (Jolliffe, 2002) is the most popularly used method, it is sensitive to the data outliers because of the square ℓ_2-norm based objective function. In real-world applications, the data outliers often largely appear in the datasets, thus PCA may not get the optimal performance.

To address this problem, multiple robust PCA methods have been presented, such as the rotational invariant L1 PCA (Ding et al., 2006) (R1PCA) and convex robust PCA (Wright et al., 2009; Xu et al., 2012). The R1PCA minimizes the $\ell_{2,1}$-norm reconstruction error by imposing the ℓ_2-norm on the feature dimension and the ℓ_1-norm on the data points dimension, such that the effect of data outliers will be reduced by the ℓ_1-norm. Later this idea was extended to robust tensor factorization (Huang & Ding, 2009). The convex robust PCA methods utilize the convex relaxation objectives, such that the global solutions can be achieved. However, all existing robust PCA methods neglect the mean calculation problem. Because the ℓ_1-norm or $\ell_{2,1}$-norm are used in different robust PCA objectives, the square ℓ_2-norm distance based mean is not the correct mean anymore.

In this paper, we propose the novel robust PCA objective functions with removing the optimal mean automatically. We first show that the Euclidean distance based mean is only valid for the traditional PCA. In robust PCA formulations, the ℓ_1-norm or $\ell_{2,1}$-norm are used as loss functions, such that the Euclidean distance based mean is not the correct one. Starting from two widely used robust PCA formulations, we propose our corresponding optimal mean robust PCA objectives. We integrate the mean calculation into the dimensionality reduction optimization objective, such that the optimal mean can be obtained to enhance the dimensionality reduction. The optimization algorithms are derived to solve the proposed non-smooth objectives with convergence analysis. Both theoretical analysis and empirical studies demonstrate our new methods can more effectively reduce data dimensionality than previous robust PCA methods.

Notations: For a vector v, we denote the ℓ_2-norm of v by

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

$\|v\|_2 = \sqrt{v^T v}$. For a matrix M, we denote the (i,j)-th element by m_{ij}, the i-th column by m_i and the i-th row by m^i. Denote $\|M\|_F^2 = Tr(M^T M)$, where $trace(\cdot)$ is the trace operator for a square matrix, and denote $\|M\|_* = Tr((M^T M)^{\frac{1}{2}})$ as the trace norm (nuclear norm). In this paper, we denote $\|M\|_{2,1} = \sum_i \|m_i\|_2$ as the $\ell_{2,1}$-norm of matrix M, denote $\|M\|_1 = \sum_{i,j} |m_{ij}|$ as the ℓ_1-norm of matrix M, and denote $\|M\|_2 = \sigma_{max}(M)$ as the ℓ_2-norm of matrix M, where $\sigma_{max}(M)$ is the largest singular value of M.

2. Principal Component Analysis Revisited

Given a data matrix $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d\times n}$, where d is the dimensionality of data points and n is the number of data points. Essentially, Principal Component Analysis (PCA) is to find a low-rank matrix to best approximate the given data matrix in the sense of Euclidean distance. Suppose the data are centered, i.e. the mean of the data is zero, PCA is to solve the following problem:

$$\min_{rank(Z)=k} \|X - Z\|_F^2 \ . \tag{1}$$

According to the full rank decomposition, any matrix $Z \in R^{d\times n}$ with rank k can be decomposed as $Z = UV^T$, where $U \in R^{d\times k}, V \in R^{n\times k}$. After denoting an identity matrix by I, the problem (1) can be rewritten as

$$\min_{U\in R^{d\times k}, V\in R^{n\times k}, U^T U = I} \left\|X - UV^T\right\|_F^2 \ . \tag{2}$$

Taking the derivative w.r.t. V and setting it to zero, we have $V = X^T U$. As a result, the problem (2) becomes:

$$\max_{U\in R^{d\times k}, U^T U = I} Tr(U^T X X^T U) \ . \tag{3}$$

The columns of the optimal solution U to problem (3) are the k eigenvectors of XX^T corresponding to the k largest eigenvalues.

In the above derivation, we suppose the mean of the data is zero. In the general case that the mean of the data is not zero, PCA is to best approximate the given data matrix with an optimal mean removed. Denote $\mathbf{1}$ as a column vector with all the elements being one, the problem of PCA becomes:

$$\min_{b, rank(Z)=k} \left\|X - b\mathbf{1}^T - Z\right\|_F^2 \ . \tag{4}$$

Note the $b \in \mathbb{R}^{d\times 1}$ in problem (4) is also a variable to be optimized. Similarly, the problem (4) can be rewritten as:

$$\min_{b, U\in R^{d\times k}, V\in R^{n\times k}, U^T U = I} \left\|X - b\mathbf{1}^T - UV^T\right\|_F^2 \ . \tag{5}$$

Taking the derivative w.r.t. V and setting it to zero, we have $V = (X - b\mathbf{1}^T)^T U$. Then the problem (5) becomes:

$$\min_{b, U\in R^{d\times k}, U^T U = I} \left\|X - b\mathbf{1}^T - UU^T(X - b\mathbf{1}^T)\right\|_F^2 \ . \tag{6}$$

Taking the derivative w.r.t. b and setting it to zero, we have $(I - UU^T)(b\mathbf{1}^T - X)\mathbf{1} = 0$. Suppose $U^\perp$ is the orthogonal complement of U, i.e. $[U, U^\perp]$ is orthonormal matrix. Then for any vector $(b\mathbf{1}^T - X)\mathbf{1}$, it can be represented as:

$$(b\mathbf{1}^T - X)\mathbf{1} = U\alpha + U^\perp\beta \, . \tag{7}$$

So we have $(I - UU^T)(U\alpha + U^\perp\beta) = 0 \Leftrightarrow U^\perp\beta = 0 \Leftrightarrow \beta = 0$. Then Eq. (7) becomes:

$$b = \frac{1}{n}(X\mathbf{1} + U\alpha), \tag{8}$$

where α could be any k-dimensional column vector. Denote a centering matrix $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, by substituting Eq. (8) into the problem (6), we have:

$$\max_{U\in R^{d\times k}, U^T U = I} Tr(U^T X H X^T U) \, . \tag{9}$$

We can see that no matter the data X is centered or not, the problem (9) is unchanged. We can simply set $\alpha = 0$ in Eq. (8), so the optimal mean in the problem (4) is $b = \frac{1}{n}X\mathbf{1}$. That is to say, in traditional PCA, we can simply center the data such that $X\mathbf{1} = 0$, and then solve Eq. (3) instead of solving Eq. (9).

3. Robust Principal Component Analysis with Optimal Mean

The problem (4) in the traditional PCA can be written as $\min\limits_{b, rank(Z)=k} \sum_{i=1}^n \|x_i - b - z_i\|_2^2$. It is known that squared loss function is very sensitive to outliers. Therefore, the squared approximation errors will make the traditional PCA not robust to outliers in the data.

Previous robust PCA methods use a non-squared loss function to improve the robustness, but still center data via the ℓ_2-norm distance based mean, which is incorrect. Instead of using the ℓ_2-norm distance based mean, we propose a new robust PCA with an optimal mean automatically removed in the given data. Our new robust PCA is to solve the following problem:

$$\min_{b, rank(Z)=k} \sum_{i=1}^n \|x_i - b - z_i\|_2 \, , \tag{10}$$

where we optimize the mean in the robust PCA objective. We can write the problem (10) in matrix form as follows:

$$\min_{b, rank(Z)=k} \left\|X - b\mathbf{1}^T - Z\right\|_{2,1} . \tag{11}$$

Similarly, the problem (11) can be rewritten as:

$$\min_{b, U\in R^{d\times k}, V\in R^{n\times k}, U^T U = I} \sum_{i=1}^n \left\|x_i - b - U(v^i)^T\right\|_2 . \tag{12}$$

For each i, by setting the derivative *w.r.t* v^i to zero, we have $v^i = (x_i - b)^T U$. Substituting it into Eq. (12), we arrive at the following problem[1]:

$$\min_{b,U\in R^{d\times k},U^TU=I} \sum_{i=1}^{n} \left\|(I-UU^T)(x_i-b)\right\|_2 . \quad (13)$$

In this paper, we use an iterative re-weighted method to solve the problem (13). The detailed algorithm is outlined in Algorithm 1, and the theoretical analysis of the algorithm is given in the last of this section. In each iteration, we solve the following problem:

$$\min_{b,U\in R^{d\times k},U^TU=I} \sum_{i=1}^{n} d_{ii} \left\|(I-UU^T)(x_i-b)\right\|_2^2, \quad (14)$$

where d_{ii} is the weights as calculated in Algorithm 1.

Taking the derivative w.r.t. b and then setting it to zero, we have $(I - UU^T)(b\mathbf{1}^T - X)D\mathbf{1} = 0$. Similarly, we can let $(b\mathbf{1}^T - X)D\mathbf{1} = U\alpha + U^{\perp}\beta$ and get $\beta = 0$, so we have $(b\mathbf{1}^T - X)D\mathbf{1} = U\alpha$ and thus we have

$$b = \frac{XD\mathbf{1}}{\mathbf{1}^T D\mathbf{1}} + \frac{U\alpha}{\mathbf{1}^T D\mathbf{1}} \quad (15)$$

where α could be any k-dimensional column vector.

Substituting Eq. (15) into the problem (14), the problem becomes

$$\max_{U\in R^{d\times k},U^TU=I} Tr(U^T X H_d X^T U), \quad (16)$$

where $H_d = D - \frac{D\mathbf{1}\mathbf{1}^T D}{\mathbf{1}^T D\mathbf{1}}$ is the weighted centering matrix. The columns of the optimal solution U to problem (16) are the k eigenvectors of XH_dX^T corresponding to the k largest eigenvalues. When the dimensionality is larger than the number of data, i.e. $d > n$, the problem (16) can be efficiently solved by the SVD of the matrix:

$$X\left(D^{\frac{1}{2}} - \frac{D\mathbf{1}\mathbf{1}^T D^{\frac{1}{2}}}{\mathbf{1}^T D\mathbf{1}}\right) . \quad (17)$$

3.1. Theoretical Analysis of Algorithm 1

Theorem 1 *The Algorithm 1 will monotonically decrease the objective of the problem (13) in each iteration until the algorithm converges.*

Proof: In the Steps 1 and 2 of Algorithm 1, denote the updated U and b by $\tilde{U}$ and $\tilde{b}$, respectively. Since the updated U and b are the optimal solutions of the problem (14) and according to the definition of d_{ii} in the Step 3 of Algorithm 1, we have:

$$\sum_{i=1}^{n} \frac{\|(I-\tilde{U}\tilde{U}^T)(x_i-\tilde{b})\|_2^2}{2\|(I-UU^T)(x_i-b)\|_2} \le \sum_{i=1}^{n} \frac{\|(I-UU^T)(x_i-b)\|_2^2}{2\|(I-UU^T)(x_i-b)\|_2} . \quad (18)$$

According to the Lemma 1 in (Nie et al., 2010), we know

$$\sum_{i=1}^{n}\left(\left\|(I-\tilde{U}\tilde{U}^T)(x_i-\tilde{b})\right\|_2 - \frac{\left\|(I-\tilde{U}\tilde{U}^T)(x_i-\tilde{b})\right\|_2^2}{2\|(I-UU^T)(x_i-b)\|_2}\right) \le \sum_{i=1}^{n}\left(\left\|(I-UU^T)(x_i-b)\right\|_2 - \frac{\left\|(I-UU^T)(x_i-b)\right\|_2^2}{2\|(I-UU^T)(x_i-b)\|_2}\right) \quad (19)$$

Summing Eq. (18) and Eq. (19) on both sides, we have

$$\sum_{i=1}^{n}\left\|(I-\tilde{U}\tilde{U}^T)(x_i-\tilde{b})\right\|_2 \le \sum_{i=1}^{n}\left\|(I-UU^T)(x_i-b)\right\|_2 \quad (20)$$

Since the problem (13) has an obvious lower bound 0, the Algorithm 1 will converge. The equality in Eq. (20) holds only when the Algorithm 1 converges. Thus, in each iteration, Algorithm 1 monotonically decreases the objective of the problem (13) until the algorithm converges. □

Theorem 2 *The Algorithm 1 will converge to a local minimal solution to the problem (13).*

Proof: The Lagrangian function of problem (13) is:

$$\mathcal{L}_1(U,b,\Lambda) = \sum_{i=1}^{n}\left\|(I-UU^T)(x_i-b)\right\|_2 - Tr((U^TU-I)\Lambda) . \quad (21)$$

Taking the derivative w.r.t. U and b respectively and setting them to zero, we get the KKT condition of the problem (13)

Algorithm 1 Algorithm to solve the problem (13).

Initialize D as an identity matrix
while not converge **do**
1. Update the columns of U by the k right singular vectors of $X(D^{\frac{1}{2}} - \frac{D\mathbf{1}\mathbf{1}^T D^{\frac{1}{2}}}{\mathbf{1}^T D\mathbf{1}})$ corresponding to the k largest singular values.
2. Update b by $b = \frac{XD\mathbf{1}}{\mathbf{1}^T D\mathbf{1}}$
3. Update the diagonal matrix D, where the i-th diagonal element of D is updated by $d_{ii} = \frac{1}{2\|(I-UU^T)(x_i-b)\|_2}$

end while

---

[1]In practice, similar to (Nie et al., 2010), the $\|z\|_2$ is replaced by $\sqrt{z^T z + \varepsilon}$, where $\varepsilon \to 0$.

as follows:

$$\frac{\partial \sum_{i=1}^{n} \left\|(I-UU^T)(x_i-b)\right\|_2}{\partial U} - U\Lambda = 0$$
$$\frac{\partial \sum_{i=1}^{n} \left\|(I-UU^T)(x_i-b)\right\|_2}{\partial b} = 0 \quad (22)$$

Using the matrix calculus, we can write the Eq. (22) as follows:

$$\sum_{i=1}^{n} \frac{(I-UU^T)(x_i-b)(b-x_i)^T U}{\|(I-UU^T)(x_i-b)\|_2} - U\Lambda = 0$$
$$\sum_{i} \frac{(I-UU^T)(b-x_i)}{\|(I-UU^T)(x_i-b)\|_2} = 0 \quad (23)$$

In each iteration of Algorithm 1, we find the optimal solution to the problem (14). Thus the converged solution of Algorithm 1 satisfies the KKT condition of problem (14). The Lagrangian function of problem (14) is:

$$\mathcal{L}_2(U,b,\Lambda) = \sum_{i=1}^{n} d_{ii} \left\|(I-UU^T)(x_i-b)\right\|_2^2 - Tr((U^TU-I)\Lambda)\,. \quad (24)$$

Taking the derivative w.r.t. U and b respectively and setting them to zero, we get the KKT condition of the problem (14) as follows:

$$\frac{\partial \sum_{i=1}^{n} d_{ii} \left\|(I-UU^T)(x_i-b)\right\|_2^2}{\partial U} - U\Lambda = 0$$
$$\frac{\partial \sum_{i=1}^{n} d_{ii} \left\|(I-UU^T)(x_i-b)\right\|_2^2}{\partial b} = 0 \quad (25)$$

Similarly, we can write the Eq. (25) as follows using the matrix calculus:

$$\sum_{i=1}^{n} 2d_{ii}(I-UU^T)(x_i-b)(b-x_i)^T U = U\Lambda$$
$$\sum_{i} 2d_{ii}(I-UU^T)(b-x_i) = 0 \quad (26)$$

According to the definition of d_{ii} in Algorithm 1, we can see that Eq. (26) is the same as Eq. (23) when the Algorithm 1 is converged. Therefore, the converged solution of Algorithm 1 satisfies Eq. (23), the KKT condition of problem (14). Thus the converged solution of Algorithm 1 is a local minimal solution to the problem (13). ☐

3.2. Extension to a General Algorithm

It is worth noting that the content in this section is a parallel work with (Nie et al., 2010)[2]. Later, we found that the reweighted algorithm applied in Algorithm 1 can be used to solve the following more general problem:

Algorithm 2 Algorithm to solve the problem (27).

Initialize $x \in \mathcal{C}$
while not converge **do**
1. For each i, calculate $D_i = h_i'(g_i(x))$
2. Update x by the optimal solution to the problem $\min_{x\in\mathcal{C}} f(x) + \sum_i Tr(D_i^T g_i(x))$

end while

$$\min_{x\in\mathcal{C}} f(x) + \sum_i h_i(g_i(x)), \quad (27)$$

where $h_i(x)$ is an arbitrary **concave** function in the domain of $g_i(x)$. The details to solve problem (27) is described in Algorithm 2, where $h_i'(g_i(x))$ denotes any supergradient of the concave function h_i at point $g_i(x)$. According to the definition of supergradient, it can be easily proved that the Algorithm 2 converges. It can be seen that the converged solution satisfies KKT condition of problem (27), thus the Algorithm 2 will converge to a local optimal solution to the problem (27). Empirical evidences show Algorithm 2 converges very fast and usually converges in 20 iterations.

Algorithm 2 is very easy to implement and powerful. For example, it can be used to minimize the ℓ_p-norm, the $\ell_{2,p}$-norm, the Schatten ℓ_p-norm (Nie et al., 2012), and many robust loss functions such as the capped (truncated) ℓ_p-norm, where $0 < p \le 2$.

More interestingly, if we need to **maximize** the objective in Eq.(27), we only require that $h_i(x)$ is an arbitrary **convex** function in the domain of $g_i(x)$. In this case, in Algorithm 2, the $h_i'(g_i(x))$ in the first step is changed to be any subgradient of h_i at point $g_i(x)$, and the 'min' in the second step is changed to 'max'. Therefore, it can be verified that the Algorithm 2 can also be used to maximize the ℓ_p-norm (Nie et al., 2011), the $\ell_{2,p}$-norm, and the Schatten ℓ_p-norm, where $p \ge 1$.

4. Optimal Mean Robust PCA with Convex Relaxation

Recently, a convex relaxed robust PCA was proposed to solve the following problem (Wright et al., 2009):

$$\min_Z \|X-Z\|_1 + \gamma\|Z\|_*\,. \quad (28)$$

To better pursuit the outliers in the data points, a recent work is proposed to solve the following problem (Xu et al., 2012):

$$\min_Z \|X-Z\|_{2,1} + \gamma\|Z\|_*\,. \quad (29)$$

[2] The motivation to derive this kind of algorithm is similar to the motivation in Eq.(21) of (Nie et al., 2009) for solving the trace ratio problem. It is a very useful trick for algorithm derivation.

Algorithm 3 Algorithm to solve the problem (30).

Let $1 < \rho < 2$. Initialize $\mu = 0.1$, $E = 0$, $\Lambda = 0$
while not converge **do**
1. Update b and Z by solving

$$\min_{b,Z} \frac{1}{2} \left\| \tilde{X} - b\mathbf{1}^T - Z \right\|_F^2 + \tilde{\gamma}\|Z\|_* \qquad (32)$$

where $\tilde{X} = X - E + \frac{1}{\mu}\Lambda$ and $\tilde{\gamma} = \frac{\gamma}{\mu}$.
2. Update E by solving

$$\min_{E} \frac{1}{2} \left\| E - \tilde{X} \right\|_F^2 + \tilde{\gamma}\|E\|_{2,1} \qquad (33)$$

where $\tilde{X} = X - b\mathbf{1}^T - Z + \frac{1}{\mu}\Lambda$ and $\tilde{\gamma} = \frac{1}{\mu}$.
3. Update Λ by $\Lambda = \Lambda + \mu(X - b\mathbf{1}^T - Z - E)$
3. Let $\mu = \min(\rho\mu, 10^8)$
end while

However, all these work don't take the optimal mean into account. Although we can center the data such that $X\mathbf{1} = 0$ before solving problem (29), the mean $b = \frac{1}{n}X\mathbf{1}$ is not necessarily the optimal mean in the problem (29) since the $\ell_{2,1}$-norm loss function instead of the Frobenious norm loss function is used in the objective.

In this section, we consider the optimal mean for the $\ell_{2,1}$-norm loss function, and propose to solve the following problem:

$$\min_{b,Z} \left\| X - b\mathbf{1}^T - Z \right\|_{2,1} + \gamma\|Z\|_* . \qquad (30)$$

We can see problem (30) is a convex relaxation of problem (11). We use Augmented Lagrangian Multiplier (ALM) method to solve the proposed problem (30).

First, problem (30) can be rewritten as:

$$\min_{b,Z,E,X-b\mathbf{1}^T-Z=E} \|E\|_{2,1} + \gamma\|Z\|_* . \qquad (31)$$

With the ALM method, we need to solve the following augmented problem in each iteration:

$$\min_{b,Z,E} \|E\|_{2,1} + \gamma\|Z\|_* + \frac{\mu}{2}\|X - b\mathbf{1}^T - Z - E + \frac{1}{\mu}\Lambda\|_F^2$$

. Solving this problem with joint variables b, Z, E is difficult, we use an alternating method to solve it. The detailed algorithm is outlined in Algorithm 3. When fix E, we solve problem (32) to update b, Z, and when fix b, Z, we solve problem (33) to update E.

The problem (33) can be easily solved and has the closed form solution as follows: $e_i = (\|\tilde{x}_i\|_2 - \tilde{\gamma})_+ \frac{\tilde{x}_i}{\|\tilde{x}_i\|_2}$, where $(s)_+$ is defined as $(s)_+ = max(0, s)$.

We will see that the problem (32) can also be solved with the closed form solution. First, we have the following lemmas:

Lemma 1 *Suppose $A = USV^T$, where the ℓ_2-norms of all the columns of U and V are 1, and S is a nonnegative diagonal matrix. Then $\|A\|_* \le TrS$.*

Proof: Note that $\|A\|_* = \max_{X^TX=I,Y^TY=I} Tr(X^TAY)$, so we have $\|A\|_* = \max_{X^TX=I,Y^TY=I} Tr(X^TUSV^TY) = \max_{X^TX=I,Y^TY=I} \sum_j s_j \sum_i x_i^T u_j v_j^T y_i \le \sum_j s_j = TrS$. The inequality in the last but one step holds since $\sum_i x_i^T u_j v_j^T y_i = v_j^T Y X^T u_j \le \sigma_{\max}(YX^T) = 1$, where $\sigma_{\max}(YX^T)$ denotes the largest singular value of matrix YX^T which is 1 since $X^TX = I, Y^TY = I$. □

Lemma 2 $\|Z\|_* = \min_{Z=AB^T} \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$.

Proof: Suppose $Z = AB^T$, then $\|Z\|_* = \left\|AB^T\right\|_* \le \sum_i \|a_i\|_2\|b_i\|_2 \le \sqrt{\sum_i \|a_i\|_2^2 \sum_i \|b_i\|_2^2} = \|A\|_F\|B\|_F \le \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$, where the second step holds according to Lemma 1 and the third step holds according to Cauchy-Schwarz inequality. On the other hand, suppose the SVD of Z is $Z = U\Sigma V^T$, let $A = U\Sigma^{\frac{1}{2}}$ and $B = V\Sigma^{\frac{1}{2}}$, then we have $Z = AB^T$ and $\|Z\|_* = \left\|U\Sigma V^T\right\|_* = Tr(\Sigma) = \frac{1}{2}(\|U\Sigma^{\frac{1}{2}}\|_F^2 + \|\Sigma^{\frac{1}{2}}V^T\|_F^2) = \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$. Therefore, under the constraint $Z = AB^T$, the minimization of $\frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$ is $\|Z\|_*$. □

Lemma 3 *If $ZH \ne Z$, then $\|ZH\|_* < \|Z\|_*$.*

Proof: Suppose $\{\hat{A}, \hat{B}\} = \arg\min_{Z=AB^T} \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2)$, then according to Lemma 2 we have $\|Z\|_* = \frac{1}{2}(\|\hat{A}\|_F^2 + \|\hat{B}\|_F^2)$ and $Z = \hat{A}\hat{B}^T$. Thus $ZH = \hat{A}(H\hat{B})^T$, according to $ZH \ne Z$ and Lemma 2, we have

$$\begin{aligned} \|ZH\|_* &= \min_{ZH=AB^T} \frac{1}{2}(\|A\|_F^2 + \|B\|_F^2) \\ &\le \frac{1}{2}(\|\hat{A}\|_F^2 + \|H\hat{B}\|_F^2) \\ &= \frac{1}{2}(\|\hat{A}\|_F^2 + \|\hat{B}\|_F^2 - \frac{1}{n}\mathbf{1}^T\hat{B}\hat{B}^T\mathbf{1}) \\ &< \frac{1}{2}(\|\hat{A}\|_F^2 + \|\hat{B}\|_F^2) = \|Z\|_* , \end{aligned} \qquad (34)$$

where the last inequality holds since $ZH \ne Z$ and $Z = \hat{A}\hat{B}^T$ indicates $\hat{B}^T\mathbf{1} \ne \mathbf{0}$. Therefore, we have $ZH \ne Z \Rightarrow \|ZH\|_* < \|Z\|_*$. □

The problem (32) can be solved according to the following theorem:

Theorem 3 *The unique optimal solution $\hat{b}$, $\hat{Z}$ to the problem (32) is $\hat{b} = \frac{1}{n}\tilde{X}\mathbf{1}$ and $\hat{Z} = U(\Sigma - \tilde{\gamma}I)_+V^T$, where $\tilde{X}H = U\Sigma V^T$ is the compact Singular Value Decomposition (SVD) of $\tilde{X}H$ and the (i,j)-th element of $(M)_+$ is defined as $max(0, m_{ij})$.*

Proof: Taking the derivative of Eq. (32) w.r.t. b and setting it to zero, we have:

$$\hat{b} = \frac{1}{n}\tilde{X}\mathbf{1} - \frac{1}{n}\hat{Z}\mathbf{1}\,. \tag{35}$$

So the problem (32) becomes:

$$\min_{b,Z} \frac{1}{2}\|\tilde{X}H - ZH\|_F^2 + \tilde{\gamma}\|Z\|_*\,. \tag{36}$$

First, we verify that $\hat{Z} = U(\Sigma - \tilde{\gamma}I)_+V^T$ satisfies:

$$0 \in \hat{Z}H - \tilde{X}H + \tilde{\gamma}\partial\|\hat{Z}\|_*\,. \tag{37}$$

Denote the compact SVD of $\tilde{X}H$ as $\tilde{X}H = U\Sigma V^T = U_1\Sigma_1V_1^T + U_2\Sigma_2V_2^T$, where the singular values in Σ_1 are all greater than $\tilde{\gamma}$ and the singular values in Σ_2 are all smaller than or equal to $\tilde{\gamma}$. Then $\hat{Z}$ can be written as $U_1(\Sigma_1 - \tilde{\gamma}I)V_1^T$. On the other hand, we have $\tilde{X}H\mathbf{1} = \mathbf{0} \Rightarrow U\Sigma V^T\mathbf{1} = \mathbf{0} \Rightarrow V^T\mathbf{1} = \mathbf{0} \Rightarrow \hat{Z}\mathbf{1} = \mathbf{0} \Rightarrow \hat{Z}H = \hat{Z}$. So we have the following equation:

$$\begin{aligned} & \tilde{X}H - \hat{Z}H \\ = \; & U_1\Sigma_1V_1^T + U_2\Sigma_2V_2^T - U_1(\Sigma_1 - \tilde{\gamma}I)V_1^T \\ = \; & \tilde{\gamma}U_1V_1^T + U_2\Sigma_2V_2^T\,. \end{aligned} \tag{38}$$

It is known (Watson, 1992) that the set of the subgradients of $\|\hat{Z}\|_*$ is:

$$\partial\|\hat{Z}\|_* = \{U_1V_1^T + M : U_1^TM = 0, MV_1 = 0, \|M\|_2 \leq 1\}$$

So we have $U_1V_1^T + \frac{1}{\tilde{\gamma}}U_2\Sigma_2V_2^T \in \partial\|\hat{Z}\|_*$, and according to Eq. (38) we have:

$$\tilde{X}H - \hat{Z}H \in \tilde{\gamma}\partial\|\hat{Z}\|_*\,. \tag{39}$$

Therefore, $\hat{Z} = U(\Sigma - \tilde{\gamma}I)_+V^T$ satisfies Eq. (37). Since the problem (32) is convex, $\hat{Z} = U(\Sigma - \tilde{\gamma}I)_+V^T$ is one of the optimal solution to the problem (32).

Unlike the problem (4) which has many optimal solutions, we further show that the solution $\hat{b}$, $\hat{Z}$ is the unique optimal solution to the problem (32).

According to Lemma 3, we know the optimal solution $\hat{Z}$ to the problem (36) must satisfy $\hat{Z}H = \hat{Z}$, thus the optimal $\hat{b}$ is $\hat{b} = \frac{1}{n}\tilde{X}\mathbf{1}$ according to Eq. (35), and the problem (36) is equivalent to the following problem:

$$\min_{b,Z} \frac{1}{2}\|\tilde{X}H - Z\|_F^2 + \tilde{\gamma}\|Z\|_*\,. \tag{40}$$

Since the problem (40) is strictly convex, the optimal solution is unique. Therefore, $\hat{b} = \frac{1}{n}\tilde{X}\mathbf{1}$, $\hat{Z} = U(\Sigma - \tilde{\gamma}I)_+V^T$ is the unique optimal solution to the problem (32). □

5. Experimental Results

The main goal of PCA is to reduce the dimensionality such that the reduced features represent and reconstruct the original data as good as possible. In the experiments, we show how well the reconstruction of the proposed new optimal mean robust PCA methods compared to the previous PCA and robust PCA methods. The compared PCA methods include original PCA (denoted as PCA), robust PCA with L1 maximization (denoted as L1PCA) (Kwak, 2008), R1PCA (Ding et al., 2006) and robust PCA with convex relaxation (solving Eq. (29), denoted as CRPCA) (Xu et al., 2012). The proposed optimal mean robust PCA methods solving Eq. (11) and Eq. (30) are denoted as RPCA-OM and CRPCA-OM, respectively.

5.1. Experimental Setup

In the experiments, we use 12 benchmark face image datasets, including AT&T (Samaria & Harter, 1994), UMIST (Graham & Allinson, 1998), Yale (face data), YaleB (Georghiades et al., 2001), Palm (Hou et al., 2009), CMU-PIE (Sim & Baker, 2003), FERET (Philips et al., 1998), MSRA, Coil (Nene et al., 1996), JAFFE, MNIST, and AR. We downloaded the image data from different websites. Some of them were resized by previous work, but this won't effect our evaluation.

In each dataset, we randomly select 20% images and randomly place a $1/4$ size of occlusion in the selected images. The reconstruction error is calculated as $\sum_i \|x_i^r - x_i^o\|_2$, where x_i^o is the original image without occlusion and x_i^r is the reconstructed image.

5.2. Reconstruction Error Comparison for Robust PCA methods

We first compare the reconstruction error of PCA, L1PCA, R1PCA and our proposed RPCA-OM methods on 12 benchmark datasets in Table 1. Because these methods can share the common reduced dimensionality, their reconstruction errors can be compared under the same dimension. In Table 1, we compare the reconstruction errors under nine different dimensions from 10 to 50. We cannot list more results due to limited space.

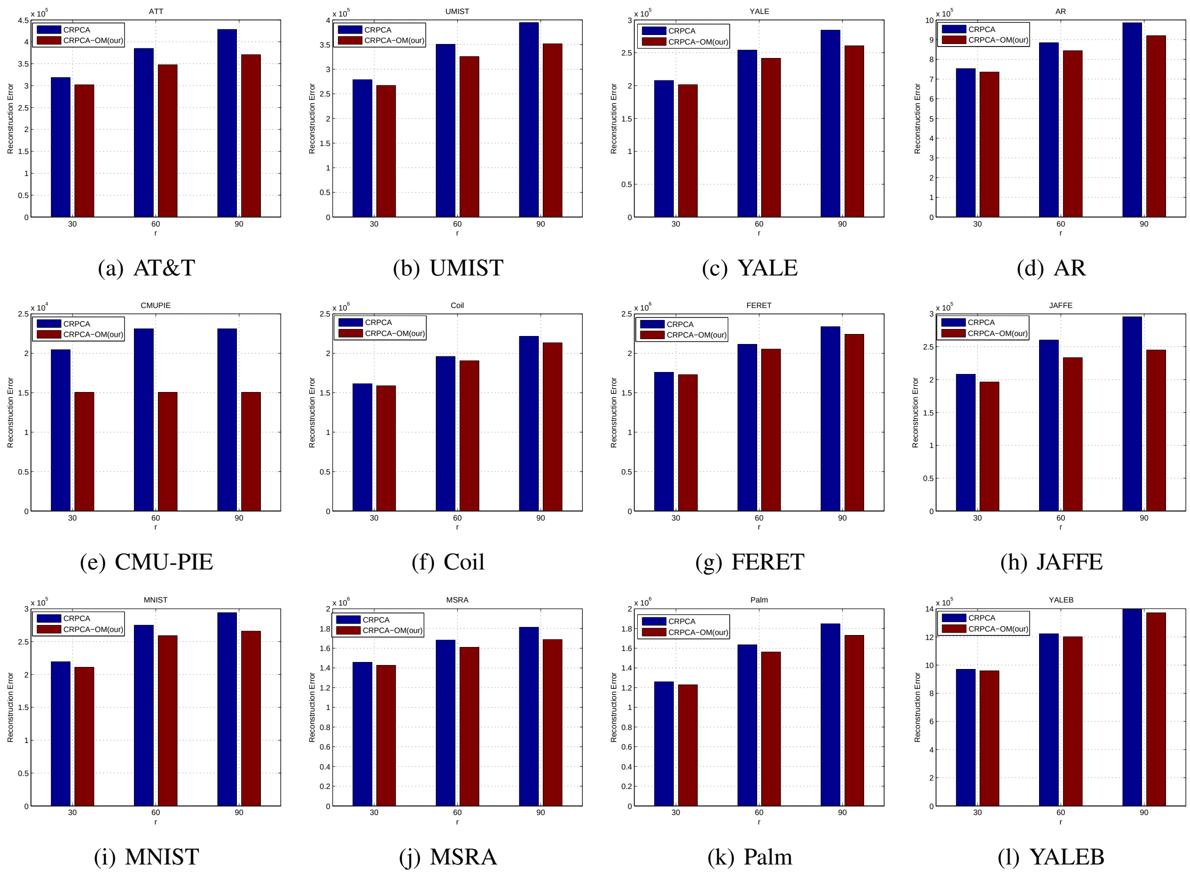
From Table 1, we can observe that:

1) The robust PCA methods R1PCA and RPCA-OM are consistently better than the other two PCA methods, when there are occlusions in the data which indicates these robust PCA methods are effective and robust to outliers and noise in the data, except for projected dimension 15 in YALEB.

2) L1PCA is better than PCA in some cases but is worse in other cases. The reason is that L1PCA is to maximize the ℓ_1-norm, but not to minimize the reconstruction error.

MNIST ($\times 10^4$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	20.964	17.354	15.849	14.866	13.890	13.193	12.559	11.924	10.776
	L1PCA	18.255	16.384	15.198	14.282	13.584	12.895	12.256	11.666	11.359
	R1PCA	17.911	16.005	14.865	**13.895**	13.044	12.350	11.719	11.148	10.774
	RPCA-OM (our)	**17.885**	**15.974**	**14.859**	13.899	**13.037**	**12.303**	**11.707**	**11.064**	**10.627**
JAFFE ($\times 10^4$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	34.429	29.642	25.499	24.791	24.020	23.188	22.321	21.066	12.855
	L1PCA	17.162	15.852	14.958	14.472	14.052	13.635	13.328	13.021	12.594
	R1PCA	16.824	15.567	14.847	14.414	13.973	13.662	13.275	12.993	12.641
	RPCA-OM (our)	**16.640**	**15.420**	**14.716**	**14.265**	**13.818**	**13.526**	**13.155**	**12.867**	**12.500**
YALEB ($\times 10^5$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	10.451	9.6717	9.0074	8.5250	8.0723	7.7539	7.4673	7.2194	6.8686
	L1PCA	9.9809	**9.3133**	8.7559	8.2912	7.8940	7.5742	7.2877	7.0594	6.8297
	R1PCA	10.020	9.3594	8.7214	8.2502	7.8399	7.5261	7.2370	6.9991	6.7686
	RPCA-OM (our)	**9.9742**	9.3290	**8.6944**	**8.2169**	**7.8154**	**7.4969**	**7.2078**	**6.9719**	**6.7420**
Coil20 ($\times 10^5$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	19.632	18.912	17.766	17.092	16.554	16.029	15.528	15.204	14.907
	L1PCA	19.694	18.091	17.095	16.446	15.913	15.497	15.124	14.799	14.509
	R1PCA	19.569	17.988	16.972	16.326	15.828	15.411	15.032	14.706	14.419
	RPCA-OM (our)	**19.397**	**17.905**	**16.903**	**16.270**	**15.766**	**15.376**	**14.995**	**14.667**	**14.380**
MSRA ($\times 10^5$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	28.314	27.593	26.326	24.612	23.127	21.938	21.511	20.592	20.039
	L1PCA	16.181	15.372	14.765	14.265	13.827	13.476	13.180	12.945	12.737
	R1PCA	16.237	15.440	14.786	14.180	13.796	13.462	13.162	12.927	12.728
	RPCA-OM (our)	**16.112**	**15.302**	**14.683**	**14.068**	**13.708**	**13.345**	**13.074**	**12.837**	**12.631**
FERET ($\times 10^5$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	21.619	20.394	19.543	18.864	18.294	17.753	17.307	16.880	16.518
	L1PCA	21.570	20.308	19.470	18.805	18.232	17.726	17.293	16.871	16.536
	R1PCA	21.481	20.302	19.435	18.791	18.220	17.673	17.232	16.811	16.442
	RPCA-OM (our)	**21.430**	**20.262**	**19.395**	**18.763**	**18.154**	**17.638**	**17.200**	**16.784**	**16.413**
CMU-PIE ($\times 10^3$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	7.8659	6.6812	5.8340	5.2123	4.7648	4.3930	4.0672	3.8033	3.5799
	L1PCA	7.8727	6.6765	5.8485	5.2148	4.7691	4.3852	4.0694	3.8020	3.5889
	R1PCA	7.8431	6.6478	5.8068	5.1777	4.7334	4.3555	4.0319	3.7716	3.5473
	RPCA-OM (our)	**7.8010**	**6.6134**	**5.7799**	**5.1547**	**4.7089**	**4.3315**	**4.0096**	**3.7547**	**3.5345**
AT&T ($\times 10^4$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	29.333	27.426	26.152	25.119	24.270	23.613	23.002	22.425	21.931
	L1PCA	28.948	27.141	26.026	24.980	24.128	23.432	22.873	22.319	21.798
	R1PCA	28.877	27.109	25.784	24.857	23.903	23.266	22.742	22.176	21.676
	RPCA-OM (our)	**28.774**	**27.036**	**25.713**	**24.756**	**23.814**	**23.153**	**22.631**	**22.064**	**21.549**
UMIST ($\times 10^4$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	28.917	27.248	25.966	24.910	24.053	23.330	22.780	22.180	21.698
	L1PCA	28.568	27.022	25.765	24.726	23.888	23.243	22.640	22.054	21.628
	R1PCA	28.486	26.971	25.709	24.603	23.740	23.050	22.532	21.964	21.448
	RPCA-OM (our)	**28.383**	**26.900**	**25.600**	**24.498**	**23.655**	**22.944**	**22.424**	**21.850**	**21.312**
AR ($\times 10^4$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	23.293	21.004	19.601	18.428	17.544	16.816	16.164	15.617	15.185
	L1PCA	23.298	21.006	19.612	18.499	17.637	16.905	16.225	15.723	15.232
	R1PCA	23.032	20.837	19.417	18.251	17.356	16.644	15.998	15.428	14.999
	RPCA-OM (our)	**22.912**	**20.767**	**19.321**	**18.193**	**17.279**	**16.530**	**15.879**	**15.306**	**14.874**
YALE ($\times 10^4$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	22.119	19.445	17.995	16.940	16.152	15.658	15.139	14.728	14.218
	L1PCA	17.959	16.591	15.514	14.872	14.310	13.745	13.308	12.874	12.434
	R1PCA	17.742	16.253	15.207	14.526	14.015	13.530	12.997	12.631	12.331
	RPCA-OM (our)	**17.692**	**15.150**	**14.461**	**14.009**	**13.507**	**12.890**	**12.889**	**12.502**	**12.200**
PALM ($\times 10^5$)	Dimension	10	15	20	25	30	35	40	45	50
	PCA	14.703	13.355	12.376	11.596	11.011	10.561	10.119	9.7871	9.4969
	L1PCA	14.734	13.373	12.377	11.628	11.033	10.580	10.156	9.7885	9.4881
	R1PCA	14.665	13.319	12.300	11.527	10.954	10.461	10.053	9.7027	9.4058
	RPCA-OM (our)	**14.651**	**13.300**	**12.287**	**11.499**	**10.935**	**10.437**	**10.035**	**9.6745**	**9.3826**

Table 1. Reconstruction error comparisons of four PCA methods on 12 benchmark datasets using different dimensions. The best reconstruction result under each dimension is bolded.



(a) AT&T (b) UMIST (c) YALE (d) AR

(e) CMU-PIE (f) Coil (g) FERET (h) JAFFE

(i) MNIST (j) MSRA (k) Palm (l) YALEB

Figure 1. Reconstruction errors under different γ obtained by CRPCA and our CRPCA-OM methods.

3) Since the optimal mean is considered in the reconstruction error minimization, our RPCA-OM method consistently outperforms other three methods in most cases.

5.3. Reconstruction Error Comparison for Convex Robust PCA methods

In CRPCA and our CRPCA-OM methods, the projection dimension cannot be selected. We can only get the reconstruction data via adjusting the parameter γ. Thus, we compare these two methods together. We choose the range of γ based on the suggestion from (Wright et al., 2009), in which the γ is suggested with the scale of $m^{\frac{1}{2}}$ (m is the dimension of matrix Z). Considering the size of images used in our experiments, we select the range of γ from 30 to 90. The reconstruction error comparison of these two methods are shown in Fig.1. From Fig.1, we can conclude that:

1) As the value of the regularization parameter γ increases, the reconstruction error for both methods increases as well, which is due to the weight we put in the reconstruction error decreases. As a result, the algorithm pays less attention to minimizing the reconstruction error.

2) Our CRPCA-OM method is consistently better than CRPCA approach, because CRPCA-OM method takes into account the optimal mean in the Eq. (29), which as we have said previously is not the Frobenious norm loss function's mean, but the $\ell_{2,1}$-norm loss function's mean.

6. Conclusions

In this paper, we proposed the novel optimal mean robust PCA models with automatically removing the correct data mean. To solve the proposed non-smooth objectives, we derive the new optimization algorithms with proved convergence. Both theoretical analysis and empirical results show our new robust PCA with optimal mean models consistently outperform the existing robust PCA methods.

Acknowledgments

Corresponding Author: Heng Huang (heng@uta.edu)
This work was partially supported by US NSF IIS-1117965, IIS-1302675, IIS-1344152.

References

Ding, Chris, Zhou, Ding, He, Xiaofeng, and Zha, Hongyuan. R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization. *Int'l Conf. Machine Learning*, 2006.

face data, YALE. http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

Georghiades, A., Belhumeur, P., and Kriegman, D. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on PAMI*, 23(6):643–660, 2001.

Graham, Daniel B and Allinson, Nigel M. Face recognition: From theory to applications. *NATO ASI Series F, Computer and Systems Sciences*, 163:446–456, 1998.

Hou, Chenping, Nie, Feiping, Zhang, Changshui, and Wu, Yi. Learning an orthogonal and smooth subspace for image classification. *IEEE Signal Process. Lett.*, 16(4): 303–306, 2009.

Huang, Heng and Ding, Chris. Robust tensor factorization using r1 norm. *CVPR*, 2009.

Jolliffe, I. T. *Principal Component Analysis*. Series: Springer series in statistics, New York: Springer-Verlag, 2nd edition, 2002.

Kwak, N. Principal component analysis based on L1-norm maximization. *IEEE Transactions on PAMI*, 30 (9):1672–1680, 2008. ISSN 0162-8828.

Nene, Sameer, Nayar, Shree K., and Murase, Hiroshi. Columbia object image library (coil-100). Technical report, 1996.

Nie, Feiping, Xiang, Shiming, Jia, Yangqing, and Zhang, Changshui. Semi-supervised orthogonal discriminant analysis via label propagation. *Pattern Recognition*, 42 (11):2615–2627, 2009.

Nie, Feiping, Huang, Heng, Cai, Xiao, and Ding, Chris. Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *NIPS*, 2010.

Nie, Feiping, Huang, Heng, Ding, Chris H. Q., Luo, Dijun, and Wang, Hua. Robust principal component analysis with non-greedy L1-norm maximization. In *IJCAI*, pp. 1433–1438, 2011.

Nie, Feiping, Huang, Heng, and Ding, Chris H. Q. Low-rank matrix recovery via efficient schatten p-norm minimization. In *AAAI*, 2012.

Philips, I., Wechsler, H., Huang, J., and Rauss, P. The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16:295–306, 1998.

Samaria, Ferdinando and Harter, Andy. Parameterisation of a stochastic model for human face identification, 1994. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

Sim, T. and Baker, S. The cmu pose, illumination, and expression database. *IEEE Transactions on PAMI*, 25 (12):1615–1617, 2003.

Watson, GA. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

Wright, J., Ganesh, A., Rao, S., and Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *NIPS*, 2009.

Xu, Huan, Caramanis, Constantine, and Sanghavi, Sujay. Robust pca via outlier pursuit. *Information Theory, IEEE Transactions on*, 58(5):3047–3064, 2012.