Robust Principal Component Analysis with Non-Greedy L1-Norm Maximization [*]

Feiping Nie, Heng Huang, Chris Ding, Dijun Luo, Hua Wang
Department of Computer Science and Engineering
University of Texas at Arlington, Arlington, Texas 76019, USA
{feipingnie,dijun.luo,huawangcs}@gmail.com, {heng,chqding}@uta.edu

Abstract

Principal Component Analysis (PCA) is one of the most important methods to handle high-dimensional data. However, the high computational complexity makes it hard to apply to the large scale data with high dimensionality, and the used L2-norm makes it sensitive to outliers. A recent work proposed principal component analysis based on L1-norm maximization, which is efficient and robust to outliers. In that work, a greedy strategy was applied due to the difficulty of directly solving the L1-norm maximization problem, which is easy to get stuck in local solution. In this paper, we first propose an efficient optimization algorithm to solve a general L1-norm maximization problem, and then propose a robust principal component analysis with non-greedy L1-norm maximization. Experimental results on real world datasets show that the non-greedy method always obtains much better solution than that of the greedy method.

1 Introduction

In many real-world applications such as face recognition and text categorization, the dimensionality of data are usually very high. Directly handle the high-dimensional data is computationally expensive and at the same time the performance could be very poor because the number of available data is always limited and the noise in the data would increase dramatically as the dimensionality increases. Dimensionality reduction or distance metric learning is one of the most important and effective methods to handle high-dimensional data [Xiang *et al.*, 2008; Yang *et al.*, 2009; Nie *et al.*, 2010b]. Among the dimensionality reduction methods, Principal Component Analysis (PCA) is one of the most widely applied methods due to its simplicity and effectiveness. Given a dataset, PCA finds a projection matrix to maximize the variance of the projected data points under this projection matrix, and the structure of original data could be effectively preserved under the projection.

In the past decades, the traditional PCA has been successfully applied in many problems. However, it has several drawbacks. First, it has to perform Singular Vector Decomposition (SVD) on input data matrix or eigen-decomposition on convariance matrix, which is computationally expensive and difficult to apply when both the number of data and the dimensionality are very high. Second, it is sensitive to outliers because it is intrinsically based on L2-norm and the outliers with large norm can be exaggerated by using the L2-norm. Many works [Baccini *et al.*, 1996; Aanas *et al.*, 2002; De La Torre and Black, 2003; Ke and Kanade, 2005; Ding *et al.*, 2006; Wright *et al.*, 2009] have devoted effort to alleviate this problem and improve the robustness to outliers. [Baccini *et al.*, 1996; Ke and Kanade, 2005] consider the problem of finding a subspace such that the sum of L1-norm distances of data points to the subspace is minimized. Although the robustness to outliers is improved by this method, it is computationally expensive and more importantly, the used L1-norm is not invariant to rotation and the performance usually very poor when applied to K-means clustering [Ding *et al.*, 2006]. To solve this problem, R1-PCA was proposed which is invariant to rotation and demonstrated favorable performance [Ding *et al.*, 2006]. However, R1-PCA iteratively performs the subspace iteration algorithm in the high-dimensional original space, which is computationally expensive. The extension of R1-PCA to tensor version can be found in [Huang and Ding, 2008].

Recently, a robust principal component analysis based on L1-norm maximization is proposed in [Kwak, 2008], and a similar work can be found in [Galpin and Hawkins, 1987]. This method is invariant to rotation and is also robust to outliers. In [Kwak, 2008], an efficient algorithm is proposed to solve the L1-norm maximization problem. The algorithm only need to perform matrix-vector multiplication, and thus can be applied in the case that both the number of data and the dimensionality are very high. Some works on its tensor version and supervised version can be found in [Li *et al.*, 2010; Liu *et al.*, 2010; Pang *et al.*, 2010]. Due to the difficulty of directly solving the L1-norm maximization problem, all these works use a greedy strategy to solve it. Specifically, the projection directions are sequentially optimized one by one. This kind of greedy method is easy to get stuck in a local solution.

In this paper, we focus on solving the L1-norm maximization problem. We first propose an efficient optimization al-

[*] This research was funded by US NSF-CCF-0830780, 0939187, 0917274, NSF DMS-0915228, NSF CNS-0923494, 1035913.

gorithm to solve a general L1-norm maximization problem. Theoretical analysis guarantees the algorithm will converge and usually converge to a local solution. The L1-norm maximization problem in [Kwak, 2008] is a special case of the general problem, and thus the proposed optimization algorithm can be used to solve it directly in a non-greedy strategy. That is, all the projection directions can be optimized simultaneously. Experimental results on real datasets show that the non-greedy method always obtains much better solution than that of the greedy method.

The rest of this paper is organized as follows: We give a brief review of the work [Kwak, 2008] in Section 2. In Section 3, we propose an efficient algorithm to solve a general L1-norm maximization problem and give theoretical analysis on it. Based on the algorithm, we solve the problem for the principal component analysis with greedy L1-norm maximization in Section 4 and propose a principal component analysis with non-greedy L1-norm maximization in Section 5. In Section 6, we present experiments to verify the effectiveness of the proposed method. Finally, we draw the conclusions in Section 7.

2 Related work

Suppose the given data are $X = [x_1, x_2, ..., x_n] \in \mathbb{R}^{d\times n}$, where n and d are the number and the dimensionality of data points respectively. Without loss of generality, the data $\{x_i\}_{i=1}^n$ are assumed to be centralized, i.e., $\sum_{i=1}^n x_i = 0$.

Denote the projection matrix $W = [w_1, w_2, ..., w_m] \in \mathbb{R}^{d\times m}$. Traditional PCA method maximizes the variance of data in the projected subspace, and to solve the following optimization problem:

$$\max_{W^TW=I} Tr(W^T S_t W), \tag{1}$$

where $S_t = \frac{1}{n}XX^T$ is the covariance matrix, I is the identity matrix and $Tr(\cdot)$ is the trace operator of a matrix. Denote the L1-norm and L2-norm of a vector by $\|\cdot\|_1$ and $\|\cdot\|_2$, respectively. The problem (1) can be reformulated as the following problem:

$$\max_{W^TW=I} \frac{1}{n}\sum_{i=1}^n \left\|W^T x_i\right\|_2^2. \tag{2}$$

Motivated by this reformulation, a recent work [Kwak, 2008] proposed to maximize the L1-norm instead of the L2-norm in PCA, and thus the robustness to outliers is improved. Then the problem becomes:

$$\max_{W^TW=I} \frac{1}{n}\sum_{i=1}^n \left\|W^T x_i\right\|_1 \tag{3}$$

Directly solving this problem is difficult, thus the author use a greedy strategy to solve it. Specifically, the m projection directions $\{w_1, w_2, ..., w_m\}$ are optimized one by one. The first projection direction w_1 is optimized by solving the following problem:

$$\max_{w_1^Tw_1=1} \sum_{i=1}^n \left|w_1^T x_i\right| \tag{4}$$

After the $(k-1)$-th projection direction w_{k-1} has been obtained, the data matrix X is transformed to $X = X - w_{k-1}(w_{k-1})^T X$, and then the k-th projection direction w_k is optimized by solving the following problem:

$$\max_{w_k^Tw_k=1} \sum_{i=1}^n \left|w_k^T x_i\right| \tag{5}$$

In this greedy method, the only problem needed to solve is the problem (5) for each k. The work in [Kwak, 2008] proposed an iterative algorithm to solve this problem. The detailed procedure is:
1) $t = 1$. Initialize $w_k^t \in \mathbb{R}^d$ such that $\|(w_k^t)\|_2 = 1$.
2) For each i, if $(w_k^t)^T x_i < 0$, $\alpha_i = -1$ otherwise $\alpha_i = 1$.
3) Let $v = \sum\limits_{i=1}^n \alpha_i x_i$, and $w_k^{t+1} = v/\|v\|_2$, $t = t+1$.
4) Iteratively perform steps 2 and 3 until converges.

In order to guarantee the algorithm converges to a local maximum, the algorithm adds an additional judgement after convergence. If there exists i such that $(w_k^t)^T x_i = 0$, then let $w_k^t = (w_k^t + \triangle w)/\|w_k^t + \triangle w\|_2$ and go to step 2, where $\triangle w$ is a small nonzero random vector. However, such operation might make the algorithm interminable (for example, suppose there is a data point x that exactly locates on the mean of the data set, then x will be zero after centralization, and thus $(w^t)^T x$ is always zero for any w^t). Moreover, it is possible that there exists i such that $(w_k^t)^T x_i = 0$ at the global maximum. In this case, the algorithm can not have the chance to find the global maximum.

Subsequently, we will first propose an efficient algorithm to solve a general L1-norm maximization problem. Based on it, we also solve the problem (5) for the principal component analysis with greedy L1-norm maximization and propose the principal component analysis with non-greedy L1-norm maximization by directly solve the problem (3). The additional judgement is not required in the new algorithms to obtain a local solution, and the non-greedy method always obtains much better solution than that of the greedy method in practice.

3 An efficient algorithm to solve a general L1-norm maximization problem

Consider a general L1-norm maximization problem as follows (we assume that the objective has an upper bound) :

$$\max_{v\in\mathcal{C}} f(v) + \sum_i |g_i(v)|. \tag{6}$$

where $f(v)$ and $g_i(v)$ for each i are arbitrary functions, and $v \in \mathcal{C}$ is an arbitrary constraint. Although there are many methods to solve the L1-norm **minimization** problem in compressed sensing and sparse learning [Donoho, 2006; Nie *et al.*, 2010a], these methods can not be used to solve the L1-norm **maximization** problem.

Rewriting the problem (6) as the following problem:

$$\max_{v\in\mathcal{C}} f(v) + \sum_i \alpha_i g_i(v), \tag{7}$$

where $\alpha_i = sgn(g_i(v))$, and $sgn(\cdot)$ is the sign function defined as follows: $sgn(x) = 1$ if $x > 0$, $sgn(x) = -1$ if

$x < 0$, and $sgn(x) = 0$ if $x = 0$. Note that α_i depends on v and thus is also a unknown variable. We propose an iterative algorithm to solve the problem (6), and prove that the proposed iterative algorithm will monotonically increase the objective of the problem (6) in each iteration, and will usually converge to a local solution.

The algorithm is described in Algorithm 1. In each iteration, α_i is calculated by current solution v, and the solution v is updated with the current α_i. The iterative procedure is repeated until the algorithm converges.

```
Initialize $v^1 \in \mathcal{C}, t = 1$ ;
while not converge do
    1. For each $i$, calculate $\alpha_i^t = sgn(g_i(v^t))$ ;
    2. $v^{t+1} = \arg\max_{v \in \mathcal{C}} f(v) + \sum_i \alpha_i^t g_i(v)$ ;
    3. $t = t + 1$ ;
end
Output: $v^t$.
```

Algorithm 1: An efficient algorithm to solve a general L1-norm maximization problem (6).

3.1 Theoretical analysis of the optimization algorithm

The convergence of the Algorithm 1 is demonstrated in the following theorem:

Theorem 1 *The Algorithm 1 will monotonically increase the objective of the problem (6) in each iteration.*

Proof: According to the step 2 in Algorithm 1, for each iteration t we have

$$f(v^{t+1}) + \sum_i \alpha_i^t g_i(v^{t+1}) \geq f(v^t) + \sum_i \alpha_i^t g_i(v^t) \qquad (8)$$

For each i, note that $\alpha_i^t = sgn(g_i(v^t))$, so we have that $\left|g_i(v^{t+1})\right| = sgn(g_i(v^{t+1}))g_i(v^{t+1}) \geq sgn(g_i(v^t))g_i(v^{t+1}) = \alpha_i^t g_i(v^{t+1})$. Then $\left|g_i(v^{t+1})\right| \geq \alpha_i^t g_i(v^{t+1})$ and note that $|g_i(v^t)| - \alpha_i^t g_i(v^t) = 0$, we have:

$$\begin{aligned} &\left|g_i(v^{t+1})\right| \geq \alpha_i^t g_i(v^{t+1}) \\ &\Rightarrow \left|g_i(v^{t+1})\right| - \alpha_i^t g_i(v^{t+1}) \geq 0 \\ &\Rightarrow \left|g_i(v^{t+1})\right| - \alpha_i^t g_i(v^{t+1}) \geq \left|g_i(v^t)\right| - \alpha_i^t g_i(v^t) \qquad (9) \end{aligned}$$

Eq.(9) holds for every i, thus we have

$$\sum_i \left(\left|g_i(v^{t+1})\right| - \alpha_i^t g_i(v^{t+1})\right) \geq \sum_i \left(\left|g_i(v^t)\right| - \alpha_i^t g_i(v^t)\right) \qquad (10)$$

Combining Eq. (8) and Eq. (10), we arrive at

$$f(v^{t+1}) + \sum_i \left|g_i(v^{t+1})\right| \geq f(v^t) + \sum_i \left|g_i(v^t)\right| \qquad (11)$$

Thus the Algorithm 1 will monotonically increase the objective of the problem (6) in each iteration t. □

As the objective of the problem (6) has an upper bound, Theorem 1 indicates that the Algorithm 1 will converge. The following theorem shows that the solution in the convergence will satisfy the KKT condition.

Theorem 2 *The solution of the Algorithm 1 in the convergence will satisfy the KKT condition of the problem (6).*

Proof: The Lagrangian function of the problem (6) is

$$\mathcal{L}(v, \lambda) = f(v) + \sum_i |g_i(v)| - h(v, \lambda), \qquad (12)$$

where $h(\lambda, v)$ is the Lagrangian term to encode the constraint $v \in \mathcal{C}$ in problem (6).

Taking the derivative[1] of $\mathcal{L}(v, \lambda)$ w.r.t v, and setting the derivative to zero, we have:

$$\frac{\partial \mathcal{L}(v, \lambda)}{\partial v} = f'(v) + \sum_i \alpha_i g_i'(v) - \frac{\partial h(v, \lambda)}{\partial v} = 0, \qquad (13)$$

where $\alpha_i = sgn(g_i(v))$.

Suppose the Algorithm 1 converges to a solution v^*, from step 2 in Algorithm 1 we have

$$v^* = \arg\max_{v \in \mathcal{C}} f(v^*) + \sum_i \alpha_i^* g_i(v^*), \qquad (14)$$

where $\alpha_i^* = sgn(g_i(v^*))$. According to the KKT condition [Boyd and Vandenberghe, 2004] of the problem in Eq.(14), we know that the solution v^* satisfies Eq.(13), which is the KKT condition of the problem (6). □

In general, satisfying the KKT condition usually indicates that the solution is a local optimum solution. Theorem 2 indicates that the Algorithm 1 will usually converge to a local solution.

We can see that both the problem (5) and the problem (3) are the special cases of the problem (6), so we can use the proposed Algorithm 1 to solve these two problems. The key step of the Algorithm 1 is to solve the problem in step 2. In the next two sections, we give detailed derivation and algorithm to solve the problem (5) and the problem (3), respectively.

4 Principal component analysis with greedy L1-norm maximization revisited

Recall that the principal component analysis with greedy L1-norm maximization only need to solve the following problem:

$$\max_{w^T w = 1} \sum_{i=1}^{n} \left|w^T x_i\right|. \qquad (15)$$

As described in Section 2, an algorithm proposed in [Kwak, 2008] can solve it. In this section, we solve it based on the Algorithm 1, and compare the differences between these two algorithms. According to the Algorithm 1, the key step to solve the problem (15) is to solve the following problem:

$$\max_{w^T w = 1} \sum_{i=1}^{n} \alpha_i w^T x_i, \qquad (16)$$

where $\alpha_i = sgn((w^t)^T x_i)$. Denote $m = \sum_{i=1}^{n} \alpha_i x_i$, then we can rewrite the problem (16) as

$$\max_{w^T w = 1} w^T m, \qquad (17)$$

[1] When $x = 0$, 0 is a subgradient of function $|x|$, so $sgn(x)$ is the gradient or a subgradient of the function $|x|$ in all the cases.

The Lagrangian function of the above problem is

$$\mathcal{L}(w, \lambda) = w^T m - \lambda(w^T w - 1), \tag{18}$$

Taking the derivative of $\mathcal{L}(w, \lambda)$ w.r.t w, and setting the derivative to zero, we have $w = m/\lambda$. Then $\lambda = \|m\|_2$ according to the constraint $w^T w = 1$. So the optimal solution to the problem (16) is $w = m/\|m\|_2$.

Based on the Algorithm 1, the algorithm to solve the principal component analysis with greedy L1-norm maximization is described in Algorithm 2. We can see that the Algorithm 2 is almost the same as the one described in Section 2, except that the values of α_i are different when $(w_k^t)^T x_i = 0$ and the Algorithm 2 does not have the additional judgement when the algorithm converges. When $(w_k^t)^T x_i = 0$, $\alpha_i = 0$ in Algorithm 2 while $\alpha_i = 1$ in the algorithm proposed in [Kwak, 2008]. Using the Algorithm 2 without the additional judgement, we can also obtain a local solution according to Theorem 2.

From Algorithm 2 we can see that the algorithm is efficient and only involves matrix-vector multiplication. The computational complexity is $O(ndmt)$, where n, d, m is the number of data, dimension of original data and the dimension of the projected data respectively, and t is the iterative number. In practice, the algorithm usually converges in ten iterations. Therefore, the computational complexity of the algorithm is linear w.r.t both data number and data dimension, which indicates the algorithm is applicable in the case that both data number and data dimension are very high. If the data are sparse, the computational complexity is further reduced to $O(nsmt)$, where s is the averaged number of non-zeros elements in a data point.

```
Input: $X$, $m$, where $X$ is centralized
Initialize $W = [w_1, w_2, ..., w_m] \in \mathbb{R}^{d \times m}$ such that $W^T W = I$ ;
for $k = 1$ to $m$ do
    Let $w_k^1 = w_k, t = 1$ ;
    while not converge do
        1. $\alpha_i = sgn((w_k^t)^T x_i)$ ;
        2. $m = \sum_{i=1}^{n} \alpha_i x_i$, and $w_k^{t+1} = m/\|m\|_2$ ;
        3. $t = t + 1$ ;
    end
    Let $X = X - w_k^t (w_k^t)^T X$ and $w_k = w_k^t$ ;
end
Output: $W \in \mathbb{R}^{d \times m}$.
```

Algorithm 2: Principal component analysis with greedy L1-norm maximization.

5 Principal component analysis with non-greedy L1-norm maximization

The original problem in [Kwak, 2008] is to solve the following problem:

$$\max_{W^T W = I} \sum_{i=1}^{n} \left\| W^T x_i \right\|_1. \tag{19}$$

Since directly solving this problem is difficult, [Kwak, 2008] turns to solve it by a greedy method. In this paper, we propose a non-greedy method to directly solve the problem (19).

Based on the Algorithm 1, the key step to solve the problem (19) is to solve the following problem:

$$\max_{W^T W = I} \sum_{i=1}^{n} \alpha_i^T W^T x_i \tag{20}$$

where the vectors $\alpha_i = sgn((W^t)^T x_i)$. Denote $M = \sum_{i=1}^{n} x_i \alpha_i^T$, then we can rewrite the problem (20) as

$$\max_{W^T W = I} Tr(W^T M) \tag{21}$$

Suppose the SVD of M is $M = U \Lambda V^T$, where $U \in \mathbb{R}^{d \times d}$, $\Lambda \in \mathbb{R}^{d \times m}$ and $V \in \mathbb{R}^{m \times m}$, then we have:

$$\begin{aligned} Tr(W^T M) &= Tr(W^T U \Lambda V^T) \\ &= Tr(\Lambda V^T W^T U) \\ &= Tr(\Lambda Z) = \sum_i \lambda_{ii} z_{ii} \end{aligned} \tag{22}$$

where $Z = V^T W^T U \in \mathbb{R}^{m \times d}$, λ_{ii} and z_{ii} are the (i, i)-th element of matrix λ and Z respectively.

Note that $ZZ^T = I$, where I is an m by m identity matrix, so $z_{ii} \leq 1$. On the other hand, $\lambda_{ii} \geq 0$ since λ_{ii} is singular value of M. Therefore, $Tr(W^T M) = \sum_i \lambda_{ii} z_{ii} \leq \sum_i \lambda_{ii}$, and when $z_{ii} = 1 (1 \leq i \leq m)$, the equality holds. That is to say, $Tr(W^T M)$ reaches the maximum when $Z = [I, \mathbf{0}]$. Recall that $Z = V^T W^T U$, thus the optimal solution to the problem Eq. (21) is

$$W = U Z^T V^T = U[I; \mathbf{0}] V^T. \tag{23}$$

```
Input: $X$, $m$, where $X$ is centralized
Initialize $W^1 \in \mathbb{R}^{d \times m}$ such that $W^T W = I, t = 1$ ;
while not converge do
    1. $\alpha_i = sgn((W^t)^T x_i)$, $M = \sum_{i=1}^{n} x_i \alpha_i^T$ ;
    2. Calculate the SVD of $M$ as $M = U \Lambda V^T$, Let $W^{t+1} = U[I; \mathbf{0}] V^T$ ;
    3. $t = t + 1$ ;
end
Output: $W^t \in \mathbb{R}^{d \times m}$.
```

Algorithm 3: Principal component analysis with non-greedy L1-norm maximization.

Based on the Algorithm 1, the algorithm to solve the principal component analysis with non-greedy L1-norm maximization is described in Algorithm 3. According to Theorem 2, we can usually obtain a local solution.

From Algorithm 2 we can see that the algorithm is also efficient. Note that $n \gg m$ in practice, thus the computational complexity of the algorithm is $O(ndmt)$, which is the same as that of the greedy method. Similarly, the algorithm usually converges in ten iterations in practice. Therefore, the computational complexity of the algorithm is also linear w.r.t both

Table 1: Dataset Descriptions.

Data set	Size	Dimensions	Classes
Jaffe	213	1024	10
Umist	575	644	20
Yale	165	3456	15
Coil20	1440	1024	20
Palm	2000	256	100
USPS	9298	256	10

data number and data dimension, which indicates the algorithm is applicable in the case that both data number and data dimension are very high. If the data are sparse, the computational complexity is further reduced to $O(nsmt)$.

5.1 Extension to kernelization and tensorization

Similar to traditional PCA, the robust principal component analysis with L1-norm maximization is also a linear method, and is difficult to handle data well with non-Gaussian distribution. A popular technique to deal with this problem is extending the linear method to kernel method. Obviously, the robust principal component analysis with L1-norm maximization is invariant to rotation and shift, so this linear method satisfies the conditions in a general kernelization framework in [Zhang *et al.*, 2010], and thus can be kernelized using the framework. Specifically, the given data are transformed by KPCA [Schölkopf *et al.*, 1998], and then perform Algorithm 3 using the transformed data as input.

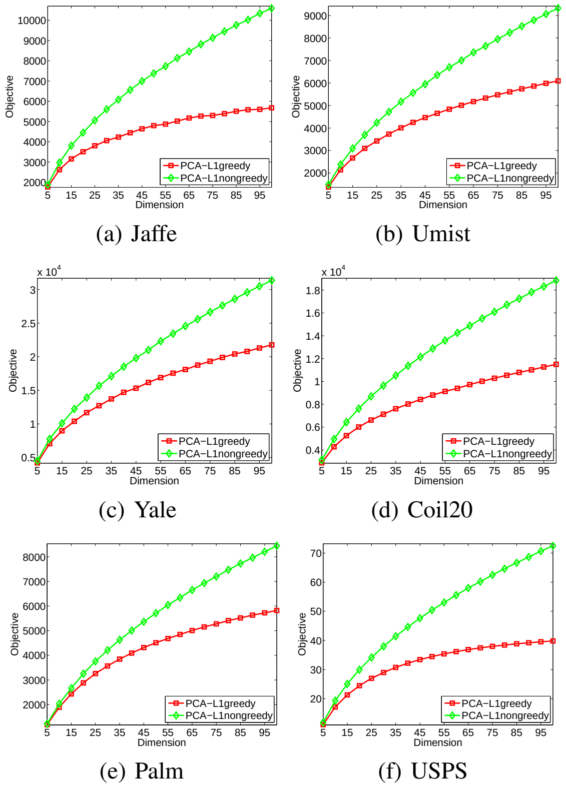
Another problem of the principal component analysis is that the method can only handle vector data. For 2D tensor or higher order tensor data, we have to vectorize the data to very high-dimensional vectors in order to apply this method. This approach will destroy the spacial information of tensor data and make the computational burden very heavy. A popular technique to deal with this problem is extending the vector method to tensor method. As the problem (19) of the principal component analysis with L1-norm maximization only includes linear operator W^Tx_i, it can be easily extended to the tensor method to handle tensor data directly. For simplicity, we only briefly discuss the case of 2D tensor, high order tensor cases can be readily extended by replacing the linear operator W^Tx_i with tensor operator [Lathauwer, 1997].

Suppose the given data are $X = [X_1, X_2, ..., X_n] \in \mathbb{R}^{r\times c\times n}$, where each data $X_i \in \mathbb{R}^{r\times c}$ is a 2D tensor, n is the number of data points. Similarly, we assume that $\{X_i\}_{i=1}^n$ are centralized, i.e., $\sum_{i=1}^n X_i = \mathbf{0}$.

In the 2D tensor case, linear operator W^Tx_i is replaced by U^TX_iV, where $U \in \mathbb{R}^{r\times r_1}$ and $V \in \mathbb{R}^{c\times c_1}$ are two projection matrices. Correspondingly, the problem (19) becomes:

$$\max_{U^TU=I_{r_1},V^TV=I_{c_1}} \sum_{i=1}^{n} \left\|U^TX_iV\right\|_1 \qquad (24)$$

As in other tensor method, problem (24) can be solved by alternative optimization technique (also named block coordinate descent). Specifically, when fixing U, the problem (24) reduced to the problem (19), and thus the V can be optimized by Algorithm 3. Similarly, U can also be optimized by Algorithm 3 when fixing V. The procedure is iteratively performed until converges.



(a) Jaffe (b) Umist

(c) Yale (d) Coil20

(e) Palm (f) USPS

Figure 1: Objective values in Eq.(3) with different dimensions obtained by PCA-L1greedy and PCA-L1nongreedy, respectively.

6 Experiments

In this section, we present experiments to demonstrate the effectiveness of the proposed principal component analysis with non-greedy L1-norm maximization (denoted by PCA-L1nongreedy) compared to the greedy method (denoted by PCA-L1greedy).

We use six image datasets from different domains to perform the experiments. A brief description of the datasets are shown in Table 1. In this experiment, we study the greedy and non-greedy optimization methods, and compare the objective values in Eq.(3) obtained by these two optimization methods.

In the first experiment, we run the greedy method and the non-greedy method with different projected dimensions m and the same initialization on each dataset. The projected dimensions varies from 5 to 100 with the interval 5. The results are shown in Figure 1. In the second experiment, we run the greedy method and the non-greedy method 50 times with the projected dimensions $m = 50$ on each dataset. In each time, the two methods use the same initialization. The results are shown in Table 2.

From Figure 1 and Table 2 we can see, the proposed non-greedy method obtains much higher objective values than that of the greedy method in all the cases. The results indicate that

Table 2: Objective values in Eq.(3) with dimension 50 obtained by PCA-L1greedy and PCA-L1nongreedy, respectively. The number of initialization is 50.

Data set	PCA-L1greedy				PCA-L1nongreedy			
	Min	Max	Min/Max	Mean	Min	Max	Min/Max	Mean
Jaffe	4722.86	4815.28	0.9808	4770.50	7349.87	7409.23	0.9920	7377.96
Umist	4649.03	4673.87	0.9947	4661.83	6316.97	6359.19	0.9934	6340.59
Yale	16058.68	16261.68	0.9875	16144.79	20964.16	21217.98	0.9880	21064.38
Coil20	8753.97	8793.97	0.9955	8778.63	12860.50	12935.98	0.9942	12891.44
Palm	4497.48	4518.04	0.9954	4507.50	5702.38	5724.27	0.9962	5712.15
USPS	34.44	34.47	0.9992	34.45	50.26	50.49	0.9954	50.39

the proposed non-greedy method always obtains much better solution to the L1-norm maximization problem (3) than the pervious greedy method.

7 Conclusions

A robust principal component analysis with non-greedy L1-norm maximization is proposed in this paper. We first propose an efficient optimization algorithm to solve a general L1-norm maximization problem, and the algorithm will usually converge to a local solution by theoretical analysis. Based on the algorithm, we directly solve the L1-norm maximization problem where the projection directions are optimized simultaneously. Similarly to the previous greedy method, the robust principal component analysis with non-greedy L1-norm maximization is also efficient, and is easy to extend to its kernel version or tensor version. Experimental results on six real world image datasets show that the proposed non-greedy method always obtains much better solution than that of the greedy method.

References

[Aanas *et al.*, 2002] H. Aanas, R. Fisker, K. Astrom, and J.M. Carstensen. Robust factorization. *IEEE Transactions on PAMI*, 24(9):1215–1225, 2002.

[Baccini *et al.*, 1996] A. Baccini, P. Besse, and A. de Faguerolles. A L1-norm PCA and heuristic approach. In *Proceedings of the International Conference on Ordinal and Symbolic Data Analysis*, volume 1, pages 359–368, 1996.

[Boyd and Vandenberghe, 2004] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[De La Torre and Black, 2003] F. De La Torre and M.J. Black. A framework for robust subspace learning. *International Journal of Computer Vision*, 54(1):117–142, 2003.

[Ding *et al.*, 2006] Chris H. Q. Ding, Ding Zhou, Xiaofeng He, and Hongyuan Zha. R1-PCA: rotational invariant L1-norm principal component analysis for robust subspace factorization. In *ICML*, pages 281–288, 2006.

[Donoho, 2006] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.

[Galpin and Hawkins, 1987] J.S. Galpin and D.M. Hawkins. Methods of L1 estimation of a covariance matrix. *Computational Statistics & Data Analysis*, 5(4):305–319, 1987.

[Huang and Ding, 2008] Heng Huang and Chris H. Q. Ding. Robust tensor factorization using r1 norm. In *CVPR*, 2008.

[Ke and Kanade, 2005] Q. Ke and T. Kanade. Robust L1 norm factorization in the presence of outliers and missing data by alternative convex programming. In *CVPR*, pages 739–746, 2005.

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

[Lathauwer, 1997] Lieven De Lathauwer. *Signal Processing based on Multilinear Algebra*. PhD thesis, Faculteit der Toegepaste Wetenschappen. Katholieke Universiteit Leuven, 1997.

[Li *et al.*, 2010] Xuelong Li, Yanwei Pang, and Yuan Yuan. L1-norm-based 2DPCA. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(4), 2010.

[Liu *et al.*, 2010] Yang Liu, Yan Liu, and Keith C. C. Chan. Multilinear maximum distance embedding via l1-norm optimization. In *AAAI*, 2010.

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

[Nie *et al.*, 2010b] Feiping Nie, Dong Xu, Ivor Wai-Hung Tsang, and Changshui Zhang. Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932, 2010.

[Pang *et al.*, 2010] Yanwei Pang, Xuelong Li, and Yuan Yuan. Robust tensor analysis with L1-norm. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(2):172–178, 2010.

[Schölkopf *et al.*, 1998] Bernhard Schölkopf, Alex J. Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

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

[Xiang *et al.*, 2008] Shiming Xiang, Feiping Nie, and Changshui Zhang. Learning a mahalanobis distance metric for data clustering and classification. *Pattern Recognition*, 41(12):3600–3612, 2008.

[Yang *et al.*, 2009] Yi Yang, Yueting Zhuang, Dong Xu, Yunhe Pan, Dacheng Tao, and Stephen J. Maybank. Retrieval based interactive cartoon synthesis via unsupervised bi-distance metric learning. In *ACM Multimedia*, pages 311–320, 2009.

[Zhang *et al.*, 2010] Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel PCA. *Neurocomputing*, 73(4-6):959–967, 2010.