GLOBALLY CONVERGENT ACCELERATED PROXIMAL ALTERNATING MAXIMIZATION METHOD FOR L1-PRINCIPAL COMPONENT ANALYSIS

Peng Wang  Huikang Liu  Anthony Man-Cho So

Department of Systems Engineering and Engineering Management, CUHK, Hong Kong

ABSTRACT

In this paper, we consider a $\ell_1$-PCA problem under the large-scale data sample scenario, which has extensive applications in science and engineering. Previous algorithms for the problem either are not scalable or do not have good convergence guarantees. Our contribution is threefold. First, we develop a novel accelerated version of the proximal alternating maximization method to solve the $\ell_1$-PCA problem. Second, by exploiting the Kurdyka-Łojasiewicz property of the problem, we show that our proposed method enjoys global convergence to a critical point, which improves upon existing convergence guarantees of other first-order methods for the $\ell_1$-PCA problem. Third, we demonstrate via numerical experiments on both real-world and synthetic datasets that our proposed method is scalable and more efficient and accurate than other methods in the literature.

Index Terms— $\ell_1$-PCA, extrapolation, accelerated proximal alternating maximization, Kurdyka-Łojasiewicz inequality, global convergence

1. INTRODUCTION

Principal Component Analysis (PCA) is a fundamental data analytic tool that has found many applications in various areas of science and engineering [8]. Roughly speaking, PCA aims to find a non-trivial lower-dimensional subspace that can explain most of the variance in the data, and a common formulation is given by

$$\max_{B \in \mathbb{R}^{D \times K}} \|X^T B\|_F^2 \quad \text{s.t.} \quad B^T B = I, \quad (1.1)$$

where $X \in \mathbb{R}^{D \times N}$ is the data matrix consisting of $N$ data samples $x_1, \ldots, x_N \in \mathbb{R}^D$ and $\|A\|_F = \left(\sum_{i,j} |A_{i,j}|^2\right)^{1/2}$ denotes the Frobenius norm of the matrix $A$. Problem (1.1), which we shall refer to as the $\ell_2$-PCA, can be solved by performing a singular value decomposition (SVD) of the data matrix $X$ [7]. Unfortunately, $\ell_2$-PCA is sensitive to the outliers in the data and can lead to a poor performance in the presence of outliers. In practice, the outliers can arise as a result of transmission error, bursty-noise effect, etc. This motivates the development of variants of PCA that are robust against outliers. One such variant is the $\ell_1$-PCA, which was proposed and studied in, e.g., [3][6][20][9]. Instead of maximizing the Frobenius norm of $X^T B$, $\ell_1$-PCA seeks to maximize its $\ell_1$-norm; i.e.,

$$\max_{B \in \mathbb{R}^{D \times K}} \|X^T B\|_1 \quad \text{s.t.} \quad B^T B = I, \quad (1.2)$$

where $\|A\|_1 = \sum_{i,j} |A_{i,j}|$ denotes the $\ell_1$-norm of the matrix $A$. Unlike the $\ell_2$-PCA problem (1.1), which admits an essentially closed-form solution that can be computed in polynomial time, the $\ell_1$-PCA problem (1.2) is nonsmooth nonconvex and is NP-hard in general [15]. As such, various heuristics have been proposed to tackle Problem (1.2). Kwak [10] presented a first-order method with fixed-point iterations to solve (1.2) when $K = 1$. Subsequently, Nie et al. [19] extended the above first-order method to handle the case where $K \geq 2$. In both works, the authors showed that the iterates generated by their respective algorithms have a convergent subsequence (i.e., the so-called subsequential convergence property), and that any limit point of the iterates is a critical point of Problem (1.2). Recently, Markopoulos et al. [15] proposed an exact algorithm for solving (1.2) with complexity $O(N^{DK-K+1})$. Note that the algorithm has polynomial-time complexity when $D$ and $K$ are fixed. However, in practice, it can only handle a small number of samples (i.e., $N$ is small) and low-dimensional data (i.e., $D \ll N$). Later, Markopoulos et al. [16] proposed a sub-optimal algorithm based on the notion of bit-flipping for tackling Problem (1.2). They showed that their proposed algorithm also enjoys subsequential convergence, and that the set of limit points is a subset of those of the algorithms in [10][19]. It is worth noting that subsequential convergence is a strictly weaker property than convergence—the latter refers to the property that the sequence of iterates converge to a single point, while the former allows for the possibility that the sequence has multiple subsequences that converge to different points, which makes it numerically difficult to identify which point is a limit point of the algorithm.

In practice, lightweight methods such as first-order methods are often preferred when solving the $\ell_1$-PCA problem (1.2), as the number of data samples $N$ and the data dimension $D$ are often large. In view of the above discussion, we are thus motivated to develop fast first-order methods for tackling (1.2) with strong convergence guarantees. In
this work, we propose a novel accelerated version of the proximal alternating maximization (PAM) method in [1] for solving Problem (1.2). Our method, which we call accelerated PAM (APAM), performs a linear extrapolation on one block-variable of the $\ell_1$-PCA formulation, which can not only accelerate the original PAM method but also return solution with better quality than existing methods empirically. It is worth mentioning that such acceleration technique is mainly developed for methods that solve convex optimization problems in the literature, such as heavy-ball method [21, 23], Nesterov’s accelerated gradient descent [17, 18] and FISTA [4]. Recently, there have been some works that aim to extend this technique to methods that solve nonconvex optimization problems. In particular, Li and Lin [11] proposed monotone and nonmonotone accelerated proximal gradient methods to solve general nonsmooth nonconvex problems, while Li et al. [12] analyzed the convergence of the accelerated proximal gradient method with momentum for nonconvex problems. However, since the proximal operator $\partial f$ is not denoted by $A$, the indicator function of the set $A$, the methods in [11, 12] are ineffective for solving Problem (1.2). By contrast, our proposed (A)PAM method can take advantage of the structure of the $\ell_1$-PCA problem (1.2), thus resulting in extremely efficient computations. Moreover, by utilizing the so-called Kurdyka-Łojasiewicz (KL) property (see [1, 2]) of a carefully constructed Lyapunov function associated with Problem (1.2), we are able to show that the iterates generated by the (A)PAM method converge globally to a single critical point of (1.2). To the best of our knowledge, this is the first global convergence result for an accelerated version of the PAM method and improves upon the subsequential convergence results in [10, 19, 16] for the $\ell_1$-PCA problem (1.2).

Besides the notations introduced earlier, we use $1 \leq A \leq 1$ to denote $-1 \leq A_{ij} \leq 1, \forall i,j$ and $\delta A$ to denote the indicator function of the set $A$; i.e., $\delta A(A)$ is 0 if $A \in A$ and $\infty$ otherwise. Furthermore, the subdifferential of a proper lower semicontinuous function $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is denoted by $\partial f$; see, e.g., [22, Chapter 8].

2. ACCELERATED PAM FOR $\ell_1$-PCA

Our proposed (A)PAM method for the $\ell_1$-PCA problem is based on the following two-block reformulation of Problem (1.2):

$$\max_{B^k \in B} \|X^T B\|_1 = \max_{A \in A, B \in B} H(A, B),$$  

(2.1)

where $H(A, B) := \text{tr}(A^T X^T B)$, $A := \{A \in \mathbb{R}^{N \times K} : -1 \leq A \leq 1\}$, and $B := \{B \in \mathbb{R}^{D \times K} : B^T B = I\}$. Specifically, given an initial point $(A^0, B^0)$, our (A)PAM method generates the sequence $\{(A^k, B^k)\}$, $k = 0, 1, 2, \ldots$, via the scheme

$$A^{k+1} = \arg \max_{A \in A} \left\{ H(A, Y^k) - \frac{1}{2\alpha} \|A - A^k\|^2_F \right\},$$  

(2.2)

and $B_{k+1} = \arg \max_{B \in B} \left\{ H(A^{k+1}, B) - \frac{1}{2\beta} \|B - B^k\|^2_F \right\},$  

(2.3)

where $Y^k = B^k + \theta(B^k - B^{k-1})$ and $\theta \in [0, 1]$ when $k \geq 1$ and $Y^0 = B^0$. The novelty here is that we perform an extrapolation step for the variable $B$ by adding a momentum term that involves the previous and current iterates. When $\theta = 0$, we recover the PAM updates in [1]. The upshot of the updates (2.2), (2.3) is that they both have closed-form solutions. Indeed, it can be easily verified that

$$A^{k+1} = \Pi_A \left[ A^k + \alpha X^T Y^k \right],$$  

(2.4)

where $\Pi_A$ is the projector onto the box $A$; i.e.,

$$\Pi_A(D) = \begin{cases} -1 & \text{if } D_{ij} < -1, \\ D_{ij} & \text{if } -1 \leq D_{ij} \leq 1, \\ 1 & \text{if } D_{ij} > 1. \end{cases}$$

On the other hand, we have

$$B^{k+1} = U^{k+1} V^{k+1}^T,$$

(2.5)

where $U^{k+1} + \beta X A^{k+1} = U^{k+1} A^{k+1} V^{k+1}^T$ is a compact SVD of $B^{k+1} + \beta X A^{k+1}$. Indeed, it can be easily verified that dates (2.2) (2.3) is that they both have closed-form solutions.

Algorithm 1 Accelerated Proximal Alternating Maximization Method (APAM) for $\ell_1$-PCA

1: Input: Step sizes $\alpha > 0, \beta > 0$; extrapolation parameter $0 \leq 1$; initial points $A^0 \in A, Y^0 = B^0 \in B$
2: for $k = 0, 1, 2, \ldots$ do
3: $A^{k+1} = \Pi_A \left[ A^k + \alpha X^T Y^k \right]$  
4: $B^{k+1} = U^{k+1} V^{k+1}^T$, where $(U^{k+1}, V^{k+1})$ is given by (2.5)
5: $Y^{k+1} = B^{k+1} + \theta(B^{k+1} - B^k)$
6: end for

3. CONVERGENCE ANALYSIS OF APAM

To facilitate our convergence analysis of the APAM method, let us rewrite (2.1) as the following unconstrained minimization problem:

$$\min_{A \in \mathbb{R}^{N \times K}, B \in \mathbb{R}^{D \times K}} \Psi(A, B),$$  

(3.1)

where $\Psi(A, B) := -H(A, B) + \delta_A(A) + \delta_B(B)$. The upshot of the form (3.1) is that it allows us to apply the powerful machinery in [1] (particularly [2, Theorem 2.9]) to analyze the convergence behavior of the APAM method. To begin, let
us construct the following Lyapunov function associated with Problem (3.1):

\[ \Theta(A, B, B') = \Psi(A, B) + \frac{\gamma}{2} \| B - B' \|_F^2, \gamma \in [L, \frac{1}{\beta}], \]

(3.2)

where \( L = \| X \| \) is the spectral norm of data matrix \( X \). For simplicity, let us write \( C^k := (A^k, B^k, B'^{-1}) \), \( \forall k = 1, 2, \ldots \). We then have the following preparatory results:

**Lemma 3.1 (Sufficient Decrease of \( \Theta \)).** Let \( \{C^k\}_{k \geq 0} \) be the sequence generated by Algorithm 1 where the step size \( \alpha < \frac{1}{L\beta} \) is a constant and the penalty parameter \( \gamma \) satisfies \( \gamma \in [L\theta, \frac{1}{\beta}] \). Then, the sequence \( \{\Theta(C^k)\}_{k \geq 0} \) satisfies

\[ \Theta(C^{k+1}) - \Theta(C^k) \leq -\rho_1 \| C^{k+1} - C^k \|_F^2, \]

for some constant \( \rho_1 > 0 \).

**Lemma 3.2 (Safeguard).** The sequence \( \{C^k\}_{k \geq 0} \) is bounded and \( W^{k+1} \in \partial \Theta(C^{k+1}) \), where

\[ W^{k+1} := \left( -X^T \Delta^{k+1} + \theta(X^T \Delta^k + \frac{1}{\alpha} (A^k - A^{k+1}), \right) \]

\[ \left( \gamma - \frac{1}{\beta} \right) \Delta^k - \gamma \Delta^k \]

and \( \Delta^k := B^k - B^{k-1} \). Furthermore, there exists a constant \( \rho_2 > 0 \) such that

\[ \| W^{k+1} \|_F \leq \rho_2 \| C^{k+1} - C^k \|_F. \]

Due to space limitation, we defer the proofs of Lemmas 3.1 and 3.2 to the full version of this paper.

**Lemma 3.3 (Relationship between \( \Psi \) and \( \Theta \)).** Let \( A \in A \) and \( B \in B \). Then,

\[ 0 \in \partial \Psi(A, B) \iff 0 \in \partial \Theta(A, B, B). \]

Lemma 3.3 can be easily verified using the properties of the subdifferential; see, e.g., [22] Chapter 8).

**Lemma 3.4 (KL Property of the Lyapunov Function).** The Lyapunov function \( \Theta \) has the KL property (see [2] Definition 2.4) for the definition).

Lemma 3.4 follows from the fact that \( \Theta(A, B, B') \) consists of polynomials (i.e., \(-H(A, B) + \frac{\gamma}{2} \| B - B' \|_F^2 \)) and indicator functions of the polyhedral set \( A \) and the smooth manifold \( B \); see, e.g., [1][2].

Armed with the above lemmas and invoking [2] Theorem 2.9], we have the following global convergence result for the APAM method.

**Theorem 3.5.** For any initialization \( A^0 \in A, B^0 \in B \), the sequence generated by the APAM method (Algorithm 1) will converge to a critical point of the \( \ell_1 \)-PCA problem (1.2).

Theorem 3.5 establishes, for the first time, the global convergence of an accelerated version of the PAM method in [1]. This is achieved by a novel construction of the Lyapunov function [3.2] associated with Problem (3.1).

### 4. NUMERICAL RESULTS

In this section, we demonstrate the superior of the APAM method for solving the \( \ell_1 \)-PCA problem (1.2) via numerical experiments. Our codes are implemented in MATLAB R2018a. The tests are conducted on a standard computing server with 100GB memory and 4 Intel(R) Xeon(R) CPUs each consisting 10 cores with 2.40GHz.

#### 4.1. Performance of Different First-Order Methods

In this subsection, we compare our APAM method with the PAM method in [1] and the non-greedy first-order fixed-point method (NFMM) in [19]. We utilize the real-world datasets epsilon, mnist downloaded from LIBSVM [5]. We employ the following stopping criterion: At each iteration \( k \), we compute the function value \( f^k \) and terminate the algorithm if there are 10 consecutive iterations such that

\[ |f^{k+m} - f^{k+m-1}| \leq \epsilon, \quad m = 1, \ldots, 10, \]

where we choose \( \epsilon = 10^{-7} \).

We implement two sets of experiments on different subsets of the datasets epsilon, mnist to compare the runtimes and objective values of APAM, PAM, and NFMM, where epsilon is a size 100000 \times 500 subset of the original one. Specifically, we set the size of subsets of the full data as 10\%, 20\%, \ldots, 100\% in each test. The step sizes for APAM and PAM are both set to \( \alpha = 10^{\epsilon} \), \( \beta = 1 \). To avoid the influence of the starting points and other random factors, we run each algorithm 10 times from different starting points. To be consistent, different algorithms have the same starting point in each test. Finally, for each algorithm in a test, we select the maximum among the 10 obtained objective values as the recorded value. Furthermore, we average the runtimes of the 10 runs as the recorded time. To compare the suboptimal value of each algorithm, we will use the value of APAM as the baseline, and compute the ratio of improvement, i.e., others’ recorded value minus APAM’s divided by APAM’s.

As seen from Fig. 1, APAM is the most efficient and accurate algorithm when compared to other two first-order algorithms, as it consumes much less time and converges to a better solution than PAM and NFMM.

#### 4.2. Computation Accuracy of Different Methods

In this subsection, we investigate how far the objective values returned by APAM, PAM, NFMM, and the bit-flipping method (BFM) in [16] are from the global optimal value of the \( \ell_1 \)-PCA problem (1.2). Note that BFM is a high-order method. To do so, we make use of the exact algorithm proposed in [15] to check the optimality of the aforementioned methods. It should be noted that the algorithm in [15] has a polynomial

\[ \frac{1}{2} \text{We set } \theta = 1 \text{ unless specified otherwise.} \]

\[ \frac{1}{2} \text{https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/} \]
complexity of $O(N^D)$ when $D$ is fixed and $D \ll N$. We set the step sizes for APAM and PAM to $\alpha = 10$, $\beta = 10$. Here, we define the relative gap between an obtained objective value and the global optimal value as

$$\Delta(B, B^*) = \frac{\|X^T B\|_1 - \|X^T B\|_1}{\|X^T B\|_1}, \quad (4.1)$$

where $B^*$ is an optimal solution to Problem (1.2) and $B$ is a solution obtained by any of the aforementioned methods. The quantity $\Delta(B, B^*)$ is called the performance degradation ratio in [16]. We generate 1000 data matrices $\{X_1, \ldots, X_{1000}\}$ randomly with $N = 20$, $D = 3$, and $K = 2$. Based on 1000 tests, we calculate the empirical cumulative distribution functions (CDFs) of the performance degradation ratios of the algorithms and plot them in Fig. 2.

According to the above experiments, when the number of initialization is 5 or 15, BFM can return the optimal solution 92% and 97% of the time, respectively, while APAM can return the optimal solution about 84% and 96%, respectively. These numbers are much better than PAM and NFFM. Since APAM is a first-order method and BFM is a local exhaustive search procedure, it is normal that BFM has a better performance than APAM in terms of performance degradation ratio.

4.3. Convergence Performance of APAM with Different Extrapolation Step Size

In this subsection, we demonstrate empirically that APAM is superior to PAM in both runtime and convergence performance. Towards that end, we run APAM with $\theta = 0.25, 0.5, 0.75, 1$ and PAM (which corresponds to APAM with $\theta = 0$) on different subsets of the real-world dataset protein downloaded from LIBSVM under the same setting as that in Section 4.1. We then compare the runtimes and function values of these five settings of $\theta$ in APAM. As is obvious from Fig. 3, the extrapolation step greatly accelerates the algorithm, with $\theta = 1$ being the most efficient and accurate setting among the five.

5. CONCLUSIONS

In this paper, we proposed an efficient and high-accuracy first-order method called APAM to solve the $\ell_1$-PCA problem and established its global convergence to a critical point by exploiting the KL-property of the problem. Moreover, our experimental results showed that the APAM method has great potential when compared with existing algorithms in terms of both computational efficiency and accuracy. An interesting future direction would be to determine the convergence rate of our proposed APAM method by further exploiting the KL property; see [11][12][14][13] for some related works.
6. REFERENCES


