• RESEARCH PAPER •

A generalized power iteration method for solving quadratic problem on the Stiefel manifold

Feiping NIE[1*], Rui ZHANG[1*] & Xuelong LI[2]

[1]*School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL),*
Northwestern Polytechnical University, Xi'an 710072, China;
[2]*Center for OPTical IMagery Analysis and Learning (OPTIMAL),*
State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics,
Chinese Academy of Sciences, Xi'an 710119, China

Abstract   In this paper, we first propose a novel generalized power iteration (GPI) method to solve the quadratic problem on the Stiefel manifold (QPSM) as $\min_{W^{\mathrm{T}}W=I} \mathrm{Tr}(W^{\mathrm{T}}AW - 2W^{\mathrm{T}}B)$ along with the theoretical analysis. Accordingly, its special case known as the orthogonal least square regression (OLSR) is under further investigation. Based on the aforementioned studies, we then majorly focus on solving the unbalanced orthogonal procrustes problem (UOPP). As a result, not only a general convergent algorithm is derived theoretically but the efficiency of the proposed approach is verified empirically as well.

Keywords   quadratic problem, Stiefel manifold, power iteration, procrustes problem, orthogonal least square regression

1   Introduction

The orthogonal procrustes problem (OPP) is the least square problem on the Stiefel manifold. The OPP originates from the factor analysis in psychometrics during 1950s and 1960s [1, 2]. The major purpose is to determine an orthogonal matrix that rotates the factor matrix to best fit some hypothesis matrix. The balanced case of the OPP was surveyed in multiple introductory textbooks such as [3, 4].

Recently, due to the wide applications of the orthogonal regression in computer science, see [5, 6], solving the unbalanced OPP (UOPP) is under increasing concern. Multiple approaches are proposed to solve UOPP such as the expansion balanced algorithm (EB), the right hand side and the left hand side relaxation (RSR), (LSR), the successive projection (SP) and the Lagrangian relaxation (LR). In [7], the EB method employs the expanded balanced OPP as its objective function. In [8, 9] respectively, the RSR and the LSR approaches update the solution row by row or column by column iteratively based on solving the least square regression with a quadratic equality constraint (LSQE). In [10], the SP method updates the solution column by column by virtue of the projection method combined with correction

---

* Corresponding author (email: feipingnie@gmail.com, ruizhang8633@gmail.com)

techniques (PMCT) discussed by [11], which is efficient to solve LSQE. In [12], the LR method solves UOPP by selecting different Lagrangian multipliers.

All the approaches mentioned above could converge to the solution of UOPP successfully, whereas they deal with more complex procedures, which represent high orders of complexity. Furthermore, all these methods initialize the parameters deliberately to optimize their proposed algorithms. Last but not least, all these approaches are unable to deal with a more general problem known as the quadratic problem on the Stiefel manifold (QPSM).

To address the referred deficiencies, we derive a novel generalized power iteration method (GPI) for QPSM in order to efficiently solve the orthogonal least square regression (OLSR) and UOPP with a random initial guess and concise computational steps. In sum, the proposed GPI method can deal with a more general problem known as QPSM than other approaches. Furthermore, the experimental results show that the proposed GPI method not only takes much less CPU time for the convergence but becomes more efficient dealing with the data matrix of large dimension as well.

Notations. For any matrix M, Frobenius norm is defined as $\|M\|_F^2 = \mathrm{Tr}(M^\mathrm{T}M)$, where $\mathrm{Tr}(\cdot)$ is the trace operator. For any positive integer n, I_n denotes a $n \times n$ identity matrix.

2 Power iteration method revisited

The power iteration method is an iterative algorithm to seek the dominant eigenvalue and the related eigenvector of any given symmetric matrix $A \in \mathbb{R}^{m \times m}$, where the dominant eigenvalue is defined as the greatest eigenvalue in magnitude. The power iteration can be performed as the following steps:

(1) Initialization. Random initialize a vector $w \in \mathbb{R}^{m \times 1}$, which has a nonzero component in the direction of the dominant eigenvector.

(2) Update $m \leftarrow Aw$.

(3) Calculate $q = \frac{m}{\|m\|_2}$.

(4) Update $w \leftarrow q$.

(5) Iteratively perform the steps (2)–(4) until convergence.

The power iteration could be further extended to the orthogonal iteration (also called subspace iteration or simultaneous iteration) method to find the first k ($k \leqslant m$) dominant eigenvalues and their associated eigenvectors for the given matrix A. The orthogonal iteration method could be described as the following iterative algorithm:

(1) Initialization. Random initialize $W \in \mathbb{R}^{m \times k}$.

(2) Update $M \leftarrow AW$.

(3) Calculate $QR = M$ via the compact QR factorization of M, where $Q \in \mathbb{R}^{m \times k}$ and $R \in \mathbb{R}^{k \times k}$.

(4) Update $W \leftarrow Q$.

(5) Iteratively perform the steps (2)–(4) until convergence.

Apparently, the orthogonal iteration method above indicates a normalization process, which is similar as the normalization in the power iteration method. When the matrix A is positive semi-definite (psd), the orthogonal iteration method is equivalent to solving the following optimization problem:

$$\max_{W^\mathrm{T}W=I_k} \mathrm{Tr}(W^\mathrm{T}AW). \tag{1}$$

Therefore, the orthogonal iteration method is equivalent to the following steps under the psd matrix A:

(1) Initialization. Random initialize $W \in \mathbb{R}^{m \times k}$.

(2) Update $M \leftarrow AW$.

(3) Calculate $USV^\mathrm{T} = M$ via the compact SVD method of M, where $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{k \times k}$.

(4) Update $W \leftarrow UV^\mathrm{T}$.

(5) Iteratively perform the steps (2)–(4) until convergence.

From the observation, the solution of the above algorithm as WK differs from the solution of the orthogonal iteration method as W by the form, where $KK^\mathrm{T} = I_k$. However, the difference between

the solutions of these two algorithms does not affect the objective value of the problem (1) due to the following derivation:

$$\text{Tr}((WK)^{\text{T}}AWK) = \text{Tr}(W^{\text{T}}AWKK^{\text{T}}) = \text{Tr}(W^{\text{T}}AW).$$

3 Quadratic problem on the Stiefel manifold

The Stiefel manifold $\nu_{m,k}$ is a set of the matrices $W \in \mathbb{R}^{m \times k}$, which have orthonormal columns as $\nu_{m,k} = \{W \in \mathbb{R}^{m \times k} : W^{\text{T}}W = I_k\}$.

In this section, a novel approach is derived to unravel the following QPSM [13] as

$$\min_{W^{\text{T}}W=I_k} \text{Tr}(W^{\text{T}}AW - 2W^{\text{T}}B), \tag{2}$$

where $W \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{m \times k}$ and the symmetric matrix $A \in \mathbb{R}^{m \times m}$. In order to solve the problem (2), QPSM in (2) can be further relaxed into

$$\max_{W^{\text{T}}W=I_k} \text{Tr}(W^{\text{T}}\tilde{A}W) + 2\text{Tr}(W^{\text{T}}B), \tag{3}$$

where $\tilde{A} = \alpha I_m - A \in \mathbb{R}^{m \times m}$. The relaxation parameter α is an arbitrary constant such that $\tilde{A}$ is a positive definite (pd) matrix. To be more specific, the relaxation parameter α could be easily set as the dominant eigenvalue of A, which could be fast obtained by the power method discussed in the previous section. Instead of the method of the Lagrangian multipliers to deal with an optimization problem with orthogonal constraints, one may use a geometric optimization algorithm tailored to the Stiefel manifold, such as, for example, the one surveyed in [14].

Accordingly, the Lagrangian function for the problem (3) can be written as

$$L_1(W, \Lambda) = \text{Tr}(W^{\text{T}}\tilde{A}W) + 2\text{Tr}(W^{\text{T}}B) - \text{Tr}(\Lambda(W^{\text{T}}W - I_k)). \tag{4}$$

From Eq. (4), we could obtain the KKT condition for the problem (3) as

$$\frac{\partial L_1}{\partial W} = 2\tilde{A}W + 2B - 2W\Lambda = 0, \tag{5}$$

which is difficult to solve directly. Thus, motivated by [15] and the power iteration method mentioned in Section 2, we could propose the following iterative algorithm:

(1) Initialization. Random initialize $W \in \mathbb{R}^{m \times k}$ such that $W^{\text{T}}W = I_k$.

(2) Update $M \in \mathbb{R}^{m \times k} \leftarrow 2\tilde{A}W + 2B$.

(3) Calculate W^* by solving the following problem:

$$\max_{W^{\text{T}}W=I_k} \text{Tr}(W^{\text{T}}M). \tag{6}$$

(4) Update $W \leftarrow W^*$.

(5) Iteratively perform the steps (2)–(4) until convergence.

Besides, a closed form solution of the problem (6) can be achieved by the following derivation.

Suppose the full SVD of M is $M = \mathbb{U}\Sigma\mathbb{V}^{\text{T}}$ with $\mathbb{U} \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times k}$ and $\mathbb{V} \in \mathbb{R}^{k \times k}$, then we have

$$\text{Tr}(W^{\text{T}}M) = \text{Tr}(W^{\text{T}}\mathbb{U}\Sigma\mathbb{V}^{\text{T}}) = \text{Tr}(\Sigma\mathbb{V}^{\text{T}}W^{\text{T}}\mathbb{U}) = \text{Tr}(\Sigma Z) = \sum_{i=1}^{k} \sigma_{ii}z_{ii},$$

where $Z = \mathbb{V}^{\text{T}}W^{\text{T}}\mathbb{U} \in \mathbb{R}^{k \times m}$ with z_{ii} and σ_{ii} being the (i,i)-th elements of the matrix Z and Σ, respectively.

Note that $ZZ^{\mathrm{T}} = I_k$, thus $|z_{ii}| \leqslant 1$. On the other hand, $\sigma_{ii} \geqslant 0$ since σ_{ii} is a singular value of the matrix M. Therefore, we have

$$\mathrm{Tr}(W^{\mathrm{T}}M) = \sum_{i=1}^{k} z_{ii}\sigma_{ii} \leqslant \sum_{i=1}^{k} \sigma_{ii}.$$

Apparently, the equality holds when $z_{ii} = 1$, $(1 \leqslant i \leqslant k)$. That is to say, $\mathrm{Tr}(W^{\mathrm{T}}M)$ reaches the maximum when the matrix $Z = [I_k, 0] \in \mathbb{R}^{k \times m}$. Recall that $Z = \mathbb{V}^{\mathrm{T}}W^{\mathrm{T}}\mathbb{U}$, thus the optimal solution to the problem (6) can be represented as

$$W = \mathbb{U}Z^{\mathrm{T}}\mathbb{V}^{\mathrm{T}} = \mathbb{U}[I_k; 0]\mathbb{V}^{\mathrm{T}}. \tag{7}$$

Since Eq. (7) is based upon the full SVD of the matrix M, Eq. (7) can be rewritten as $W = UV^{\mathrm{T}}$ via the compact SVD of the matrix M, where $M = USV^{\mathrm{T}}$ with $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{k \times k}$.

Based on the above analysis, the GPI can be summarized in Algorithm 1.

We will prove that the proposed Algorithm 1 converges monotonically to the local minimum of QPSM (2).

Algorithm 1 Generalized power iteration method (GPI)

Input: The symmetric matrix $A \in \mathbb{R}^{m \times m}$ and the matrix $B \in \mathbb{R}^{m \times k}$.
Initialize a random $W \in \mathbb{R}^{m \times k}$ satisfying $W^{\mathrm{T}}W = I_k$ and α via power method such that $\tilde{A} = \alpha I_m - A \in \mathbb{R}^{m \times m}$ is a positive definite matrix.
(1) Update $M \leftarrow 2\tilde{A}W + 2B$.
(2) Calculate $USV^{\mathrm{T}} = M$ via the compact SVD method of M where $U \in \mathbb{R}^{m \times k}$, $S \in \mathbb{R}^{k \times k}$ and $V \in \mathbb{R}^{k \times k}$.
(3) Update $W \leftarrow UV^{\mathrm{T}}$.
Iteratively perform the steps (1)–(3) until the algorithm converges.

Step (3) of Algorithm 1 is an instance of a class of methods, called manifold retractions, to update a matrix on the Stiefel manifold, that were discussed in details in [16].

Lemma 1. If the symmetric matrix $\tilde{A} \in \mathbb{R}^{m \times m}$ is positive definite (pd), then

$$\mathrm{Tr}(\tilde{W}^{\mathrm{T}}\tilde{A}\tilde{W}) - 2\mathrm{Tr}(\tilde{W}^{\mathrm{T}}\tilde{A}W) + \mathrm{Tr}(W^{\mathrm{T}}\tilde{A}W) \geqslant 0,$$

where $\tilde{W} \in \mathbb{R}^{m \times k}$ and $W \in \mathbb{R}^{m \times k}$ are arbitrary matrices.

Proof. Since the matrix $\tilde{A}$ is positive definite (pd), we could rewrite $\tilde{A} = L^{\mathrm{T}}L$ via Cholesky factorization. Therefore, we have the following proof for Lemma 1 as

$$\|L\tilde{W} - LW\|_F^2 \geqslant 0$$
$$\Rightarrow \mathrm{Tr}(\tilde{W}^{\mathrm{T}}\tilde{A}\tilde{W}) - 2\mathrm{Tr}(\tilde{W}^{\mathrm{T}}\tilde{A}W) + \mathrm{Tr}(W^{\mathrm{T}}\tilde{A}W) \geqslant 0.$$

Theorem 1. The Algorithm 1 decreases the value of the objective function in (2) monotonically in each iteration until it converges.

Proof. Suppose the updated W is $\tilde{W}$ in Algorithm 1, then we have

$$\mathrm{Tr}(\tilde{W}^{\mathrm{T}}M) \geqslant \mathrm{Tr}(W^{\mathrm{T}}M), \tag{8}$$

since $\tilde{W}$ is the optimal solution of the problem (6). Based on the fact that $M = 2\tilde{A}W + 2B$, Eq. (8) can be further illustrated as

$$2\mathrm{Tr}(\tilde{W}^{\mathrm{T}}\tilde{A}W) + 2\mathrm{Tr}(\tilde{W}^{\mathrm{T}}B) \geqslant 2\mathrm{Tr}(W^{\mathrm{T}}\tilde{A}W) + 2\mathrm{Tr}(W^{\mathrm{T}}B). \tag{9}$$

Based on Lemma 1 and Eq. (9), we could infer that

$$\mathrm{Tr}(\tilde{W}^{\mathrm{T}}\tilde{A}\tilde{W}) + 2\mathrm{Tr}(\tilde{W}^{\mathrm{T}}B) \geqslant \mathrm{Tr}(W^{\mathrm{T}}\tilde{A}W) + 2\mathrm{Tr}(W^{\mathrm{T}}B)$$
$$\Rightarrow \mathrm{Tr}(\tilde{W}^{\mathrm{T}}A\tilde{W}) - 2\mathrm{Tr}(\tilde{W}^{\mathrm{T}}B) \leqslant \mathrm{Tr}(W^{\mathrm{T}}AW) - 2\mathrm{Tr}(W^{\mathrm{T}}B),$$

which indicates that Algorithm 1 decreases the objective value of QPSM in (2) in each iteration until the algorithm converges.

Theorem 2. Algorithm 1 converges to a local minimum of the QPSM problem (2).

Proof. Since Algorithm 1 performs based on solving the problem (6) in each iteration, the Lagrangian function for the solution of Algorithm 1 can be represented as

$$L_2(W, \Lambda) = \text{Tr}(W^{\text{T}} M) - \text{Tr}(\Lambda(W^{\text{T}} W - I_k)). \tag{10}$$

Therefore, the solution of Algorithm 1 satisfies the following KKT condition:

$$\frac{\partial L_2}{\partial W} = M - 2W\Lambda = 0. \tag{11}$$

Generally speaking, the matrix M will be updated by $\tilde{W}$ in each iteration under Algorithm 1. Since Algorithm 1 converges to the optimal solution W, i.e., $\tilde{W} = W$ due to Theorem 1, Eq. (11) can be further formulated by substituting $M = 2\tilde{A}W + 2B$ as

$$\frac{\partial L_2}{\partial W} = 2\tilde{A}W + 2B - 2W\Lambda = 0. \tag{12}$$

By comparing (5) and (12), we could draw the conclusion that the solution of Algorithm 1 and the problem (3) satisfy the same KKT condition.

Therefore, Algorithm 1 converges to a local minimum of QPSM (2) since the problems (2) and (3) are equivalent.

Besides, the problem (6) has an unique solution under full column-rank matrix M due to the uniqueness of the SVD method. On the other hand, the experimental results in Section 5 represent that the proposed GPI method uniformly converges to the same objective value with a large amount of random initial guesses. Based on the unique solution of the problem (6) and the associated experimental results, it is rational to conjecture that the proposed GPI method converges to the global minimum of QPSM.

4 Two special cases of quadratic problem on the Stiefel manifold

4.1 Orthogonal least square regression

The orthogonal least square regression (OLSR) can be written as

$$\min_{W^{\text{T}}W = I_k, b} \|X^{\text{T}}W + \mathbf{1}b^{\text{T}} - Y\|_F^2, \tag{13}$$

where the data matrix $X \in \mathbb{R}^{m \times n}$ and the hypothesis matrix $Y \in \mathbb{R}^{n \times k}$ with $\mathbf{1} = (1, 1, \ldots, 1)^{\text{T}} \in \mathbb{R}^{n \times 1}$. Moreover, $W \in \mathbb{R}^{m \times k}$ is the regression matrix and $b \in \mathbb{R}^{k \times 1}$ is the bias vector. Obviously, b is free from any constraint. By virtue of the extreme value condition with regard to b, we can derive as

$$\frac{\partial \|X^{\text{T}}W + \mathbf{1}b^{\text{T}} - Y\|_F^2}{\partial b} = 0$$
$$\Rightarrow W^{\text{T}}X\mathbf{1} + b\mathbf{1}^{\text{T}}\mathbf{1} - Y^{\text{T}}\mathbf{1} = 0$$
$$\Rightarrow b = \frac{1}{n}(Y^{\text{T}}\mathbf{1} - W^{\text{T}}X\mathbf{1}).$$

By substituting the above result as $b = \frac{1}{n}(Y^{\text{T}}\mathbf{1} - W^{\text{T}}X\mathbf{1})$, Eq. (13) can be simplified to

$$\min_{W^{\text{T}}W = I_k} \|H(X^{\text{T}}W - Y)\|_F^2, \tag{14}$$

where $H = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\text{T}}$.

Accordingly, the problem (14) can be further reformulated into

$$\min_{W^{\text{T}}W = I_k} \text{Tr}(W^{\text{T}}AW - 2W^{\text{T}}B), \tag{15}$$

in which

$$\begin{cases} A = XHX^{\mathrm{T}}, \\ B = XHY. \end{cases}$$

Apparently, Eq. (15) is in the exact same form as QPSM in (2). Therefore, OLSR in (13) can be solved via Algorithm 1.

4.2 Unbalanced orthogonal procrustes problem

Definition 1. With $Q \in \mathbb{R}^{m \times k}$, $E \in \mathbb{R}^{n \times m}$ and $G \in \mathbb{R}^{n \times k}$, we name the optimization problem

$$\min_{Q^{\mathrm{T}}Q=I_k} \|EQ - G\|_F^2 \tag{16}$$

(1) balanced OPP if and only if $m = k$;
(2) unbalanced orthogonal procrustes problem (UOPP) if and only if $m > k$. Especially when Q serves as a column vector ($k = 1$), the problem (16) degenerates to

$$\min_{q^{\mathrm{T}}q=1} \|Eq - g\|_2^2, \tag{17}$$

which is known as the least square problem with a LSQE. In [17, 18], LSQE in (17) can be solved with the closed form solution.

4.2.1 *Balanced orthogonal procrustes problem revisited*

To solve the balanced OPP ($m = k$), we could expand Eq. (16) into

$$\min_{Q^{\mathrm{T}}Q=I_k} \|EQ - G\|_F^2 \Rightarrow \min_{Q^{\mathrm{T}}Q=I_k} \|E\|_F^2 + \|G\|_F^2 - 2\mathrm{Tr}(Q^{\mathrm{T}}E^{\mathrm{T}}G) \Rightarrow \max_{Q^{\mathrm{T}}Q=I_k} \mathrm{Tr}(Q^{\mathrm{T}}E^{\mathrm{T}}G),$$

which is same as the problem (6) with treating $E^{\mathrm{T}}G = M$.

Thus, the balanced OPP has the analytical solution of the closed form (7).

4.2.2 *Unbalanced orthogonal procrustes problem*

When $m > k$, UOPP (16) can be expanded into

$$\min_{Q^{\mathrm{T}}Q=I_k} \|EQ - G\|_F^2 \Rightarrow \min_{Q^{\mathrm{T}}Q=I_k} \mathrm{Tr}(Q^{\mathrm{T}}E^{\mathrm{T}}EQ - 2Q^{\mathrm{T}}E^{\mathrm{T}}G). \tag{18}$$

Denote $E^{\mathrm{T}}E = A$ and $E^{\mathrm{T}}G = B$, then Eq. (18) is in the exact same form as QPSM (2). Based on Algorithm 1, Algorithm 2 can be proposed to converge to a local minimum of UOPP monotonically due to the theoretical supports proved in Section 3.

Algorithm 2 GPI for solving UOPP in (16)

Input: The matrix $E \in \mathbb{R}^{n \times m}$ and the matrix $G \in \mathbb{R}^{n \times k}$ where $m > k$.
Initialize $Q \in \mathbb{R}^{m \times k}$ and γ such that $Q^{\mathrm{T}}Q = I_k$ and the matrix $\gamma I_m - E^{\mathrm{T}}E$ is positive definite, respectively.
While not converge **do**
(1) Update matrix $M \leftarrow 2(\gamma I_m - E^{\mathrm{T}}E)Q + 2E^{\mathrm{T}}G$.
(2) Calculate $U \in \mathbb{R}^{m \times k}$ and $V \in \mathbb{R}^{k \times k}$ via the compact SVD of M as $M = USV^{\mathrm{T}}$.
(3) Update $Q \leftarrow UV^{\mathrm{T}}$.
End while
Return Q.

Generally speaking, QPSM cannot be reformulated into UOPP while UOPP could always be rewritten into QPSM. Therefore, the GPI method is more general than other approaches, which can only cope with UOPP. Based on the experimental results involved in the next section, the proposed GPI method takes much less time to converge to the solution of UOPP.
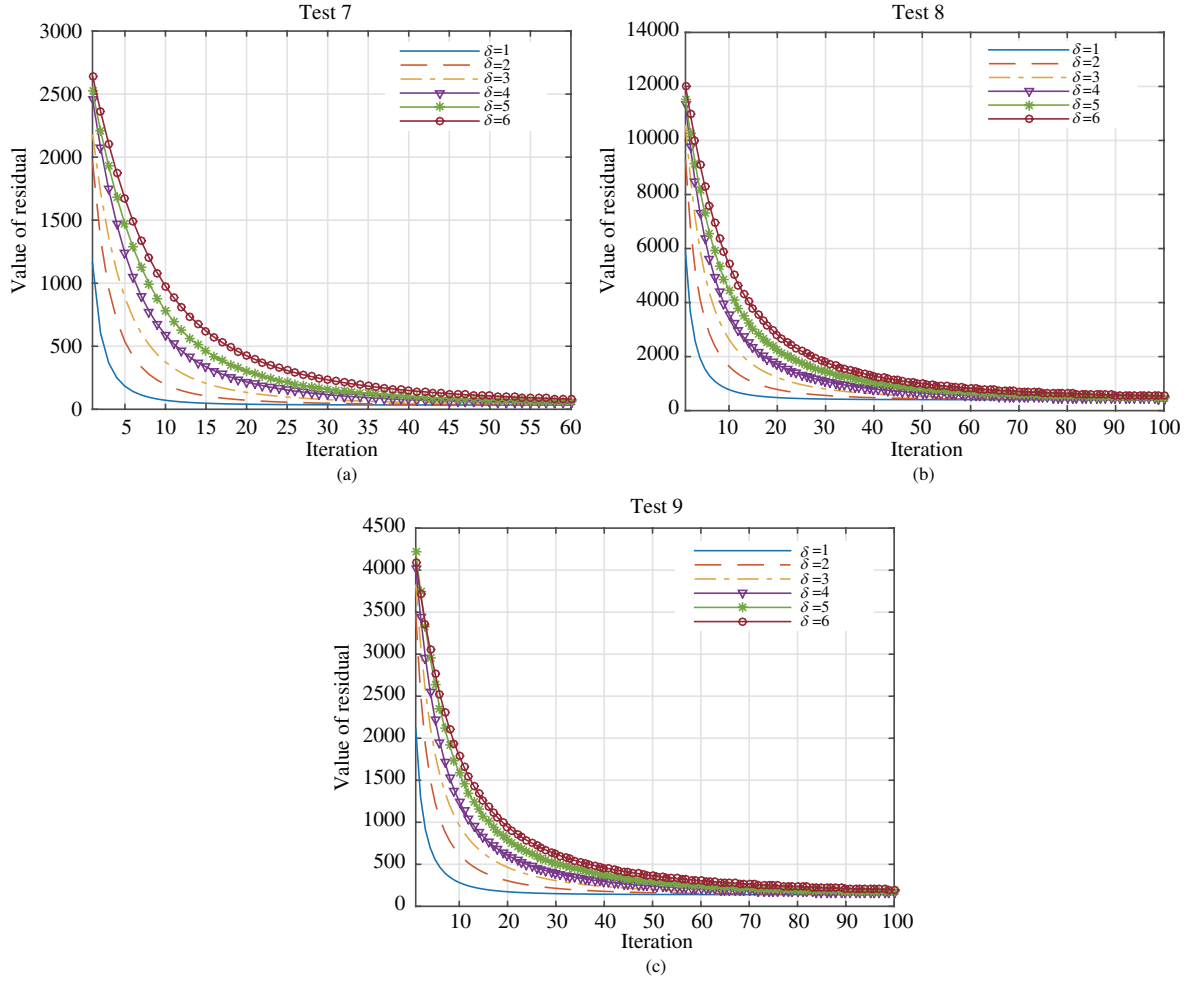
Figure 1   (Color online) Comparisons of 6 different values of δ are performed under the GPI method with 3 different data matrices. (a) (50, 100, 30); (b) (80, 170, 80); (c) (40, 120, 60).

5   Experimental results

In this section, we analyze and report the numerical results of the GPI method represented by both Algorithms 1 and 2. We randomly choose the test data matrix with normally distributed singular values.

Besides, the computer we use is MacBook Air, whose CPU is 1.4 GHz Intel Core i5, RAM is 4 GB 1600 MHz DDR3 and operating system is OS X Yosemite 10.10.5.

Case 1 (parameter dependence): Firstly, we try to investigate the GPI method in Algorithm 2 via varying the relaxation parameter γ. Suppose l_e is the largest eigenvalue of $E^{\mathrm{T}}E$, then we can let $\gamma = \delta l_e$ such that $\gamma I_m - E^{\mathrm{T}}E$ is a positive definite matrix, where δ is an arbitrary constant.

From Figure 1, we can further notice that although the convergence rate for Algorithm 2 is inversely proportional to the value of γ, the relaxation parameter γ does not affect the uniform convergence of the GPI method.

Case 2 (CPU time comparison for solving UOPP): Secondly, we further investigate the proposed GPI method in Algorithm 2 by comparing it with five existing approaches mentioned in Section 1 as EB [7], RSR [8], LSR [9], SP [10] and LR [12].

Based on solving LSQE problem, RSR [8] and LSR [9] respectively update the solution row by row and column by column iteratively. EB [7] utilizes the expanded balanced OPP as the objective function. SP [10] employs the projection method combined with correction techniques (PMCT) [11]. LR [12] solves UOPP by fixing different Lagrangian multipliers. The proposed GPI method includes two terms as $E^{\mathrm{T}}E$ outside the loop and $\tilde{A}W$ within the loop, whose orders of complexity are m^2n and m^2k, respectively.

Table 1 Orders of complexity for 6 algorithms[a]

	RSR [8]	LSR [9]	SP [10]
Order of the complexity	$O(mnk + m^3kt)$	$O(mnk + m^3kt)$	$O(mnk + (m^2n + m^3)t)$
	LR [12]	EB [7]	GPI (ours)
Order of the complexity	$O(m^2n + nk^2 + m^2kt)$	$O(m^3 + (m^2n + m^3)t)$	$O(m^2n + m^2kt)$

a) t stands for the iteration number and (n, m, k) stands for the dimension.

Table 2 Comparisons of CPU time (s) under the square matrix E for case 2[b]

$(n = m = 200)$	RSR [8]	LSR [9]	SP [10]	LR [12]	EB [7]	GPI (ours)
$k = 10$	64.940	23.426	2.386	0.541	0.337	**0.228**
$k = 15$	136.020	21.635	3.221	1.134	0.347	**0.226**
$k = 20$	229.851	20.560	5.054	1.806	0.445s	**0.273**
$(n = m = 1000)$	RSR [8]	LSR [9]	SP [10]	LR [12]	EB [7]	GPI (ours)
$k = 10$	–	842.849	132.232	3.869	11.440	**1.290**
$k = 15$	–	851.231	196.761	5.180	12.534	**1.434**
$k = 20$	–	860.746	260.132	7.700	12.625	**1.575**

b) Iteration stops when $\|EQ_{i-1} - G\|_F^2 - \|EQ_i - G\|_F^2 \leqslant \tau$ where $\tau = 10^{-3}$.

Table 3 Comparisons of CPU time (s) under the general dimension for case 2[b]

Dimension (n, m, k)	RSR [8]	LSR [9]	SP [10]	LR [12]	EB [7]	GPI (ours)
$(5000, 500, 15)$	713.156	528.034	450.028	20.709	16.554	**3.581**
$(10000, 1000, 30)$	–	–	–	56.772	191.970	**9.384**
$(3000, 3000, 90)$	–	–	–	186.125	395.401	**17.320**
$(30000, 1500, 30)$	–	–	–	306.132	1056.311	**19.440**
$(5000, 4000, 100)$	–	–	–	405.937	1187.512	**30.128**
$(100000, 3000, 50)$	–	–	–	–	–	**215.173**

Besides, these two terms have the highest orders of complexity for the proposed GPI method. Besides, the order of the complexity for each method is shown in Table 1.

The comparative results are based on fixing E as the square matrix at first hand (Table 2) and then extend E to a more general case (Table 3) afterwards. (Mark '–' in Tables 2 and 3 represents that it takes too much time to record in the tables.)

(1) From Figure 2, we notice that the existing methods as EB [7], RSR [8], LSR [9], SP [10] LR [12] and the proposed GPI method converge to the same objective value under the same input data. Besides, our proposed GPI method converges faster than other approaches during iteration.

(2) From Table 1, the proposed GPI method has the lowest order of complexity due to its succinct computational process to obtain the optimal solution. During the experiments, we observe that the iteration number t for the LR method is usually very large for the convergence. Thus, the time consumption for LR method is much larger than that for the proposed GPI method though orders of complexity for these two approaches seem close. Besides, the GPI method becomes more efficient when n (the number of data) is large.

(3) From Table 2, the proposed Algorithm 2 (GPI) serves as the most efficient method under the square matrix case.

(4) From Table 3, we can observe that LSR [9], SP [10] and RSR [8] are unable to compete with LR [12], EB [7] and GPI due to the complex updating procedures including the expanded OPP and solving LSQE. Especially when the dimension increases, the superiority of our proposed GPI method would be more obvious.

Case 3 (CPU time comparison for solving LSQE): Finally, the projection method combined with correction techniques (PMCT) [11] is compared to the GPI method in Algorithm 2 targeting at solving the least square regression with a LSQE in (17). Actually, solving LSQE (17) is no different from solving
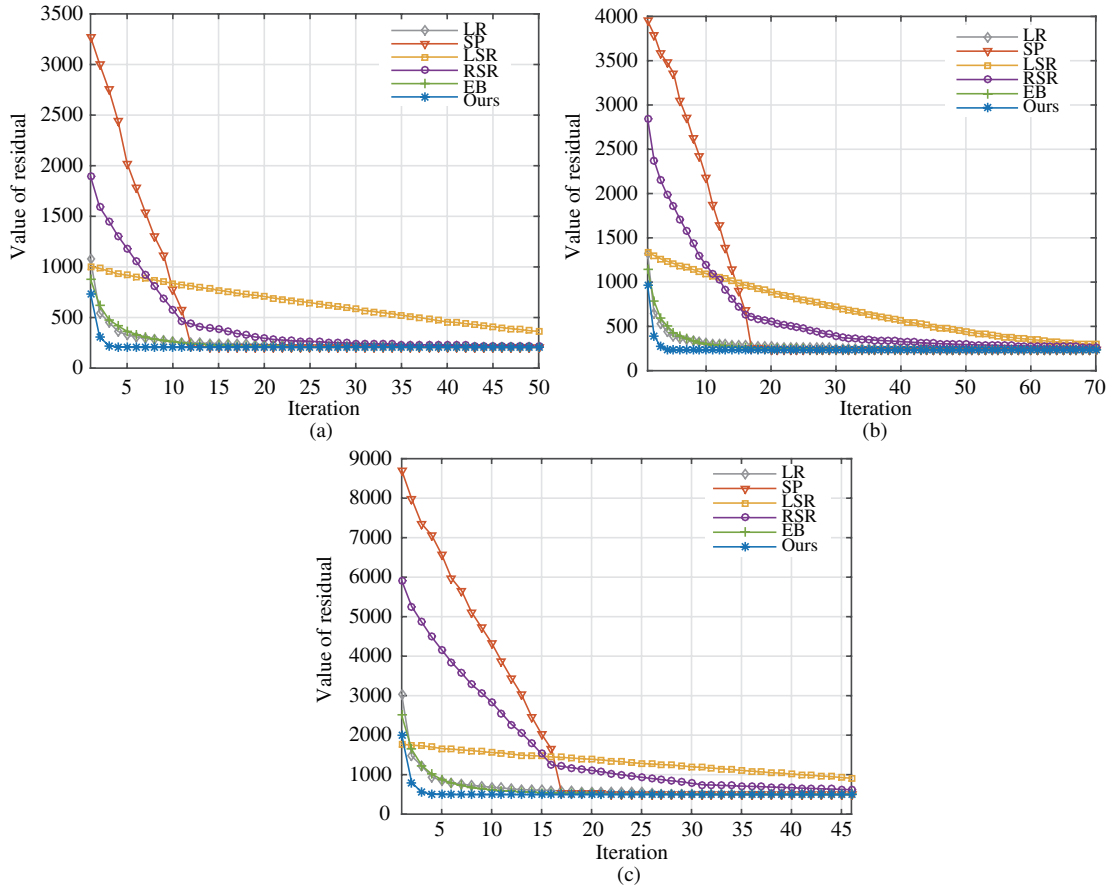
Figure 2 (Color online) Comparisons of the convergence rate are performed for 6 approaches including EB [7], RSR [8], LSR [9], SP [10] LR [12] and our GPI method under 3 different data matrices. (a) (100, 10, 100); (b) (100, 15, 100); (c) (200, 15, 200).
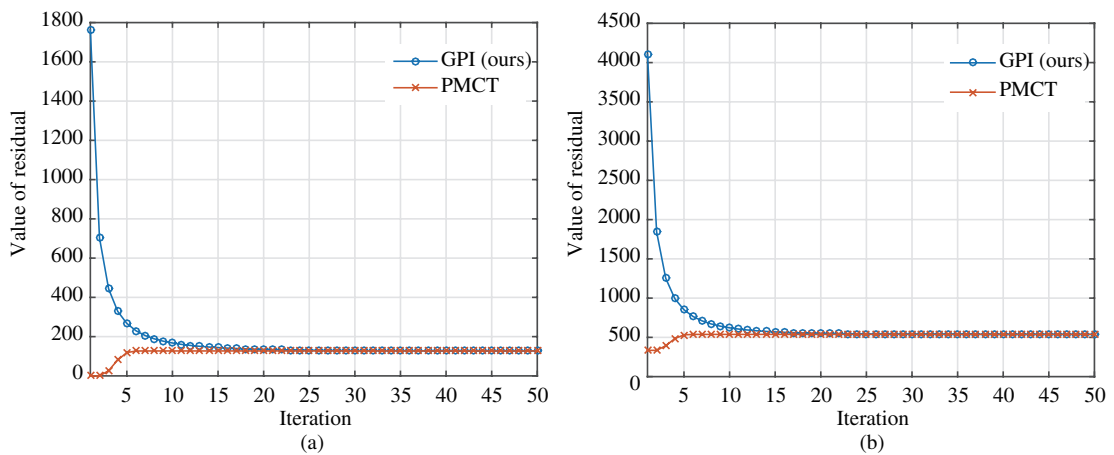

Figure 3 (Color online) Comparisons of PMCT [11] and GPI are performed over 2 different data matrices. (a) (900, 1000); (b) (2000, 1700).

UOPP (16) under $k = 1$.

(1) From Figure 3, we can notice that PMCT [11] and Algorithm 2 (GPI) converge to the same objective value though in terms of the different patterns.

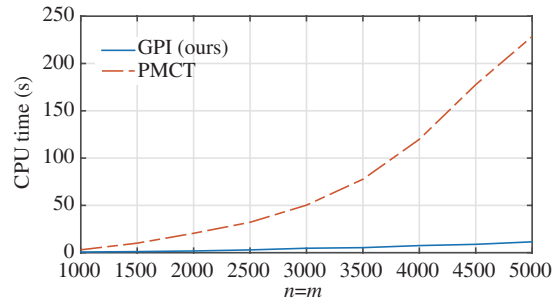(2) From Figure 4, Algorithm 2 (GPI) takes much less time for convergence than PMCT [11] does.

Figure 4   (Color online) CPU time comparison under case 3, dimension(n, m).

6   Concluding remarks

In this paper, we analyze the QPSM by deriving a novel GPI method. Based on the proposed GPI method, two special and significant cases of QPSM known as the orthogonal least square regression and the unbalanced orthogonal procrustes problem are under further investigation. With the theoretical supports, the GPI method decreases the objective value of the QPSM problem monotonically to a local minimum until convergence. Eventually, the effectiveness and the superiority of the proposed GPI method are verified empirically. In sum, the proposed GPI method not only takes less CPU time to converge to the optimal solution with a random initial guess but becomes much more efficient especially for the data matrix of large dimension as well.

Conflict of interest   The authors declare that they have no conflict of interest.

References

1   Green B. The orthogonal approximation of an oblique simple structure in factor analysis. Psychometrika, 1952, 17: 429–440

2   Hurley J, Cattell R. The procrustes program: producing direct rotation to test a hypothesized factor structure. Behav Sci, 1962, 6: 258–262

3   Golub G H, van Loan C F. Matrix Computations. Baltimore: The Johns Hopkins University Press, 1989

4   Thomas V. Algorithms for the weighted orthogonal procrustes problem and other least squares problems. Dissertation for Ph.D. Degree. Umeå: Umeå University, 2006

5   Souza P, Leite C, Borges H, et al. Online algorithm based on support vectors for orthogonal regression. Pattern Recogn Lett, 2013, 34: 1394–1404

6   Chu M, Trendafilov N. The orthogonally constrained regression revisted. J Comput Graph Stat, 2001, 10: 746–771

7   Green B, Goers J. A problem with congruence. In: Proceedings of the Annual Meeting of the Psychometric Society, Monterey, 1979

8   Park H. A parallel algorithm for the unbalanced orthogonal procrustes problem. Parall Comput, 1991, 17: 913–923

9   Bojanczyk A, Lutoborski A. The procrustes problem for orthogonal stiefel matrices. SIAM J Sci Comput, 1999, 21: 1291–1304

10   Zhang Z Y, Du K Q. Successive projection method for solving the unbalanced procrustes problem. Sci China Ser A-Math, 2006, 49: 971–986

11   Zhang Z Y, Huang Y S. A projection method for least square problems with quadratic equality constraint. SIAM J Matr Anal Appl, 2003, 25: 188–212

12   Xia Y, Han Y W. Partial lagrangian relaxation for the unbalanced orthogonal procrustes problem. Math Meth Oper Res, 2014, 79: 225–237

13   Journée M, Nesterov Y, Richtárik P, et al. Generalized power method for sparse principal component analysis. J Mach Learn Res, 2008, 11: 517–553

14   Fiori S. Formulation and integration of learning differential equations on the Stiefel manifold. IEEE Trans Neural Netw, 2005, 16: 1697–1701

15   Nie F P, Yuan J J, Huang H. Optimal mean robust principal component analysis. In: Proceedings of the 31st International Conference on Machine Learning, Beijing, 2014. 2755–2763

16   Kaneko T, Fiori S, Tanaka T. Empirical arithmetic averaging over the compact stiefel manifold. IEEE Trans Signal Process, 2013, 61: 883–894

17   Gander W, Golub G H, von Matt U. A constrained eigenvalue problem. In: Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms. Berlin: Springer, 1991. 677–686

18   Joachims T. Transductive learning via spectral graph partitioning. In: Proceedings of the 20th International Conference on Machine Learning, Washington, 2003. 51–60