

RESEARCH

Open Access



Strictly contractive Peaceman–Rachford splitting method to recover the corrupted low rank matrix

Zheng-Fen Jin¹ , Zhongping Wan^{2*} and Zhiyong Zhang³

*Correspondence:

mathwanzhp@whu.edu.cn

²School of Mathematics and Statistics, Wuhan University, Wuhan, P.R. China

Full list of author information is available at the end of the article

Abstract

The strictly contractive Peaceman–Rachford splitting method (SC-PRSM) attracts much attention on solving the separable convex programming. In this paper, the SC-PRSM is first applied to recover the corrupted low rank matrix, which extends the application of the SC-PRSM. At each iteration, we just solve two easy subproblems, where one subproblem has a closed solution and another needs to solve linear equations by the conjugate gradient method. Finally, numerical comparisons with the existing types of the alternating direction method of multipliers show that the SC-PRSM is efficient and competitive for recovering the low rank matrix problems.

MSC: 90C25; 90C06; 90C90

Keywords: Low rank matrix; Strictly contractive Peaceman–Rachford splitting method; Nuclear norm minimization

1 Introduction

Recovering a corrupted low rank matrix from a small amount of observations has attracted a lot of attention in many applications, such as online recommendation system and collaborative filtering [1], the Joster joke data [2], DNA data [3] and the famous Netflix problem [4]. This problem is the famous matrix completion (MC) problem, and its mathematical formula can be expressed as

$$\min_{X \in \mathbb{R}^{m \times n}} \|X\|_*, \quad \text{s.t. } X_{ij} = M_{ij}, \quad i, j \in \Omega, \quad (1)$$

where Ω is a given set of the index pairs (i, j) , $\|X\|_*$ is the nuclear norm, which is defined as the sum of its singular values. Assuming the matrix X has r positive singular values of $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$, we can get $\|X\|_* = \sum_{i=1}^r \sigma_i(X)$. It is well known that the nuclear norm is the best convex approximation of the rank function over the unit ball of matrices with norm less than one [5]. Furthermore, a general form of the MC problem is nuclear norm minimization with affine constraint, which can be depicted as

$$\min_{X \in \mathbb{R}^{m \times n}} \|X\|_*, \quad \text{s.t. } \mathcal{A}(X) = b, \quad (2)$$

where $\mathcal{A} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^p$ is a linear map and $b \in \mathbb{R}^p$ is a given measurement vector. The b is always contaminated by noise in practical applications, so the problem (2) can be relaxed to the following regularized nuclear norm minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} \|X\|_* + \frac{\gamma}{2} \|\mathcal{A}(X) - b\|_2^2, \tag{3}$$

where γ is the regularization parameter, which balances the two terms for obtaining the optimal solution.

Recently, many efficient algorithms could solve this problem (3), such as SDPT3 [6], singular value thresholding (SVT) [7], fixed point continuation with the approximate SVD (FPCA) method [5], a proximal point algorithm [8], an accelerated proximal gradient (APG) algorithm [9], the alternation direction method of multiplier (ADMM) type of algorithms [10–12], etc. But to the best of our knowledge, there are few studies on the application of SC-PRSM for recovering the corrupted low rank matrix problems. In this paper, we are going to further study the SC-PRSM for the problem (3).

The strictly contractive Peaceman–Rachford splitting method (SC-PRSM) is proposed in [13] by attaching an underdetermined relaxation factor α to the penalty parameter in the steps of Lagrange multiplier updating, which guarantees the convergence of PRSM proposed in [14] without some restrictive assumptions. It is worth mentioning that the difference between the PRSM and the ADMM is the addition of the intermediate update of the multipliers, which offers the same set of advantages for the two variable values. Recently, the SC-PRSM is also widely applied in solving many valuable problems, which can be referred to [15–18]. In this paper, we focus on applying the SC-PRSM to solve the problem (3) based on the proposed method of IADM-CG [12]. Moreover, we give some numerical comparisons to illustrate the advantage of the SC-PRSM for the problem (3).

The rest of this paper is organized as follows. In Sect. 2, we will give some preliminaries for this paper. In Sect. 3, we will give the construction of SC-PRSM for the problem (3). Some results of numerical experiments will be reported in Sect. 4. Finally, some conclusions are given.

2 Preliminaries

In this section, we mainly review some preliminaries on the ADMM and the SC-PRSM for further application of the SC-PRSM. Both of them are applied to solve the following convex minimization model with linear constraints and a separable objective function:

$$\min\{\theta_1(x) + \theta_2(y) \mid Ax + By = b, x \in \mathcal{X}, y \in \mathcal{Y}\}, \tag{4}$$

where $A \in \mathbb{R}^{m \times n_1}$, $B \in \mathbb{R}^{m \times n_2}$, $b \in \mathbb{R}^m$, $\mathcal{X} \subset \mathbb{R}^{n_1}$ and $\mathcal{Y} \subset \mathbb{R}^{n_2}$ are closed convex sets, and $\theta_1(x) : \mathcal{X} \rightarrow \mathbb{R}^m$ and $\theta_2(y) : \mathcal{Y} \rightarrow \mathbb{R}^m$ are convex functions.

The iterative scheme of ADMM [19, 20] for (4) reads

$$\begin{cases} x_{k+1} = \operatorname{argmin}\{\theta_1(x) - (\lambda_k)^T(Ax + By_k - b) + \frac{\beta}{2} \|Ax + By_k - b\|^2, x \in \mathcal{X}\}, \\ y_{k+1} = \operatorname{argmin}\{\theta_2(y) - (\lambda_k)^T(Ax_{k+1} + By - b) + \frac{\beta}{2} \|Ax_{k+1} + By - b\|^2, y \in \mathcal{Y}\}, \\ \lambda_{k+1} = \lambda_k - \beta[Ax_{k+1} + By_{k+1} - b], \end{cases} \tag{5}$$

where the λ is the Lagrangian multiplier associated with the linear constrains and the $\beta > 0$ is a penalty parameter. Moreover we note that the ADMM is viewed as an application of the Douglas–Rachford splitting method to the dual problem of (4) as analyzed in [21], and for its convergence performance one may be referred to [22, 23].

Applying the PRSM [24] to the dual problem of (4) [21], we can obtain the iterative schemes as follows:

$$\begin{cases} x_{k+1} = \operatorname{argmin}\{\theta_1(x) - (\lambda_k)^T(Ax + By_k - b) + \frac{\beta}{2}\|Ax + By_k - b\|^2, x \in \mathcal{X}\}, \\ \lambda_{k+\frac{1}{2}} = \lambda_k - \beta[Ax_{k+1} + By_k - b], \\ y_{k+1} = \operatorname{argmin}\{\theta_2(y) - (\lambda_{k+\frac{1}{2}})^T(Ax_{k+1} + By - b) + \frac{\beta}{2}\|Ax_{k+1} + By - b\|^2, y \in \mathcal{Y}\}, \\ \lambda_{k+1} = \lambda_{k+\frac{1}{2}} - \beta[Ax_{k+1} + By_{k+1} - b], \end{cases} \tag{6}$$

where updating the $\lambda_{k+\frac{1}{2}}$ is the only difference between the PRSM and the ADMM, which makes the two variable values to be treated fairly. But the PRSM needs more restrictive assumptions than ADMM to guarantee convergence [21]. The reader may refer to [24, 25] for more numerical verifications of the PRSM.

In order to overcome the weakness of a strict contraction of PRSM’s iterative sequence, He et al. [13] found that when an underdetermined relaxation factor $\alpha \in (0, 1)$ is attached to the penalty parameter β in the steps of Lagrange multipliers updating in (6), the resulting sequence becomes strictly contractive with respect to the solution set of (4). And they pointed out that the PRSM with an underdetermined relaxation factor can possibly establish a worst-case $\mathcal{O}(1/t)$ convergence rate in a nonergodic sence by the strict property. So they named this method SC-PRSM in [13] and gave its iterative scheme for (4) as follows:

$$\begin{cases} x_{k+1} = \operatorname{argmin}\{\theta_1(x) - (\lambda_k)^T(Ax + By_k - b) + \frac{\beta}{2}\|Ax + By_k - b\|^2, x \in \mathcal{X}\}, \\ \lambda_{k+\frac{1}{2}} = \lambda_k - \alpha\beta[Ax_{k+1} + By_k - b], \\ y_{k+1} = \operatorname{argmin}\{\theta_2(y) - (\lambda_{k+\frac{1}{2}})^T(Ax_{k+1} + By - b) + \frac{\beta}{2}\|Ax_{k+1} + By - b\|^2, y \in \mathcal{Y}\}, \\ \lambda_{k+1} = \lambda_{k+\frac{1}{2}} - \alpha\beta[Ax_{k+1} + By_{k+1} - b], \end{cases} \tag{7}$$

where $\alpha \in (0, 1)$. As showed in [13], the SC-PRSM ensures the sequence generated by (7) to be strictly contractive with respect to the solution set of (4). And without any further assumption on the model (4), they established worse-case $\mathcal{O}(1/t)$ convergence rates for (7). Moreover, the applications of SC-PRSM in machine learning and image processing illustrated that the SC-PRSM is numerically more efficient. Therefore, in this paper, we will further study the SC-PRSM for recovering a corrupted low rank matrix.

3 Algorithm

In this section, we will give the processing of extending the SC-PRSM to the problem of (3) based on the proposed the method of IADM-CG. Firstly, by introducing an auxiliary variable Y , the problem of (3) can be translated into the following form:

$$\min_{X,Y} \|X\|_* + \frac{\gamma}{2} \|\mathcal{A}(Y) - b\|_2^2, \quad \text{s.t. } X = Y. \tag{8}$$

Then its corresponding augmented Lagrangian function can be obtained as follows:

$$\mathcal{L}_{\mathcal{A}}(X, Y, Z) = \|X\|_* + \frac{\gamma}{2} \|\mathcal{A}(Y) - b\|_2^2 - \langle Z, X - Y \rangle + \frac{\mu}{2} \|X - Y\|_F^2, \tag{9}$$

where $Z \in \mathbb{R}^{m \times n}$ is the Lagrangian multiplier, $\mu > 0$ is the penalty parameter, $\langle \cdot \rangle$ represents the inner product of matrices or vectors.

Given $\{X_k, Y_k, Z_k\}$, applying the SC-PRSM for the problem of (8), we can obtain the following iteration equations:

$$X_{k+1} = \operatorname{argmin}_X \mathcal{L}_{\mathcal{A}}(X, Y_k, Z_k), \tag{10}$$

$$Z_{k+\frac{1}{2}} = Z_k - \alpha \mu (X_{k+1} - Y_k), \tag{11}$$

$$Y_{k+1} = \operatorname{argmin}_Y \mathcal{L}_{\mathcal{A}}(X_{k+1}, Y, Z_{k+\frac{1}{2}}), \tag{12}$$

$$Z_{k+1} = Z_{k+\frac{1}{2}} - \alpha \mu (X_{k+1} - Y_{k+1}), \tag{13}$$

where $\alpha \in (0, 1)$ is a relaxation factor. Observing the above iteration scheme, we can note that the two subproblem with respect to X and Y are mainly needed to solve. And the problem of minimizing X can be reformulated into

$$\begin{aligned} X_{k+1} &= \operatorname{argmin}_X \|X\|_* - \langle Z_k, X - Y_k \rangle + \frac{\mu}{2} \|X - Y_k\|_F^2 \\ &= \operatorname{argmin}_X \|X\|_* + \frac{\mu}{2} \left\| X - \left(Y_k + \frac{1}{\mu} Z_k \right) \right\|_F^2. \end{aligned} \tag{14}$$

Given the singular soft-thresholding operator proposed in [7], we can get

$$X_{k+1} = \mathcal{S}_{1/\mu} \left(Y_k + \frac{1}{\mu} Z_k \right). \tag{15}$$

Given the X_{k+1} , we can update the Lagrangian multiplier $Z_{k+\frac{1}{2}}$ by (11). Given $\{X_{k+1}, Z_{k+\frac{1}{2}}\}$, we will compute the iteration of Y_{k+1} .

Firstly, we note the augmented Lagrangian function with respect to Y as $Q(Y)$, which is expressed as

$$\begin{aligned} Q(Y) &= \mathcal{L}_{\mathcal{A}}(X_{k+1}, Y, Z_{k+\frac{1}{2}}) \\ &= \frac{\gamma}{2} \|\mathcal{A}(Y) - b\|_2^2 - \langle Z_{k+\frac{1}{2}}, X_{k+1} - Y \rangle + \frac{\mu}{2} \|X_{k+1} - Y\|_F^2. \end{aligned}$$

It is not hard to see that the $Q(Y)$ is a convex quadratic function, its gradient is

$$G(Y) = Z_{k+\frac{1}{2}} - \mu(X_{k+1} - Y) + \gamma \mathcal{A}^*(\mathcal{A}(Y) - b).$$

Let $G(Y) = 0$, then we obtain

$$(\mu I + \gamma(\mathcal{A}^* \mathcal{A}))Y = \mu X_{k+1} - Z_{k+\frac{1}{2}} + \gamma \mathcal{A}^* b, \tag{16}$$

so the Y_{k+1} can be expressed as

$$Y_{k+1} = (\mu I + \gamma \mathcal{A}^* \mathcal{A})^{-1} (\mu X_{k+1} - Z_{k+\frac{1}{2}} + \gamma \mathcal{A}^* b), \tag{17}$$

where I is an identity matrix, and \mathcal{A}^* is the adjoint of \mathcal{A} . It seems that the above linear system is easy to solve, but it may be expensive to solve directly in practice when the scale is large. Thus we consider applying the linear conjugate gradient method [12] for solving it iteratively. Now we let $C = \mu I + \gamma \mathcal{A}^* \mathcal{A}$, and $D_k = \mu X_{k+1} - Z_{k+\frac{1}{2}} + \gamma \mathcal{A}^* b$. Let $\widehat{Y}_0 = Y^k$, $\widehat{R}_0 = C\widehat{Y}_0 - D_k$ and $\widehat{P}_0 = -\widehat{R}_0$, and then the sequence $\{\widehat{Y}_i\}$ can be acquired iteratively as follows:

$$\begin{cases} \alpha_i = -\frac{\langle \widehat{R}_i, \widehat{P}_i \rangle}{\langle \widehat{P}_i, C\widehat{P}_i \rangle}, \\ \widehat{Y}_{i+1} = \widehat{Y}_i + \alpha_i \widehat{P}_i, \\ \widehat{R}_{i+1} = C\widehat{Y}_{i+1} - D_k, \\ \beta_{i+1} = \frac{\langle \widehat{R}_{i+1}, C\widehat{P}_i \rangle}{\langle \widehat{P}_i, C\widehat{P}_i \rangle}, \\ \widehat{P}_{i+1} = -\widehat{R}_{i+1} + \beta_{i+1} \widehat{P}_i, \end{cases} \tag{18}$$

and set $Y^{k+1} = \widehat{Y}_i$. It is worth being mentioned that the linear conjugate gradient method is very efficient to solve the linear system and has good convergence properties, for which one may refer to [26].

In the last step, we update the Lagrangian multiplier Z_{k+1} by (13).

To sum up, the expending of PRSM for problem (3) has the same form for solving the two subproblems of X and Y with IADM-CG. The one difference is the addition of the intermediate update of the multipliers $Z_{k+\frac{1}{2}}$. The other difference is adding a relaxation factor $\alpha \in (0, 1)$ to guarantee the convergence property of the SC-PRSM. Now we give the iteration scheme of SC-PRSM for solving Eq. (3) as follows:

Initialization: Given $Y_0, Z_0, \epsilon > 0, \gamma > 0, \mu > 0$ and i . Let $C = \mu I + \gamma(\mathcal{A}^* \mathcal{A}), k = 0$.

Step 1. Stop if some terminated condition is satisfied. Otherwise, continue.

Step 2. Compute X_{k+1} via (15) with the given Y_k and Z_k .

Step 3. Compute $Z_{k+\frac{1}{2}}$ via (11) with the given X_{k+1}, Y_k and Z_k .

Step 4. Given $\{X_{k+1}, Y_k, Z_{k+\frac{1}{2}}\}$, let $D = \mu X_{k+1} - Z_{k+\frac{1}{2}} + \gamma \mathcal{A}^* b$, update Y_{k+1} by (1)–(3).

(1) Let $\widehat{Y}_0 = Y_k, \widehat{R}_0 = C\widehat{Y}_0 - D_k$ and $\widehat{P}_0 = -\widehat{R}_0$. Let $i = 0$;

(2) if $\widehat{R}_i > \epsilon$ and $i < \bar{i}$ are satisfied;

compute \widehat{Y}_i by (18);

let $i = i + 1$;

(3) Let $Y_{k+1} = \widehat{Y}_i$.

Step 5. Given $X_{k+1}, Y_{k+1}, Z_{k+\frac{1}{2}}$, compute Z_{k+1} via (13).

Step 6. Let $k = k + 1$. Turn to **Step 1**.

Remark 1 In the above scheme, the constants of ϵ and \bar{i} are to guarantee the exactness of Y_{k+1} . Generally in numerical experiments, we set $\epsilon = 10^{-2}$ and $\bar{i} = 5$ to ensure the availability of the proposed algorithm.

At the end of this section, we state the convergence of SC-PRSM without proof. One should refer to [13] or [15] for more details.

Theorem 1 Let $\{(X^k, Y^k, Z^k)\}$ be a sequence generated by SC-PRSM. The sequence $\{(X^k, Y^k, Z^k)\}$ converges to some $\{(X^*, Y^*, Z^*)\}$.

4 Numerical experiments

In this section, we will report some numerical results on the SC-PRSM for matrix nuclear norm minimization problems, including the matrix complete problems and the regularized least square nuclear norm minimization in the cases of containing noise and noiseless. We firstly give the meanings of the different signal and the sets of parameters. The quantities m and n represent the dimension of the matrix, and r is the rank of the matrix. We denote by p the number of measurements. Given $r \leq \min(m, n)$, we can generate $M = M_L M_R^T$, where the matrices $M_L \in \mathbb{R}^{m \times r}$ and $M_R \in \mathbb{R}^{n \times r}$ are generated with independent identically distributed Gaussian entries [12]. The subset Ω of p elements is selected uniformly at random entries form $\{(i, j) : i = 1, \dots, m, j = 1, \dots, n\}$. And the partial discrete cosine transform (DCT) matrix is chosen as the linear map \mathcal{A} . Since the DCT matrix-vector multiplication is implemented implicitly by FFT, this makes us test the numerical experiments more efficiently [27]. We get the linear measurements $b = \mathcal{A}(M) + \omega$, where ω is the additive Gaussian noise of zero mean and standard deviation σ , which will be varied in different situations.

Now $sr = p/(mn)$ represents the sampling ratio, and $dr = r(m + n - r)$ is the number of degrees of freedom for a real-valued rank r matrix. As mentioned in [28, 29], the problem can be viewed as an easy problem when the ratio p/dr is greater than 3. On the contrary, it is regarded as a hard problem. Another ratio $FR = r(m + n - r)/p$ is also important for successfully recovering the matrix M . If $FR > 1$, it is impossible to recover the matrix because there is an infinite number of matrices X with rank r with the given entries [5]. Therefore the FR varies in $(0, 1)$ in this paper. In addition, we take $\mu = 2.5/\min(m, n)$, $\gamma = 2^4$ and $\alpha = 0.5$.

In all tests, the optimal solution produced by the proposed method is denoted X^* . The relative error is used to measure the quality of X^* to original M , i.e.

$$\text{RelErr} = \frac{\|X^* - M\|_F}{\|M\|_F}. \tag{19}$$

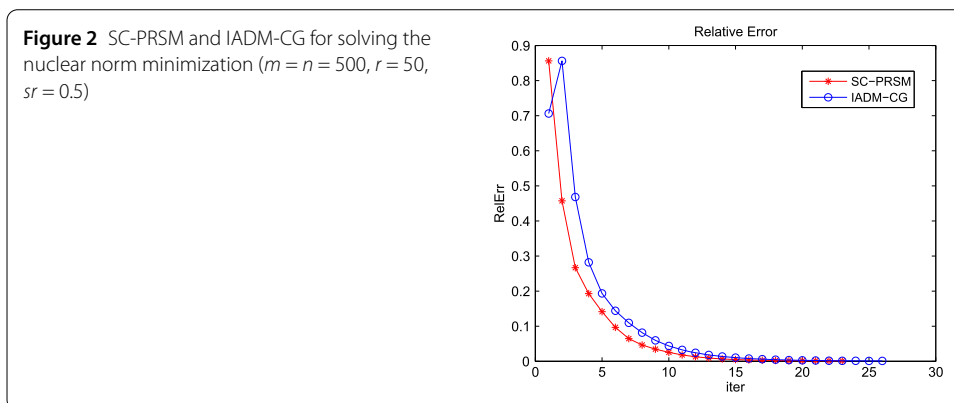
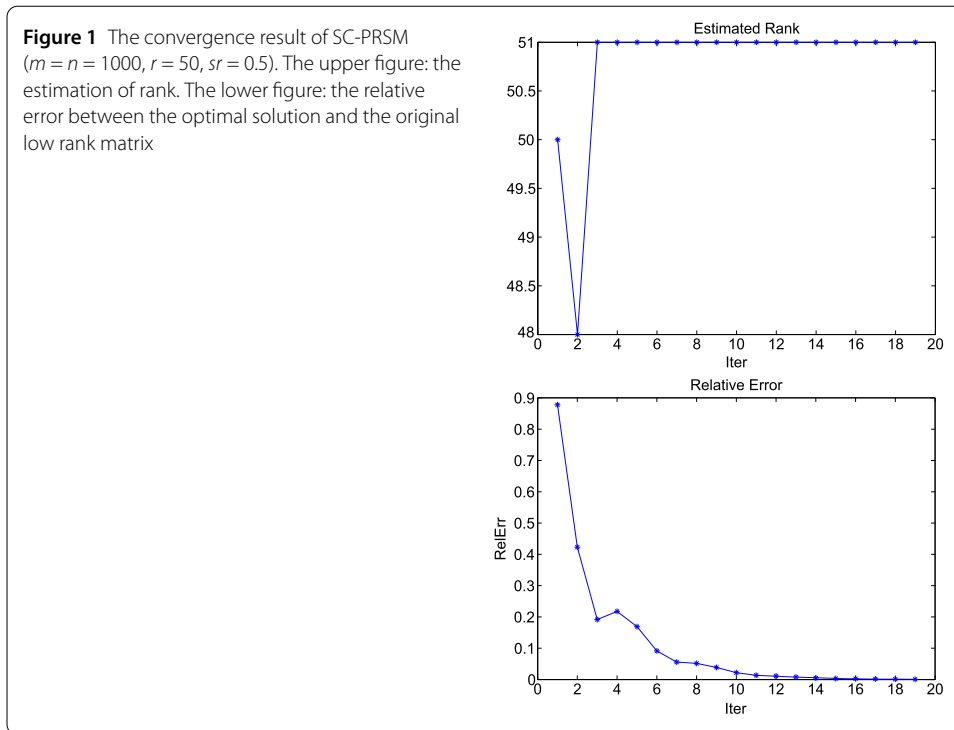
If the corresponding RelErr is less than 10^{-3} , it can be seen that M is recovered successfully by X^* , which has been used in [5, 7]. In all the tests, we take $\text{RelErr} = 10^{-4}$ as the terminal condition. In addition, we apply the same technique for solving the matrix singular value decomposition (SVD) as in [10, 12, 27, 30]

All numerical experiments were performed under Windows 7 premium and MATLAB v7.8(2009a) running on a Lenovo laptop with an Intel core CPU at 2.4 GHz and 2 GB memory.

4.1 Tests on the nuclear norm minimization problems

In this subsection, we mainly solve the problem (3) in different cases to illustrate the efficiency of the SC-PRSM.

In the first test, we mainly use the SC-PRSM for the problem (3) and report the numerical results of the relative error and the estimated rank, which can be seen in Fig. 1. Here we set $m = n = 1000$, $r = 50$, $sr = 0.5$. Observing Fig. 1, we can get close to the real rank just in the third iteration, and the relative error reduces to the given measure with less 20 iterations. So we can conclude that the SC-PRSM is efficient for the nuclear norm minimization.



In the second experiment, we compare the SC-PRSM with the IADM-CG [12] for solving the problem (3). The numerical results can be seen in Fig. 2. In this test, we apply the SC-PRSM and the IADM-CG [12] for solving the nuclear norm minimization problems in the cases of $m = n = 500, r = 50, sr = 0.5$. Then we compare the efficiency of the above two methods by displaying Fig. 2. From Fig. 2, we can see that the SC-PRSM needs less running time than the IADM-CG, and the SC-PRSM can attain a better accuracy for the solution. Thus we can see that the SC-PRSM needs to update the multiplier twice at each iteration, but it improves the accuracy of the solution of each iteration produced by the primal algorithm. Most of the running time is spent on solving the SVD decomposition of the matrix, so the SC-PRSM shows some improvement but not too dramatic. In the next test, we compare the SC-PRSM with the IADM-CG, IADM_BB [10] and IADM_NNLS [11] for solving the nuclear norm minimization with different settings. The numerical results is displayed in Table 1. Observing Table 1, we can see that the SC-PRSM and IADM-CG needs less

Table 1 SC-PRSM, IADM-CG, IADM_BB, IADM_NNLS for solving the easy nuclear norm minimization, $m = n$

(m, r)	sr	p/dr	SC-PRSM		IADM-CG		IADM_BB		IADM_NNLS	
			Time	RelErr	Time	RelErr	Time	RelErr	Time	RelErr
(128, 3)	0.4	8.64	1.19	8.87e-4	10.28	1.54e-3	12.27	3.08e-3	0.28	9.29e-4
	0.6	13.0	0.7	9.38e-4	0.76	9.45e-4	10.73	1.82e-3	0.28	7.73e-4
	0.8	17.3	0.5	7.30e-4	0.53	6.59e-4	0.96	9.26e-4	11.22	1.00e+0
(256, 5)	0.4	10.3	7.02	9.93e-4	5.08	8.08e-4	51.93	1.42e-3	0.60	6.57e-4
	0.6	15.5	3.05	7.66e-4	2.82	8.70e-4	5.67	8.39e-4	0.90	8.13e-4
	0.8	20.7	1.9	6.69e-4	2.10	4.37e-4	4.27	7.47e-4	30.6	9.99e-1
(512, 10)	0.4	10.3	24.2	9.34e-4	26.70	6.14e-4	64.50	8.08e-4	2.84	9.49e-4
	0.6	15.5	16.9	5.91e-4	17.93	8.60e-4	40.30	7.53e-4	3.68	6.79e-4
	0.8	20.7	10.6	6.05e-4	11.05	9.28e-4	22.79	9.18e-4	133.14	9.97e-1
(1024, 20)	0.4	10.3	132.8	9.51e-4	138.19	8.48e-4	301.47	9.57e-4	18.30	7.86e-4
	0.6	15.5	75.3	9.12e-4	79.50	8.66e-4	179.30	8.62e-4	17.01	8.48e-4
	0.8	20.7	54.9	5.72e-4	55.26	9.27e-4	119.30	9.20e-4	728.88	9.98e-1

Table 2 SC-PRSM, IADM-CG, IADM_BB, IADM_NNLS for solving the hard nuclear norm minimization, $m = n$

(m, r)	sr	p/dr	SC-PRSM		IADM-CG		IADM_BB		IADM_NNLS	
			Time	RelErr	Time	RelErr	Time	RelErr	Time	RelErr
(100, 10)	0.5	2.63	6.7	2.29e-3	7.75	2.29e-3	8.45	4.55e-3	14.52	1.10e-1
(200, 20)	0.5	2.63	4.8	9.70e-4	25.39	1.03e-3	30.27	2.05e-3	19.77	2.61e-2
(300, 30)	0.5	2.63	8.9	7.60e-4	8.99	9.28e-4	75.19	1.36e-3	47.95	1.62e-2
(400, 40)	0.5	2.63	14.6	8.38e-4	17.04	8.93e-4	64.25	9.99e-4	95.31	5.07e-3
(500, 50)	0.5	2.63	22.9	9.43e-4	30.00	8.11e-4	62.22	9.96e-4	58.31	9.96e-4

computing time than the IADM_BB for getting the same accuracy of the solution. Comparing SC-PRSM with IADM_NNLS, we can note that SC-PRSM needs more running time but it can solve the case of the $sr = 0.8$ successfully. However, the IADM_NNLS cannot attain a high accuracy of the solution within the given largest iteration steps. So in some sense, the SC-PRSM is more efficient.

In the fourth test, we compare SC-PRSM with IADM-CG, IADM_BB and IADM_NNLS for solving the hard problem of the nuclear norm minimization. From Table 2, it is easy to see that the SC-PRSM is more efficient than the IADM_NNLS and IADM_BB in the aspects of running time and accuracy of solution. And the SC-PRSM needs a little bit less running time than IADM-CG for getting a similar accuracy of solution. By the above limited numerical tests, it is illustrated that the SC-PRSM is promising and efficient for solving the hard problems of nuclear norm minimization.

In the last numerical test of this subsection, we apply the SC-PRSM for solving the nuclear norm minimization with different level noise. We set $m = n = 200$ and $\sigma = 10^{-1}, 10^{-2}, 10^{-4}$, respectively. It can be observed from Fig. 3 that the SC-PRSM can deal with the nuclear norm minimization problem with $\sigma = 0.1$ successfully. As is well known, as the level of noise is lower, the problem can be solved more easily. When the σ is 10^{-4} , the accuracy of solution can attain 10^{-4} . Therefore, this numerical test illustrated that the SC-PRSM is efficient for solving the low rank minimization problem with Gaussian noise.

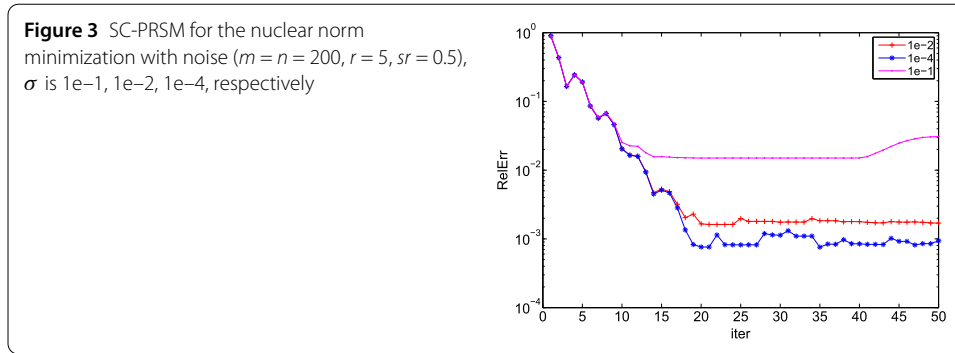


Table 3 SC-PRSM, IADM-CG, IADM_BB and IADM_NNLS for solving the matrix completion problems, $m = n, sr = 0.5$

(m, r)	p/dr	SC-PRSM		IADM-CG		IADM_BB		IADM_NNLS	
		Time	RelErr	Time	RelErr	Time	RelErr	Time	RelErr
(500, 5)	25.13	4.40	8.83e-4	4.82	3.92e-4	9.15	9.13e-4	0.38	6.06e-4
(500, 10)	12.63	5.00	8.72e-4	5.15	9.95e-4	7.51	9.12e-4	0.52	5.71e-4
(500, 15)	8.46	4.80	9.28e-4	5.12	9.93e-4	7.74	9.08e-4	0.70	5.11e-4
(500, 20)	6.38	5.19	7.12e-4	5.60	9.12e-4	8.25	9.29e-4	0.82	7.24e-4
(500, 25)	5.13	5.12	7.70e-4	5.89	1.00e-3	9.40	9.01e-4	1.37	6.45e-4
(500, 30)	4.30	6.75	8.46e-4	8.72	8.63e-4	12.24	8.67e-4	1.66	9.76e-4
(500, 35)	3.70	7.40	9.06e-4	8.56	8.16e-4	11.83	9.53e-4	2.66	8.95e-4
(500, 40)	3.26	8.10	9.89e-4	11.65	8.00e-4	15.04	9.72e-4	3.74	9.12e-4
(500, 45)	2.91	8.60	8.42e-4	11.59	9.69e-4	15.78	9.47e-4	7.24	9.66e-4
(500, 50)	2.632	11.16	9.52e-4	13.35	9.75e-4	18.17	9.89e-4	17.21	9.91e-4

4.2 Tests on the matrix completion problems

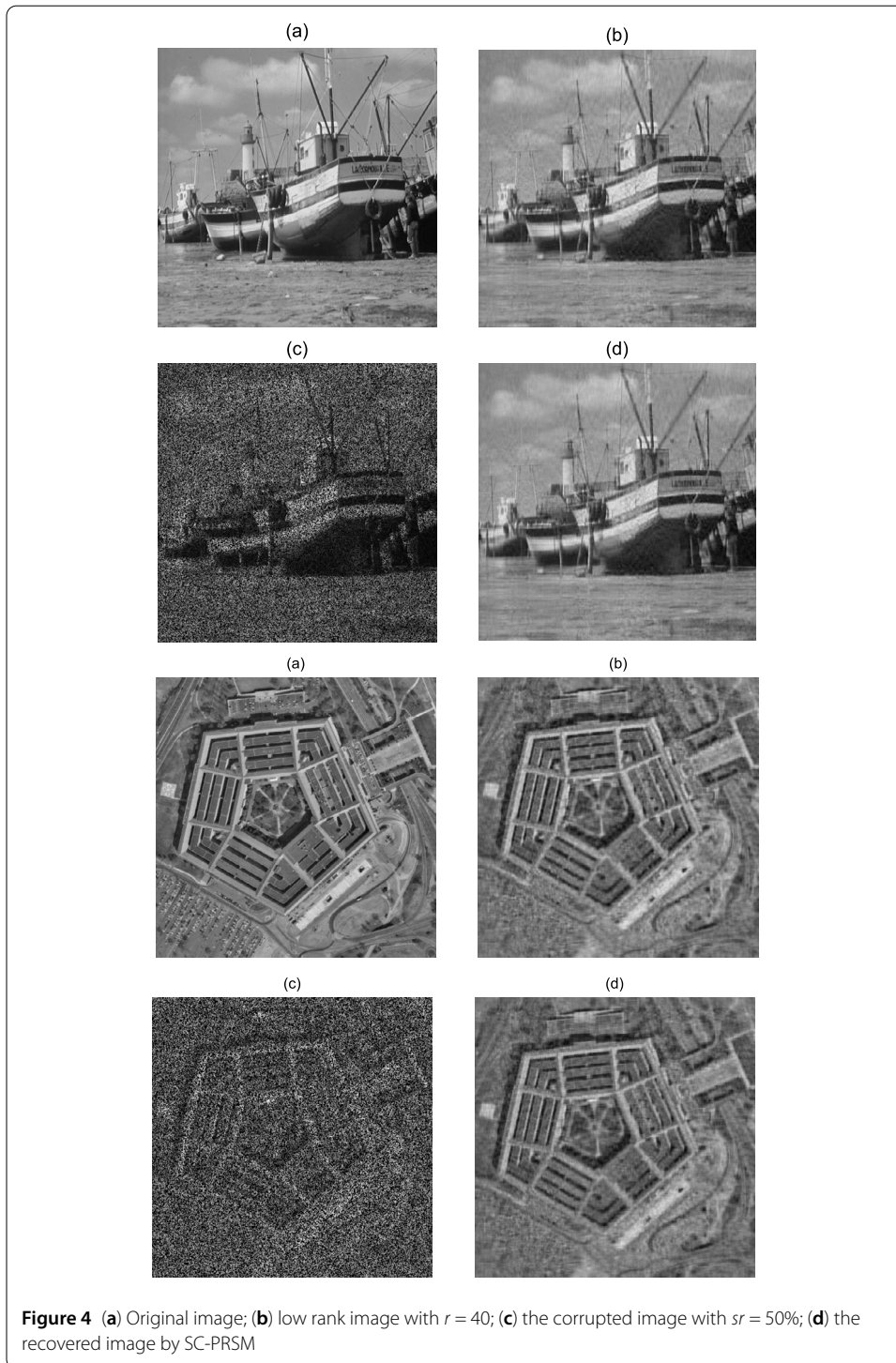
In this subsection, we will apply the SC-PRSM for solving the matrix completion problems and further verify the proposed method. The matrix completion problem can be reformulated as follows:

$$\min_{X \in \mathbb{R}^{m \times n}} \|X\|_* + \frac{\gamma}{2} \sum_{(i,j) \in \Omega} |X_{ij} - M_{ij}|^2, \quad \forall (i,j) \in \Omega. \tag{20}$$

Firstly, we use the SC-PRSM, IADM_NNLS, IADM_BB and IADM-CG to solve the low rank matrix completion problems in the noiseless case. In this test, when tol reaches $1e-3$, the SC-PRSM, IADM-CG and IADM_BB are terminated. For IADM_NNLS, all parameters are by default expected to have $\text{opts.tol_relchg} = 1e-3$.

Observing Table 3, the SC-PRSM is better than the IADM-CG and IADM_BB on the aspects of running time and accuracy of solution, but it needs more running time than IADM_NNLS. From the end line of Table 3, we can see that the SC-PRSM is more efficient than others for solving the hard problems. Thus we can conclude that the SC-PRSM is more efficient than IADM-CG and IADM_BB and comparable with IADM_NNLS by the limited tests.

In the last test, the numerical results are shown in Fig. 4. We apply the SC-PRSM for recovering the corrupted images, where the dimension of “boat” and “pentagon” is 512×512 and 450×450 , respectively. Firstly, we deal with the original image (a) by singular value decomposition to get a low rank image (b), where the rank is 40. And then we choose 50% elements of the known observations to obtain the corrupted image (c). Finally, we use the SC-PRSM to recover (c) and get a completed low rank image (d). Comparing (b) with (d),



we can find that the SC-PRSM can recover the corrupted low rank image successfully, which further illustrates its practicability.

5 Conclusions

In this paper, we mainly devoted our attention to proposing a SC-PRSM method based on the IADM-CG for solving the nuclear norm minimization problem, which expanded

the SC-PRSM to recover the corrupted low rank matrix. The classical ADMM is to solve the X -subproblem and Y -subproblem orderly and then to update the Lagrange multiplier. However, the SC-PRSM updates once the Lagrange multiplier after computing the X -subproblem and Y -subproblem, respectively. We introduce the relax factor α into the computing of the Lagrange multiplier for improving the convergence efficiency of the proposed method. Numerical results illustrated that the SC-PRSM can solve the nuclear norm minimization problems successfully. The last real test on recovering the corrupted low rank images further demonstrated the efficiency and practicability of SC-PRSM.

Acknowledgements

Not applicable.

Funding

The work of Z. Wan is supported by the National Natural Science Foundation of China Grant No. 11871383. The work of Z. Jin is supported by the Plan for Scientific Innovation Talent of Henan Province Grant No. 174200510011. The work of Z. Zhang is supported by the Program for Innovative Research Team (in Science and Technology) in University of Henan Province Grant No. 15IRTSTHN010.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

All authors contributed equally to this work. All authors read and approved the final manuscript.

Author details

¹School of Mathematics and Statistics, Henan University of Science and Technology, Luoyang, China. ²School of Mathematics and Statistics, Wuhan University, Wuhan, P.R. China. ³Information Engineering College, Henan University of Science and Technology, Luoyang, P.R. China.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 25 January 2019 Accepted: 6 May 2019 Published online: 22 May 2019

References

1. Srebro, N.: Learning with matrix factorizations. PhD thesis, Citeseer (2004)
2. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. *Inf. Retr.* **4**(2), 133–151 (2001)
3. Spellman, P.T., Sherlock, G., Zhang, M.Q., Iyer, V.R., Anders, K., Eisen, M.B., Brown, P.O., Botstein, D., Futcher, B.: Comprehensive identification of cell cycle-regulated genes of the yeast *saccharomyces cerevisiae* by microarray hybridization. *Mol. Biol. Cell* **9**(12), 3273–3297 (1998)
4. Netflix prize website. <http://www.netflixprize.com>
5. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. *Math. Program.* **128**(1–2), 321–353 (2011)
6. Tütüncü, R.H., Toh, K.-C., Todd, M.J.: Solving semidefinite-quadratic-linear programs using sdpt3. *Math. Program.* **95**(2), 189–217 (2003)
7. Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* **20**(4), 1956–1982 (2010)
8. Liu, Y.-J., Sun, D., Toh, K.-C.: An implementable proximal point algorithmic framework for nuclear norm minimization. *Math. Program.* **133**(1–2), 399–436 (2012)
9. Toh, K.-C., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pac. J. Optim.* **6**(615–640), 15 (2010)
10. Xiao, Y.-H., Jin, Z.-F.: An alternating direction method for linear-constrained matrix nuclear norm minimization. *Linear Algebra Appl.* **19**(3), 541–554 (2012)
11. Yang, J., Yuan, X.: Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Math. Comput.* **82**(281), 301–329 (2013)
12. Jin, Z.-F., Wang, Q., Wan, Z.: Recovering low-rank matrices from corrupted observations via the linear conjugate gradient algorithm. *J. Comput. Appl. Math.* **256**, 114–120 (2014)
13. He, B., Liu, H., Wang, Z., Yuan, X.: A strictly contractive Peaceman–Rachford splitting method for convex programming. *SIAM J. Optim.* **24**, 1011–1040 (2014)
14. Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, San Diego (1982)
15. Li, X., Yuan, X.: A proximal strictly contractive Peaceman–Rachford splitting method for convex programming with applications to imaging. *SIAM J. Imaging Sci.* **8**, 1332–1365 (2015)
16. Gu, Y., Jiang, B., Han, D.: A semi-proximal-based strictly contractive Peaceman–Rachford splitting method (2015). arXiv preprint. [arXiv:1506.02221](https://arxiv.org/abs/1506.02221)

17. Li, M., Yuan, X.: A strictly contractive Peaceman–Rachford splitting method with logarithmic-quadratic proximal regularization for convex programming. *Math. Oper. Res.* **40**, 842–858 (2015)
18. Sun, M., Liu, J.: A proximal Peaceman–Rachford splitting method for compressive sensing. *J. Appl. Math. Comput.* **50**, 349–363 (2016)
19. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Comput. Math. Appl.* **2**, 17–40 (1976)
20. Glowinski, R., Marocco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Rev. Fr. Autom. Inform. Rech. Opér.* **9**, 41–76 (1975)
21. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Fortin, M., Glowinski, R. (eds.) *Augmented Lagrange Method: Applications to the Solution of Boundary-Valued Problems*, pp. 299–331. North Holland, Amsterdam (1983)
22. He, B., Yuan, X.: On the $o(1/n)$ convergence rate of the Douglas–Rachford alternating direction method. *SIAM J. Imaging Sci.* **50**, 700–709 (2012)
23. Boyd, S.P., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**, 1–122 (2011)
24. Bertsekas, D.P.: *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, New York (1982)
25. Glowinski, R., Le Tallec, P.: *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*. SIAM, Philadelphia (1989)
26. Kelley, C.T.: *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, Philadelphia (1995)
27. Jin, Z.-F., Wan, Z., Jiao, Y., Lu, X.: An alternating direction method with continuation for nonconvex low rank minimization. *J. Sci. Comput.* **66**(2), 849–869 (2016)
28. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. *Found. Comput. Math.* **9**(6), 717–772 (2009)
29. Malek-Mohammadi, M., Babaie-Zadeh, M., Amini, A., Jutten, C.: Recovery of low-rank matrices under affine constraints via a smoothed rank function. *IEEE Trans. Signal Process.* **62**(4), 981–992 (2014)
30. Jin, Z.-F., Wan, Z., Zhao, X., Xiao, Y.: A penalty decomposition method for rank minimization problem with affine constraints. *Appl. Math. Model.* **39**, 4859–4870 (2015)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
