An Implementable Proximal Point Algorithmic Framework for Nuclear Norm Minimization

Yong-Jin Liu*, Defeng Sun†, and Kim-Chuan Toh‡

July 13, 2009; First revision: March 16, 2010;
Second revision: October 02, 2010

Abstract

The nuclear norm minimization problem is to find a matrix with the minimum nuclear norm subject to linear and second order cone constraints. Such a problem often arises from the convex relaxation of a rank minimization problem with noisy data, and arises in many fields of engineering and science. In this paper, we study inexact proximal point algorithms in the primal, dual and primal-dual forms for solving the nuclear norm minimization with linear equality and second order cone constraints. We design efficient implementations of these algorithms and present comprehensive convergence results. In particular, we investigate the performance of our proposed algorithms in which the inner sub-problems are approximately solved by the gradient projection method or the accelerated proximal gradient method. Our numerical results for solving randomly generated matrix completion problems and real matrix completion problems show that our algorithms perform favorably in comparison to several recently proposed state-of-the-art algorithms. Interestingly, our proposed algorithms are connected with other algorithms that have been studied in the literature.

Key words. Nuclear norm minimization, proximal point method, rank minimization, gradient projection method, accelerated proximal gradient method.

AMS subject classification. 46N10, 65K05, 90C22, 90C25.

*Faculty of Science, Shenyang Aerospace University, Shenyang, 110136, P.R. China (yongjin.liu.77@gmail.com). Part of this work was done while the author was with the Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576. The author's research is supported in part by the National Young Natural Science Foundation of China under project grant No. 11001180.

†Department of Mathematics and Risk Management Institute, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (matsundf@nus.edu.sg). This author's research is supported in part by Academic Research Fund under grant R-146-000-104-112.

‡Department of Mathematics, National University of Singapore, 10 Lower Kent Ridge Road, Singapore 119076 (mattohkc@nus.edu.sg); and Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576.

1 Introduction

Let $\Re^{n_1\times n_2}$ be the linear space of all $n_1\times n_2$ real matrices equipped with the inner product $\langle X, Y\rangle = \mathrm{Tr}(X^TY)$ and its induced norm $\|\cdot\|$, i.e., the Frobenius norm. Let $\mathcal{S}^n \subset \Re^{n\times n}$ be the space of $n\times n$ symmetric matrices. For any $X \in \Re^{n_1\times n_2}$, the nuclear norm $\|X\|_*$ of X is defined as the sum of its singular values and the operator norm $\|X\|_2$ of X is defined as the largest singular value. Let $\mathcal{Q} := \{0\}^{m_1} \times \mathcal{K}^{m_2}$, where the notation $\mathcal{K}^{m_2}$ stands for the second order cone (or ice-cream cone, or Lorentz cone) of dimension m_2, defined by

$$\mathcal{K}^{m_2} := \{x = (x_0; \bar{x}) \in \Re \times \Re^{m_2-1} : \|\bar{x}\| \le x_0\}. \tag{1}$$

In particular, $\mathcal{K}^1$ is the set of nonnegative reals $\Re_+$.

In this paper, we are interested in the following nuclear norm minimization (NNM) problem with linear equality and second order cone constraints:

$$\begin{array}{ll} \min & f_0(X) := \|X\|_* \\ \text{s.t.} & X \in \mathcal{F}_P := \{X \in \Re^{n_1\times n_2} \ : \ \mathcal{A}(X) \in b + \mathcal{Q}\}, \end{array} \tag{2}$$

where the linear transformation $\mathcal{A} : \Re^{n_1\times n_2} \to \Re^m$ and the vector $b \in \Re^m$ are given. Here, $m = m_1 + m_2$. We should emphasize that for the ease of presentation, we have considered the cone $\mathcal{Q} = \{0\}^{m_1} \times \mathcal{K}^{m_2}$. But the theory and algorithms developed in this paper can easily be extended to the more general cone which has the form: $\mathcal{Q} = \{0\}^{m_1} \times \mathcal{K}^{p_1} \times \cdots \times \mathcal{K}^{p_t}$, where for each $1 \le j \le t$, $\mathcal{K}^{p_j}$ is a second order cone. In particular, since $\mathcal{K}^1 = \Re_+$, the case of linear inequality constraints of the form $\mathcal{A}(X) \ge b$ also fits the analysis in our framework by considering $\mathcal{Q} = \Re_+ \times \cdots \times \Re_+$.

The NNM problem (2) often arises from the convex relaxation of a rank minimization problem with noisy data, and arises in many fields of engineering and science, see, e.g., [1, 3, 17, 18, 21, 27]. The rank minimization problem refers to finding a matrix $X \in \Re^{n_1\times n_2}$ to minimize $\mathrm{rank}(X)$ subject to linear constraints, i.e.,

$$\min\left\{\mathrm{rank}(X) \ : \ \mathcal{A}(X) = b, \ X \in \Re^{n_1\times n_2}\right\}. \tag{3}$$

Problem (3) is NP-hard in general and it is computationally hard to directly solve it in practice. Recent theoretical results (see, e.g., [3, 14, 40]), which were built upon recent breakthroughs in the emerging field of compressed sensing or compressive sampling pioneered by Candès and Tao [11] and Donoho [15], showed that under certain conditions, the rank minimization problem (3) may be solved via its tightest convex approximation:

$$\min\left\{\|X\|_* \ : \ \|b - \mathcal{A}(X)\| \le \delta, \ X \in \Re^{n_1\times n_2}\right\}, \tag{4}$$

where $\delta > 0$ estimates the uncertainty about the observation b if it is contaminated with noise. It can be readily seen that problem (4) is a special application of problem (2), see,

e.g., [2]. A frequent alternative to (4) is to consider solving the following nuclear norm regularized linear least squares problem (see, e.g., [31, 45]):

$$\min\Big\{\frac{1}{2}\|\mathcal{A}(X)-b\|^2+\mu\|X\|_* \ :\ X\in\Re^{n_1\times n_2}\Big\}, \tag{5}$$

where $\mu>0$ is a given parameter. For problems where a reasonable estimation of δ is possible, problem (4) is often preferred over problem (5). By checking the optimality conditions for problems (4) and (5), we can easily see that these two problems are equivalent to each other if δ and μ are chosen suitably. But, in in general it is difficult to determine δ a priori given μ or vice versa without knowing the solutions to problems (4) and (5). Therefore, it is more natural to consider solving (4) directly if δ is known, rather than solving (5). To the best of our knowledge, however, there has been no work developing algorithms for directly solving (4) when δ may be known. This is the main motivation of the paper to present the results concerning the proximal point algorithms for solving problem (2), which includes problem (4) as a special case.

The NNM problem (2) can equivalently be reformulated as the following semidefinite programming (SDP) problem (see, e.g., [28, 40]):

$$\min\Big\{(\mathrm{Tr}(W_1)+\mathrm{Tr}(W_2))/2 \ :\ \mathcal{A}(X)\in b+\mathcal{Q},\ \ \begin{bmatrix}W_1, X;\\ X^T, W_2\end{bmatrix}\succeq 0\Big\}, \tag{6}$$

whose dual is:

$$\max\Big\{b^Ty \ :\ y\in\mathcal{Q}^*,\ \ \begin{bmatrix}I_{n_1}, \mathcal{A}^*(y);\\ \mathcal{A}^*(y)^T, I_{n_2}\end{bmatrix}\succeq 0\Big\}, \tag{7}$$

where $X\in\Re^{n_1\times n_2}, W_1\in\mathcal{S}^{n_1}, W_2\in\mathcal{S}^{n_2}$, $\mathcal{Q}^*(:=\Re^{m_1}\times\mathcal{K}^{m_2})$ is the dual cone of $\mathcal{Q}$, and $\mathcal{A}^*$ denotes the adjoint of $\mathcal{A}$. Here, the notation "$\succeq 0$" means positive semidefiniteness. This suggests that one can use well developed SDP solvers based on interior point methods, such as SeDuMi [44] and SDPT3 [47], to solve (6) or (7) and therefore solve (2), see, e.g., [14, 40] for this approach in solving (2) with only linear equality constraints. However, these SDP solvers usually cannot solve (6) or (7) when both n_1 and n_2 are much larger than 100 or m is larger than $5,000$ since they need to solve large systems of linear equations to compute Newton directions.

Due to the difficulties in solving the SDP reformulation (6) or (7), several methods have been proposed to solve (2) directly with only linear equality constraints. In [40], Recht, Fazel and Parrilo considered the projected subgradient method. However, the convergence of the projected subgradient method in [40] is not known since problem (2) is a nonsmooth problem. Recht, Fazel and Parrilo [40] also made use of the low rank factorization technique introduced by Burer and Monteiro [8, 9] to solve (2) with only linear equality constraints. The potential difficulty of this method is that the low rank factorization formulation is no longer convex and the rank of the optimal matrix is generally unknown a priori. Recently, Cai, Candès and Shen [10] introduced the singular value thresholding (SVT) algorithm to solve a regularized version of (2), i.e.,

$$\min\Big\{\lambda\|X\|_*+\frac{1}{2}\|X\|^2 \ :\ \mathcal{A}(X)=b,\ X\in\Re^{n_1\times n_2}\Big\}, \tag{8}$$

where $\lambda > 0$ is a given parameter[1]. The SVT algorithm is actually a gradient method applied to the dual problem of (8).

In this paper, we develop three proximal point algorithms for solving (2) in the primal, dual and primal-dual forms, all of which are based on the classic ideas of the general proximal point method studied in [32, 42]. In addition, we show that some of the recently proposed fast methods for solving (2) are actually either truncated or special cases of these three algorithms.

The first algorithm for solving (2), namely, the primal proximal point algorithm (PPA), is the application of the general proximal point method to the primal problem (2). Given a sequence of positive parameters λ_k such that

$$0 < \lambda_k \uparrow \lambda_\infty \leq +\infty \tag{9}$$

and an initial point $X^0 \in \Re^{n_1 \times n_2}$, the primal PPA for solving (2) generates a sequence $\{X^k\}$ by the following scheme:

$$X^{k+1} \approx \arg\min_{X \in \mathcal{F}_P} \Big\{ f_0(X) + \frac{1}{2\lambda_k} \|X - X^k\|^2 \Big\}. \tag{10}$$

Here, (9) means that $\{\lambda_k\}$ is a nondecreasing sequence of positive parameters that converges to λ_∞, which is allowed to take the $+\infty$ value.

The second algorithm, namely, the dual PPA, is the application of the general proximal point method to the dual problem of (2), which, as a by-product, yields an optimal solution to problem (2). The dual problem associated with (2) is as follows:

$$\max_{y \in \mathcal{Q}^*} g_0(y), \tag{11}$$

where g_0 is the concave function defined by

$$g_0(y) = \inf \Big\{ f_0(X) + \langle y, b - \mathcal{A}(X) \rangle \ : \ X \in \Re^{n_1 \times n_2} \Big\}.$$

Given a sequence $\{\lambda_k\}$ satisfying (9) and an initial point $y^0 \in \mathcal{Q}^*$, the sequence $\{y^k\} \subset \mathcal{Q}^*$ generated by the dual PPA is as follows:

$$y^{k+1} \approx \operatorname*{argmax}_{y \in \mathcal{Q}^*} \Big\{ g_0(y) - \frac{1}{2\lambda_k} \|y - y^k\|^2 \Big\}. \tag{12}$$

The third algorithm, namely, the primal-dual PPA, is the application of the general proximal point method to the monotone operator corresponding to the convex-concave Lagrangian function, which generates a sequence $\{(X^k, y^k)\}$ by taking (X^{k+1}, y^{k+1}) to be an approximate solution to the following problem:

$$\min_{X \in \Re^{n_1 \times n_2}} \max_{y \in \mathcal{Q}^*} \Big\{ f_0(X) + \langle y, b - \mathcal{A}(X) \rangle + \frac{1}{2\lambda_k} \|X - X^k\|^2 - \frac{1}{2\lambda_k} \|y - y^k\|^2 \Big\}, \tag{13}$$

[1]The SVT algorithm has been applied to the corresponding regularized counterpart of problem (4) with noise in the revised version of [10].

where a sequence $\{\lambda_k\}$ satisfying (9) and an initial point $(X^0, y^0) \in \Re^{n_1 \times n_2} \times \mathcal{Q}^*$ are given.

A key issue in the PPAs mentioned above for solving (2) is how to solve the regularized problems (10), (12) and (13) efficiently. Based on the duality theory for convex programming, we develop the Moreau-Yosida regularization of the functions in (10) and (12) (see Section 2), which is important for the realizations of the general proximal point method for maximal monotone operators. It turns out that these algorithms require solving an inner sub-problem per iteration, which is a nonsmooth unconstrained convex optimization problem or a smooth convex optimization problem with simple constraints (see Section 3). Another aspect of the PPAs for solving (2) is how to formulate an implementable stopping criterion for approximately solving the inner sub-problems that still guarantees the global convergence and the rate of local convergence of these algorithms. In [42], Rockafellar introduced two criteria for inclusion problems with maximal monotone operators (see (39a) and (39b)). We will put these criteria in concrete and implementable forms in the context of problem (2) (see Remarks 3.1 and 3.4), and present comprehensive convergence results.

Besides the theoretic results on the PPAs for solving (2), we also investigate the performance of the aforementioned algorithms in which the inner sub-problems are solved by either the gradient projection method or the accelerated proximal gradient method. We design efficient implementations for these algorithms and present numerical results for solving randomly generated matrix completion problems and matrix completion problems arising from real applications. Our numerical results show that our algorithms perform favorably in comparison to recently proposed state-of-the-art algorithms in the literature including the SVT algorithm [10], the fixed point algorithm and the Bregman iterative algorithm [31], and an accelerated proximal gradient algorithm [45].

Our contribution in this paper is three fold. First, we provide a proximal point algorithmic framework for the NNM problem with complete convergence analysis. Our algorithms, which can handle conic constraints as well as linear equality constraints, are the applications of the general proximal point method to the primal, dual and primal-dual forms, respectively. We establish the connections between our algorithms and other algorithms that have been studied in the literature recently. In particular, the SVT algorithm [10] is just one gradient step of the primal PPA for solving the NNM problem (see Remark 3.2), and the Bregman iterative algorithm [31] is a special case of the dual PPA with a fixed parameter at each iteration for solving the NNM problem without second order cone constraints (see Remark 3.5). Second, we introduce checkable stopping criteria applied to our algorithms for solving (2). An important feature of the proposed stopping criteria is that they can be efficiently implemented in practice. These stopping criteria are extendable to more general cases. Third, our algorithms are proposed to solve the NNM problem with second order cone constraints, which are more applicable to practical problems with noisy data. Consequently, our algorithms are often able to obtain a more accurate solution when the practical problem is contaminated with noise.

The rest of this paper is organized as follows. In Section 2, we review and develop

some results related to the Moreau-Yosida regularization for subsequent discussions. In Section 3, we propose inexact PPAs for solving (2) in the primal, dual, and primal-dual forms and present comprehensive convergence results for our proposed algorithms. In Section 4, we discuss implementation issues of the PPAs, including the first-order methods applied to the inner sub-problems of the PPAs and the efficient computation of singular value decompositions. Numerical results for large matrix completion problems including the randomly generated examples and the real data from the Netflix Prize Contest are reported in Section 5. We make final conclusions and list possible directions for future research in Section 6.

2 The Moreau-Yosida regularization

For the sake of subsequent analysis, in this section we review and develop some results related to the Moreau-Yosida regularization.

Assume that $\mathcal{X}$ is a finite-dimensional real Hilbert space. Let $\phi : \mathcal{X} \to (-\infty, +\infty]$ be a proper, lower semicontinuous, convex function (cf. [41]). We wish to solve the following (possibly nondifferentiable) convex program:

$$\min_{x \in \mathcal{X}} \phi(x)\,. \tag{14}$$

For a given parameter $\lambda > 0$, we denote by Φ_λ the Moreau [33]-Yosida [48] regularization of ϕ associated with λ, which is defined by

$$\Phi_\lambda(x) = \min_{z \in \mathcal{X}} \Big\{ \phi(z) + \frac{1}{2\lambda}\|z - x\|^2 \Big\}, \quad x \in \mathcal{X}. \tag{15}$$

Let $p_\lambda(x)$ be the unique minimizer to (15), i.e.,

$$p_\lambda(x) = \arg\min_{z \in \mathcal{X}} \Big\{ \phi(z) + \frac{1}{2\lambda}\|z - x\|^2 \Big\}. \tag{16}$$

Then p_λ is called the proximal point mapping associated with ϕ.

We summarize below some well-known properties (see, e.g., [23]) of Φ_λ and p_λ without proofs. For additional properties, see, e.g., [23, 25].

Proposition 2.1. *Let Φ_λ and p_λ be defined as in (15) and (16), respectively. Then, the following properties hold for any $\lambda > 0$:*

(1). Φ_λ is a continuously differentiable convex function defined on $\mathcal{X}$ with its gradient being given by

$$\nabla \Phi_\lambda(x) = \frac{1}{\lambda}(x - p_\lambda(x)) \in \partial\phi(p_\lambda(x)), \tag{17}$$

where $\partial\phi$ is the subdifferential mapping of ϕ (cf. [41]). Moreover, $\nabla\Phi_\lambda(\cdot)$ is globally Lipschitz continuous with modulus $1/\lambda$.

(2). For any $x, x' \in \mathcal{X}$, one has

$$\langle p_\lambda(x) - p_\lambda(x'), x - x' \rangle \geq \|p_\lambda(x) - p_\lambda(x')\|^2.$$

It follows that $p_\lambda(\cdot)$ is globally Lipschitz continuous with modulus 1.

(3). The set of minimizers of (14) is exactly the set of minimizers of

$$\min_{x \in \mathcal{X}} \Phi_\lambda(x),$$

and x^* minimizes ϕ if and only if $\nabla \Phi_\lambda(x^*) = 0$ or equivalently $p_\lambda(x^*) = x^*$.

The following two examples on the Moreau-Yosida regularization are very useful for our subsequent development.

Example 2.1. (**The metric projection onto closed convex sets**) *Let $\mathcal{C} \subseteq \mathcal{X}$ be a closed convex set. Then, the metric projection of $x \in \mathcal{X}$ onto $\mathcal{C}$, denoted by $\Pi_{\mathcal{C}}(x)$, is the unique minimizer of the following convex program in the variable $u \in \mathcal{X}$:*

$$\min_{u \in \mathcal{X}} \Big\{ \chi_{\mathcal{C}}(u) + \frac{1}{2}\|u - x\|^2 \Big\},$$

where $\chi_{\mathcal{C}}$ is the indicator function over $\mathcal{C}$. Note that $\Pi_{\mathcal{C}}(\cdot)$ is exactly the proximal point mapping associated with $\chi_{\mathcal{C}}(\cdot)$. In particular, $\Pi_{\mathcal{K}^{m_2}}(\cdot)$ is the metric projector onto the second order cone $\mathcal{K}^{m_2}$. For any $x = (x_0; \bar{x}) \in \Re \times \Re^{m_2 - 1}$, by a direct calculation we have (cf. [16])

$$\Pi_{\mathcal{K}^{m_2}}(x) = \begin{cases} \frac{1}{2}\Big(1 + \frac{x_0}{\|\bar{x}\|}\Big)(\|\bar{x}\|; \bar{x}) & \text{if} \quad |x_0| < \|\bar{x}\|, \\ (x_0; \bar{x}) & \text{if} \quad \|\bar{x}\| \leq x_0, \\ 0 & \text{if} \quad \|\bar{x}\| \leq -x_0. \end{cases}$$

Example 2.2. (**The proximal mapping of the nuclear norm function**) *Let $\mathcal{P}_\lambda(\cdot)$ be the proximal point mapping associated with $f_0(\cdot)$. That is, for any X, $\mathcal{P}_\lambda(X)$ is the unique minimizer to*

$$S_\lambda(X) := \min_{Y \in \Re^{n_1 \times n_2}} \Big\{ f_0(Y) + \frac{1}{2\lambda}\|Y - X\|^2 \Big\}. \tag{18}$$

Then, by Proposition 2.1, we know that $S_\lambda(X)$ is continuously differentiable with

$$\nabla S_\lambda(X) = \frac{1}{\lambda}(X - \mathcal{P}_\lambda(X))$$

and

$$\langle \mathcal{P}_\lambda(X) - \mathcal{P}_\lambda(X'), X - X' \rangle \geq \|\mathcal{P}_\lambda(X) - \mathcal{P}_\lambda(X')\|^2, \quad \forall X,\, X' \in \Re^{n_1 \times n_2},$$

and thus $\mathcal{P}_\lambda(\cdot)$ is globally Lispchitz continuous with modulus 1.

For any given $X \in \Re^{n_1\times n_2}$, $\mathcal{P}_\lambda(X)$ admits an analytical solution. In fact, assume that X is of rank r and has the following singular value decomposition (SVD):

$$X = U\Sigma V^T, \quad \Sigma = \mathrm{diag}(\{\sigma_i\}_{i=1}^r), \tag{19}$$

where $U \in \Re^{n_1\times r}$ and $V \in \Re^{n_2\times r}$ have orthonormal columns, respectively, and the positive singular values σ_i are arranged in descending order. Then, from (18), one can easily derive (see, e.g., [10, 31]) [2] *that*

$$\mathcal{P}_\lambda(X) = U\mathrm{diag}(\max\{\sigma_i - \lambda, 0\})V^T, \tag{20}$$

and hence

$$S_\lambda(X) = \frac{1}{2\lambda}\Big(\|X\|^2 - \|\mathcal{P}_\lambda(X)\|^2\Big). \tag{21}$$

In order to develop the PPAs for solving (2), we need the following related concepts.

Let $l : \Re^{n_1\times n_2} \times \Re^m \to \Re$ be the ordinary Lagrangian function for (2) in the extended form:

$$l(X, y) := \begin{cases} f_0(X) + \langle y,\, b - \mathcal{A}(X)\rangle & \text{if } \ y \in \mathcal{Q}^*, \\ -\infty & \text{if } \ y \notin \mathcal{Q}^*. \end{cases} \tag{22}$$

The essential objective function in (2) is

$$f(X) := \sup_{y\in\Re^m} l(X, y) = \begin{cases} f_0(X) & \text{if } \ X \in \mathcal{F}_P, \\ +\infty & \text{if } \ X \notin \mathcal{F}_P, \end{cases} \tag{23}$$

while the essential objective function in (11) is

$$g(y) := \inf_{X\in\Re^{n_1\times n_2}} l(X, y) = \begin{cases} \inf_X\{f_0(X) + \langle y, b - \mathcal{A}(X)\rangle\} & \text{if } \ y \in \mathcal{Q}^*, \\ -\infty & \text{if } \ y \notin \mathcal{Q}^*. \end{cases} \tag{24}$$

In the following, we calculate the Moreau-Yosida regularizations of f and g, which play an important role in the analysis of the PPAs for solving (2).

We first calculate the Moreau-Yosida regularization of f. Let F_λ be the Moreau-Yosida regularization of f in (23) associated with λ, i.e.,

$$F_\lambda(X) = \min_{Y\in\Re^{n_1\times n_2}} \Big\{f(Y) + \frac{1}{2\lambda}\|Y - X\|^2\Big\}. \tag{25}$$

[2] Donald Goldfarb first reported the formula (20) at the "Foundations of Computational Mathematics Conference'08" held at the City University of Hong Kong, Hong Kong, China, June 2008.

Then, from (23), we obtain that

$$\begin{aligned} F_\lambda(X) &= \min_{Y\in\Re^{n_1\times n_2}} \sup_{y\in\Re^m} \Big\{ l(Y,y) + \frac{1}{2\lambda}\|Y-X\|^2 \Big\} \\ &= \sup_{y\in\Re^m} \min_{Y\in\Re^{n_1\times n_2}} \Big\{ l(Y,y) + \frac{1}{2\lambda}\|Y-X\|^2 \Big\} \\ &= \sup_{y\in\mathcal{Q}^*} \min_{Y\in\Re^{n_1\times n_2}} \Big\{ \|Y\|_* + \langle y, b-\mathcal{A}(Y)\rangle + \frac{1}{2\lambda}\|Y-X\|^2 \Big\}, \end{aligned} \tag{26}$$

where the interchange of $\min_Y$ and $\sup_y$ follows from the growth properties in Y [41, Theorem 37.3] and the third equality holds from (22). Note that

$$\begin{aligned} \Theta_\lambda(y;X) &:= \min_{Y\in\Re^{n_1\times n_2}} \Big\{ \|Y\|_* + \langle y,\, b-\mathcal{A}(Y)\rangle + \frac{1}{2\lambda}\|Y-X\|^2 \Big\} \\ &= \langle b,\, y\rangle + \frac{1}{2\lambda}\|X\|^2 - \frac{1}{2\lambda}\|X+\lambda\mathcal{A}^*(y))\|^2 + \min_Y \Big\{ \|Y\|_* + \frac{1}{2\lambda}\|Y-(X+\lambda\mathcal{A}^*(y))\|^2 \Big\} \\ &= \langle b,\, y\rangle + \frac{1}{2\lambda}\|X\|^2 - \frac{1}{2\lambda}\|\mathcal{P}_\lambda[X+\lambda\mathcal{A}^*(y)]\|^2. \end{aligned} \tag{27}$$

Thus

$$F_\lambda(X) = \sup_{y\in\mathcal{Q}^*} \Theta_\lambda(y;X). \tag{28}$$

By the Saddle Point Theorem (see, e.g., [41, Theorem 28.3]), combining (26) with Example 2.2, we know that $\mathcal{P}_\lambda[X+\lambda\mathcal{A}^*(y_\lambda(X))]$ is the unique solution to (25) for any $y_\lambda(X)$ such that

$$y_\lambda(X) \in \arg\sup_{y\in\mathcal{Q}^*} \Theta_\lambda(y;X), \tag{29}$$

where $\Theta_\lambda(y;X)$ is defined as in (27). Consequently, we have that

$$F_\lambda(X) = \Theta_\lambda(y_\lambda(X);X) \tag{30}$$

and

$$\nabla F_\lambda(X) = \frac{1}{\lambda}\big(X - \mathcal{P}_\lambda[X+\lambda\mathcal{A}^*(y_\lambda(X))]\big). \tag{31}$$

Next, we turn to the Moreau-Yosida regularization of g. Let G_λ be the Moreau-Yosida regularization of g associated with λ, i.e.,

$$G_\lambda(y) = \max_{z\in\Re^m} \Big\{ g(z) - \frac{1}{2\lambda}\|z-y\|^2 \Big\}. \tag{32}$$

Then, from (24), we obtain that

$$\begin{aligned} G_\lambda(y) &= \max_{z\in\mathcal{Q}^*} \inf_{X\in\Re^{n_1\times n_2}} \Big\{ \|X\|_* + \langle z, b-\mathcal{A}(X)\rangle - \frac{1}{2\lambda}\|z-y\|^2 \Big\} \\ &= \inf_{X\in\Re^{n_1\times n_2}} \max_{z\in\mathcal{Q}^*} \Big\{ \|X\|_* + \langle z, b-\mathcal{A}(X)\rangle - \frac{1}{2\lambda}\|z-y\|^2 \Big\} \\ &= \inf_{X\in\Re^{n_1\times n_2}} \Big\{ \|X\|_* + \frac{1}{2\lambda}\big(\|\Pi_{\mathcal{Q}^*}[y+\lambda(b-\mathcal{A}(X))]\|^2 - \|y\|^2\big) \Big\}, \end{aligned} \tag{33}$$

where the interchange of $\max_z$ and $\inf_X$ again follows from the growth properties in z [41, Theorem 37.3] and the third equality is due to Example 2.1. By the Saddle Point Theorem again, we also know that $\Pi_{\mathcal{Q}^*}[y+\lambda(b-\mathcal{A}(X_\lambda(y)))]$ is the unique optimal solution to (32) for any $X_\lambda(y)$ satisfying

$$X_\lambda(y) \in \arg\inf_{X\in\Re^{n_1\times n_2}} \Big\{\|X\|_* + \Psi_\lambda(X;y)\Big\}, \tag{34}$$

where $\Psi_\lambda(X;y)$ is defined by

$$\Psi_\lambda(X;y) := \frac{1}{2\lambda}\big(\|\Pi_{\mathcal{Q}^*}[y+\lambda(b-\mathcal{A}(X))]\|^2 - \|y\|^2\big). \tag{35}$$

Consequently, we have that

$$G_\lambda(y) = \|X_\lambda(y)\|_* + \Psi_\lambda(X_\lambda(y);y) \tag{36}$$

and

$$\nabla G_\lambda(y) = \frac{1}{\lambda}\big(\Pi_{\mathcal{Q}^*}\big[y+\lambda(b-\mathcal{A}(X_\lambda(y)))\big] - y\big), \tag{37}$$

where $X_\lambda(y)$ satisfies (34).

3 The proximal point algorithm in three forms

In this section, we present the proximal point algorithm for solving (2) in the primal, dual and primal-dual forms.

Our approach is based on the classic idea of the proximal point method for solving inclusion problems with maximal monotone operators [42, 43]. We briefly review it below. Let $\mathcal{X}$ be a finite-dimensional real Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and $\mathcal{T}:\mathcal{X}\to\mathcal{X}$ be a, possibly multi-valued, maximal monotone operator. Given $x^0\in\mathcal{X}$, the idea of the proximal point method for solving the inclusion problem $0\in\mathcal{T}(x)$ is to solve iteratively a sequence of regularized inclusion problems:

$$x^{k+1} \text{ approximately solves } 0\in\mathcal{T}(x)+\lambda_k^{-1}(x-x^k),$$

or equivalently,

$$x^{k+1} \approx p_{\lambda_k}(x^k) := (I+\lambda_k\mathcal{T})^{-1}(x^k), \tag{38}$$

where the given sequence $\{\lambda_k\}$ satisfies (9). Two convergence criteria for (38) introduced by Rockafellar [42] are as follows:

$$\|x^{k+1}-p_{\lambda_k}(x^k)\| \;\le\; \varepsilon_k, \quad \varepsilon_k>0,\ \sum_{k=0}^{\infty}\varepsilon_k<\infty, \tag{39a}$$

$$\|x^{k+1}-p_{\lambda_k}(x^k)\| \;\le\; \delta_k\|x^{k+1}-x^k\|, \quad \delta_k>0,\ \sum_{k=0}^{\infty}\delta_k<\infty. \tag{39b}$$

In [42], Rockafellar showed that under mild assumptions, condition (39a) ensures the global convergence of $\{x^k\}$, i.e., the sequence $\{x^k\}$ converges to a particular solution $\overline{x}$ to $0 \in \mathcal{T}(x)$, and if in addition (39b) holds and $\mathcal{T}^{-1}$ is Lipschitz continuous at the origin, then the sequence $\{x^k\}$ locally converges at a linear rate whose ratio is, roughly speaking, proportional to $1/\lambda_\infty$ and in particular, if $\lambda_\infty = +\infty$, the convergence is superlinear. For details on the convergence of the general proximal point method, see [42, Theorem 1 & 2].

The proximal point algorithm in three different forms studied in this paper corresponds respectively to the one applied to the maximal monotone operators $\mathcal{T}_f$, $\mathcal{T}_g$ and $\mathcal{T}_l$, which can be defined as in Rockafellar [43] by:

$$\begin{cases} \mathcal{T}_f(X) &= \{Y \in \Re^{n_1 \times n_2} \ : \ Y \in \partial f(X)\}, & X \in \Re^{n_1 \times n_2}, \\ \mathcal{T}_g(y) &= \{z \in \Re^m \ : \ -z \in \partial g(y)\}, & y \in \Re^m, \\ \mathcal{T}_l(X,y) &= \{(Y,z) \in \Re^{n_1 \times n_2} \times \Re^m \ : \ (Y,-z) \in \partial l(X,y)\}, & (X,y) \in \Re^{n_1 \times n_2} \times \Re^m. \end{cases}$$

From the definition of $\mathcal{T}_f$, we can easily see that for any $Y \in \Re^{n_1 \times n_2}$,

$$\mathcal{T}_f^{-1}(Y) = \arg\min_{X \in \Re^{n_1 \times n_2}} \Big\{ f(X) - \langle Y, X \rangle \Big\}.$$

Similarly, we have that for any $z \in \Re^m$,

$$\mathcal{T}_g^{-1}(z) = \arg\max_{y \in \Re^m} \Big\{ g(y) + \langle z, y \rangle \Big\}$$

and for any $(Y,z) \in \Re^{n_1 \times n_2} \times \Re^m$,

$$\mathcal{T}_l^{-1}(Y,z) = \arg\min_{X \in \Re^{n_1 \times n_2}} \max_{y \in \Re^m} \Big\{ l(X,y) - \langle Y, X \rangle + \langle z, y \rangle \Big\}.$$

3.1 The primal form

In this subsection, we shall present the proximal point algorithm applied to the primal form of the NNM problem (2).

Given $X^0 \in \Re^{n_1 \times n_2}$, the exact primal PPA can be described as

$$X^{k+1} = p_{\lambda_k}(X^k), \tag{40}$$

where $p_{\lambda_k}(X^k)$ is defined by

$$p_{\lambda_k}(X^k) := (I + \lambda_k \mathcal{T}_f)^{-1}(X^k) = \arg\min_{X \in \Re^{n_1 \times n_2}} \Big\{ f(X) + \frac{1}{2\lambda_k} \|X - X^k\|^2 \Big\} \tag{41}$$

and the sequence $\{\lambda_k\}$ satisfying (9) is given. It can be seen easily from (40), (41), and (17) that

$$X^{k+1} = X^k - \lambda_k \nabla F_{\lambda_k}(X^k)\,. \tag{42}$$

From the computational point of view, the cost of computing the exact solution $p_{\lambda_k}(X^k)$ could be prohibitive. This motivates to consider an inexact primal PPA. Combining (31) with (29), we can introduce an inexact primal PPA to solve (2), which has the following template:

The Primal PPA. Given a tolerance $\varepsilon > 0$. Input $X^0 \in \Re^{n_1 \times n_2}$ and $\lambda_0 > 0$. Set $k := 0$. Iterate:

Step 1. Find an approximate maximizer

$$\mathcal{Q}^* \ni y^{k+1} \approx \arg\sup_{y \in \Re^m} \Big\{\theta_k(y) := \Theta_{\lambda_k}(y; X^k) - \chi_{\mathcal{Q}^*}(y)\Big\}, \tag{43}$$

where $\Theta_{\lambda_k}(y; X^k)$ is defined as in (27).

Step 2. Compute

$$X^{k+1} \;=\; \mathcal{P}_{\lambda_k}[X^k + \lambda_k \mathcal{A}^*(y^{k+1})].$$

Step 3. If $\|(X^k - X^{k+1})/\lambda_k\| \le \varepsilon$; stop; else; update λ_k such that (9) holds; end.

In the primal PPA stated above, we introduce the following stopping criteria to terminate (43):

$$\sup \theta_k - \theta_k(y^{k+1}) \;\le\; \frac{\varepsilon_k^2}{2\lambda_k}, \quad \varepsilon_k > 0, \; \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{44a}$$

$$\sup \theta_k - \theta_k(y^{k+1}) \;\le\; \frac{\delta_k^2}{2\lambda_k}\|X^{k+1} - X^k\|^2, \quad \delta_k > 0, \; \sum_{k=0}^{\infty} \delta_k < \infty, \tag{44b}$$

$$\mathrm{dist}(0, \partial\theta_k(y^{k+1})) \le \delta_k'\|X^{k+1} - X^k\|, \quad 0 \le \delta_k' \to 0. \tag{45}$$

It should be noted that one has

$$F_{\lambda_k}(X^k) = \sup \theta_k, \quad \theta_k(y^{k+1}) = \Theta_{\lambda_k}(y^{k+1}; X^k).$$

Remark 3.1. *Note that in the stopping criteria (44a) and (44b), the unknown value* $\sup \theta_k$ *can be replaced by any of its upper bounds converging to it. One particular choice is to let* $\hat{\theta}_k := \|\widehat{X}^{k+1}\|_* + (1/2\lambda_k)\|\widehat{X}^{k+1} - X^k\|^2$, *where* $\widehat{X}^{k+1} := \Pi_{\mathcal{F}_P}(X^{k+1})$. *It follows from (25) and (26) that*

$$\hat{\theta}_k = \|\widehat{X}^{k+1}\|_* + (1/2\lambda_k)\|\widehat{X}^{k+1} - X^k\|^2 \ge F_{\lambda_k}(X^k) = \sup \theta_k.$$

Consequently, the stopping criteria (44a) and (44b) can be replaced by the following im-

plementable conditions:

$$\hat{\theta}_k - \theta_k(y^{k+1}) \;\leq\; \frac{\varepsilon_k^2}{2\lambda_k}, \quad \varepsilon_k > 0,\ \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{46a}$$

$$\hat{\theta}_k - \theta_k(y^{k+1}) \;\leq\; \frac{\delta_k^2}{2\lambda_k}\|X^{k+1} - X^k\|^2, \quad \delta_k > 0,\ \sum_{k=0}^{\infty} \delta_k < \infty. \tag{46b}$$

We emphasize that for the matrix completion problem (see (79)), it is easy to compute the projection $\Pi_{\mathcal{F}_P}(\cdot)$ onto the feasible set.

The following result establishes the relation between the estimates (44)-(45) and (39), which plays a key role in order to apply the convergence results in [42, Theorem 1] and [42, Theorem 2] for the general proximal point method to the primal PPA. Our proof closely follows the idea used in [43, Proposition 6].

Proposition 3.1. *Let p_{λ_k} be given as in (41), Θ_{λ_k} be given as in (27), and $X^{k+1} = \mathcal{P}_{\lambda_k}[X^k + \lambda_k \mathcal{A}^*(y^{k+1})]$. Then, one has*

$$\|X^{k+1} - p_{\lambda_k}(X^k)\|^2/(2\lambda_k) \leq F_{\lambda_k}(X^k) - \theta_k(y^{k+1}). \tag{47}$$

Proof. Since

$$\nabla_X \Theta_{\lambda_k}(y^{k+1}; X^k) = \lambda_k^{-1}(X^k - X^{k+1}), \tag{48}$$

we obtain from the convexity of $\Theta_\lambda(y; X)$ in X that the following inequality is valid for any $Y \in \Re^{n_1 \times n_2}$:

$$\begin{aligned}
&\Theta_{\lambda_k}(y^{k+1}; X^k) + \langle \lambda_k^{-1}(X^k - X^{k+1}), Y - X^k \rangle \\
\leq\;& \Theta_{\lambda_k}(y^{k+1}; Y) \;\leq\; \sup_{y \in \mathcal{Q}^*} \Big\{ \Theta_{\lambda_k}(y; Y) \Big\} \\
=\;& \sup_{y \in \Re^m} \min_{X \in \Re^{n_1 \times n_2}} \Big\{ l(X, y) + \frac{1}{2\lambda_k}\|X - Y\|^2 \Big\} = \min_{X \in \Re^{n_1 \times n_2}} \sup_{y \in \Re^m} \Big\{ l(X, y) + \frac{1}{2\lambda_k}\|X - Y\|^2 \Big\} \\
=\;& \min_{X \in \Re^{n_1 \times n_2}} \Big\{ f(X) + \frac{1}{2\lambda_k}\|X - Y\|^2 \Big\} \leq f(p_{\lambda_k}(X^k)) + \frac{1}{2\lambda_k}\|p_{\lambda_k}(X^k) - Y\|^2.
\end{aligned} \tag{49}$$

It follows from (30) and (25) that

$$\begin{aligned}
\sup_{y \in \mathcal{Q}^*} \Big\{ \Theta_{\lambda_k}(y; X^k) \Big\} = F_{\lambda_k}(X^k) \;&=\; \min_{X \in \Re^{n_1 \times n_2}} \Big\{ f(X) + \frac{1}{2\lambda_k}\|X - X^k\|^2 \Big\} \\
&=\; f(p_{\lambda_k}(X^k)) + \frac{1}{2\lambda_k}\|p_{\lambda_k}(X^k) - X^k\|^2,
\end{aligned}$$

which, together with (49) and the fact that $\theta_k(y^{k+1}) = \Theta_{\lambda_k}(y^{k+1}; X^k)$, implies that

$$\begin{aligned}
&F_{\lambda_k}(X^k) - \theta_k(y^{k+1}) \\
\geq\;& \big[\|p_{\lambda_k}(X^k) - X^k\|^2 - \|p_{\lambda_k}(X^k) - Y\|^2 - 2\langle X^{k+1} - X^k, Y - X^k \rangle\big]/(2\lambda_k) \\
=\;& \big[2\langle p_{\lambda_k}(X^k) - X^{k+1}, Y - X^k \rangle - \|Y - X^k\|^2\big]/(2\lambda_k).
\end{aligned} \tag{50}$$

Since this holds for all $Y \in \Re^{n_1 \times n_2}$, and

$$\|X^{k+1} - p_{\lambda_k}(X^k)\|^2 = \max_{Y \in \Re^{n_1 \times n_2}} \Big\{ 2\langle p_{\lambda_k}(X^k) - X^{k+1}, Y - X^k \rangle - \|Y - X^k\|^2 \Big\},$$

we can obtain the estimate (47) by taking the maximum of (50). This completes the proof. □

For convergence analysis, we need the following condition for the NNM problem (2):

$$\begin{cases} \{\mathcal{A}_i\}_{i=1}^{m_1} \text{ are linearly independent and } \exists\, \widehat{X} \in \Re^{n_1 \times n_2} \text{ such that} \\ \mathcal{A}_i(\widehat{X}) = b_i,\ i = 1, \ldots, m_1 \text{ and } (\mathcal{A}_i(\widehat{X}) - b_i)_{i=m_1+1}^{m} \in \mathrm{int}(\mathcal{K}^{m_2}), \end{cases} \tag{51}$$

where "$\mathrm{int}(\mathcal{K}^{m_2})$" denotes the interior of $\mathcal{K}^{m_2}$.

We now state the global convergence and local linear convergence of the primal PPA for solving problem (2).

Theorem 3.1. (**Global Convergence**) *Assume $\mathcal{F}_P \neq \emptyset$. Let the primal PPA be executed with stopping criterion (44a). Then the generated sequence $\{X^k\}$ is bounded and $X^k \to \overline{X}$, where $\overline{X}$ is some optimal solution to problem (2), and $\{y^k\}$ is asymptotically minimizing for problem (11).*

If problem (2) satisfies condition (51), then the sequence $\{y^k\}$ is also bounded, and any of its accumulation points is an optimal solution to problem (2).

Proof. Since the nuclear norm function is coercive, together with $\mathcal{F}_P \neq \emptyset$ due to our hypothesis, we conclude that there exists at least an optimal solution to problem (2). Moreover, Proposition 3.1 shows that (44a) implies the more general criterion (39a) for $\mathcal{T}_f$. It follows from [42, Theorem 1] that the sequence $\{X^k\}$ is bounded and converges to a solution $\overline{X}$ to $0 \in \mathcal{T}_f(\overline{X})$, i.e., a particular optimal solution to problem (2). The remainder of the conclusions follows from the proof of [43, Theorem 4] without difficulty. We omit it here. □

Theorem 3.2. (**Local Convergence**) *Assume $\mathcal{F}_P \neq \emptyset$. Let the primal PPA be executed with stopping criteria (44a) and (44b). If $\mathcal{T}_f^{-1}$ is Lipschitz continuous at the origin with modulus a_f, then $X^k \to \overline{X}$, where $\overline{X}$ is the unique optimal solution to problem (2), and*

$$\|X^{k+1} - \overline{X}\| \leq \eta_k \|X^k - \overline{X}\|, \text{ for all } k \text{ sufficiently large,}$$

where

$$\eta_k = [a_f(a_f^2 + \lambda_k^2)^{-1/2} + \delta_k](1 - \delta_k)^{-1} \to \eta_\infty = a_f(a_f^2 + \lambda_\infty^2)^{-1/2} < 1.$$

Moreover, the conclusions of Theorem 3.1 about $\{y^k\}$ are valid.

If in addition to (44b) and the condition on $\mathcal{T}_f^{-1}$, one has (45) and $\mathcal{T}_l^{-1}$ is Lipschitz continuous at the origin with modulus a_l ($\geq a_f$), then $X^k \to \overline{X}$, where $\overline{X}$ is the unique optimal solution to problem (2), and one has

$$\|y^{k+1} - \overline{y}\| \leq \eta_k' \|X^{k+1} - X^k\|, \text{ for all } k \text{ sufficiently large,}$$

where $\eta_k' = a_l(1 + \delta_k')/\lambda_k \to \eta_\infty' = a_l/\lambda_\infty$.

Proof. The proof can be obtained by following the ideas used in the proof of [43, Theorem 5] combining with Proposition 3.1. We omit it here. □

Remark 3.2. *Recall that the Tikhonov regularization method solves a sequence of subproblems of the form:*

$$\min\left\{\|X\|_* + \frac{1}{2\lambda_k}\|X\|^2 \ : \ \mathcal{A}(X) \in b + \mathcal{Q},\ X \in \Re^{n_1\times n_2}\right\}$$

with the positive sequence $\{\lambda_k\} \to +\infty$. *The primal PPA is to replace the term* $\frac{1}{2\lambda_k}\|X\|^2$ *in the Tikhonov regularization method by* $\frac{1}{2\lambda_k}\|X - X^k\|^2$. *The benefit of making this change is that in the primal PPA the sequence* $\{\lambda_k\}$ *is no longer required to tend to* $+\infty$.

From the exact primal PPA, we can see that if $X^0 = 0$ *and* $\lambda_0 = \lambda^{-1} > 0$, *then* X^1 *solves the following regularized problem:*

$$\min\left\{\lambda\|X\|_* + \frac{1}{2}\|X\|^2 \ : \ \mathcal{A}(X) \in b + \mathcal{Q},\ X \in \Re^{n_1\times n_2}\right\}. \tag{52}$$

That is, the SVT algorithm considered in [10] solves (52) by applying the gradient method to its dual problem (43) and thus it is just one gradient step of the exact primal PPA, i.e., $k = 0$ *in (42), with* $X^0 = 0$.

3.2 The dual form

In this subsection, we shall discuss the proximal point algorithm applied to the dual problem (11). This algorithm solves the dual problem (11).

Given $y^0 \in \Re^m$, the exact dual PPA can be described as

$$y^{k+1} = p_{\lambda_k}(y^k), \tag{53}$$

where $p_{\lambda_k}(y^k)$ is defined by

$$p_{\lambda_k}(y^k) = (I + \lambda_k \mathcal{T}_g)^{-1}(y^k) = \arg\max_{y\in\Re^m}\left\{g(y) - \frac{1}{2\lambda_k}\|y - y^k\|^2\right\}, \tag{54}$$

and the sequence $\{\lambda_k\}$ satisfying (9) is given. It follows from (53), (54), and (17) that

$$y^{k+1} = y^k + \lambda_k \nabla G_{\lambda_k}(y^k)\,.$$

Just like the primal PPA, it is impractical to solve (54) exactly. So we consider an inexact dual PPA in which (54) is solved approximately. In view of (37) and (34), we can state the following inexact dual PPA to solve (11):

The Dual PPA. Given a tolerance $\varepsilon > 0$. Input $y^0 \in \Re^m$ and $\lambda_0 > 0$. Set $k := 0$. Iterate:

Step 1. Find an approximate minimizer

$$X^{k+1} \approx \arg \inf_{X \in \Re^{n_1 \times n_2}} \Big\{ \psi_k(X) := \|X\|_* + \Psi_{\lambda_k}(X; y^k) \Big\}, \tag{55}$$

where $\Psi_{\lambda_k}(X; y^k)$ is defined as in (35).

Step 2. Compute

$$y^{k+1} \;=\; \Pi_{\mathcal{Q}^*}\big[y^k + \lambda_k(b - \mathcal{A}(X^{k+1}))\big].$$

Step 3. If $\|(y^k - y^{k+1})/\lambda_k\| \le \varepsilon$; stop; else; update λ_k such that (9) holds; end.

In the dual PPA, we shall consider the following stopping criteria introduced by Rockafellar [42, 43] to terminate (55):

$$\psi_k(X^{k+1}) - \inf \psi_k \;\le\; \frac{\varepsilon_k^2}{2\lambda_k}, \quad \varepsilon_k > 0, \; \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{56a}$$

$$\psi_k(X^{k+1}) - \inf \psi_k \;\le\; \frac{\delta_k^2}{2\lambda_k}\|y^{k+1} - y^k\|^2, \quad \delta_k > 0, \; \sum_{k=0}^{\infty} \delta_k < \infty, \tag{56b}$$

$$\mathrm{dist}(0, \partial\psi_k(X^{k+1})) \le \delta_k'\|y^{k+1} - y^k\|, \quad 0 \le \delta_k' \to 0. \tag{57}$$

It follows from (26) and (55) that

$$G_{\lambda_k}(y^k) = \inf \psi_k, \quad \psi_k(X^{k+1}) = \|X^{k+1}\|_* + \Psi_{\lambda_k}(X^{k+1}; y^k).$$

Remark 3.3. *Note that the dual PPA stated above actually corresponds to the method of multipliers considered in [43, Section 4] applied to problem (2).*

Remark 3.4. *The unknown value* $\inf \psi_k$ *used in stopping criteria (56a) and (56b) can be replaced by any of its lower bounds converging to it. For example, one can choose* $\check{\psi}_k := \langle b, \check{y}^{k+1}\rangle - (1/2\lambda_k)\|\check{y}^{k+1} - y^k\|^2$, *where* $\check{y}^{k+1} := y^{k+1}$ *if* $\|\mathcal{A}^*(y^{k+1})\|_2 \le 1$ *and otherwise* $\check{y}^{k+1} := y^{k+1}/\|\mathcal{A}^*(y^{k+1})\|_2$. *Then, by using the fact that* $\check{y}^{k+1}$ *is feasible to the dual problem (11), one can obtain from (24), (32), and (33) that*

$$\check{\psi}_k = \langle b, \check{y}^{k+1}\rangle - \frac{1}{2\lambda_k}\|\check{y}^{k+1} - y^k\|^2 = g(\check{y}^{k+1}) - \frac{1}{2\lambda_k}\|\check{y}^{k+1} - y^k\|^2 \le G_{\lambda_k}(y^k) = \inf \psi_k.$$

Therefore, the stopping criteria (56a) and (56b) can be replaced by the following imple-

mentable conditions:

$$\psi_k(X^{k+1}) - \check{\psi}_k \;\leq\; \frac{\varepsilon_k^2}{2\lambda_k}, \quad \varepsilon_k > 0,\ \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{58a}$$

$$\psi_k(X^{k+1}) - \check{\psi}_k \;\leq\; \frac{\delta_k^2}{2\lambda_k}\|y^{k+1} - y^k\|^2, \quad \delta_k > 0,\ \sum_{k=0}^{\infty} \delta_k < \infty. \tag{58b}$$

We are ready to state the global convergence and local linear convergence of the dual PPA for solving problem (2).

Theorem 3.3. (**Global Convergence**) *Let the dual PPA be executed with stopping criterion (56a). If condition (51) holds for problem (2), then the sequence $\{y^k\} \subset \mathcal{Q}^*$ generated by the dual PPA is bounded and $y^k \to \overline{y}$, where $\overline{y}$ is some optimal solution to problem (11). Moreover, the sequence $\{X^k\}$ is also bounded, and any of its accumulation points is an optimal solution to problem (2).*

Proof. This corresponds to [43, Theorem 4]. □

Theorem 3.4. (**Local Convergence**) *Let the dual PPA be executed with stopping criterion (56a) and (56b). Assume that condition (51) holds for problem (2). If $\mathcal{T}_g^{-1}$ is Lipschitz continuous at the origin with modulus a_g, then $y^k \to \overline{y}$, where $\overline{y}$ is the unique optimal solution to problem (11), and*

$$\|y^{k+1} - \overline{y}\| \leq \eta_k \|y^k - \overline{y}\|, \text{ for all } k \text{ sufficiently large},$$

where

$$\eta_k = [a_g(a_g^2 + \lambda_k^2)^{-1/2} + \delta_k](1 - \delta_k)^{-1} \to \eta_\infty = a_g(a_g^2 + \lambda_\infty^2)^{-1/2} < 1.$$

Moreover, the conclusions of Theorem 3.3 about $\{X^k\}$ are valid.

If in addition to (56b) and the condition on $\mathcal{T}_g^{-1}$, one has (57) and $\mathcal{T}_l^{-1}$ is Lipschitz continuous at the origin with modulus a_l ($\geq a_g$), then $y^k \to \overline{y}$, where $\overline{y}$ is the unique optimal solution to problem (11), and one has

$$\|X^{k+1} - \overline{X}\| \leq \eta_k' \|y^{k+1} - y^k\|, \text{ for all } k \text{ sufficiently large},$$

where $\eta_k' = a_l(1 + \delta_k')/\lambda_k \to \eta_\infty' = a_l/\lambda_\infty$.

Proof. The conclusions can be obtained by applying the results of [43, Theorem 5] to problem (2). □

Remark 3.5. *From the dual PPA, we observe that if $y^0 = 0$, then*

$$y^1 = \lambda_0 \Pi_{\mathcal{Q}^*}[b - \mathcal{A}(X^1)],$$

where X^1 (approximately) solves the following penalized problem of (2):

$$\min\left\{\frac{1}{2}\|\Pi_{\mathcal{Q}^*}[b-\mathcal{A}(X)]\|^2+\lambda_0{}^{-1}\|X\|_* \ :\ X\in\Re^{n_1\times n_2}\right\}. \tag{59}$$

For the special case of (2) with equality constraints only, then with $y^0=0$,

$$y^1=\lambda_0(b-\mathcal{A}(X^1)),$$

where X^1 (approximately) solves the following penalized problem:

$$\min\left\{\frac{1}{2}\|\mathcal{A}(X)-b\|^2+\lambda_0^{-1}\|X\|_* \ :\ X\in\Re^{n_1\times n_2}\right\}. \tag{60}$$

Again, this says that y^1 is the result for one outer gradient iteration of the dual PPA. The problem (60) corresponds to the nuclear norm regularized linear least squares problems considered in [31, 45].

The Bregman iterative method considered in [31] for solving problem (2) with equality constraints only can be described as:

$$\begin{cases} b^{k+1}=b^k+(b-\mathcal{A}(X^k)) \\ X^{k+1}=\arg\min\limits_{X\in\Re^{n_1\times n_2}}\left\{\frac{1}{2}\|\mathcal{A}(X)-b^{k+1}\|^2+\mu\|X\|_*\right\} \end{cases} \tag{61}$$

for some fixed $\mu>0$. By noting that $b^{k+1}=\mu y^{k+1}$ with $\mu=\lambda_k{}^{-1}$, we know that in this case the Bregman iterative method [31] is actually a special case of the exact dual PPA with $\lambda_k\equiv\mu^{-1}$.

3.3 The primal-dual form

In this subsection, we shall discuss the proximal point algorithm applied to compute a saddle point of the Lagrangian function l.

Given $(X^0,y^0)\in\Re^{n_1\times n_2}\times\Re^m$, the exact primal-dual PPA can be described as

$$(X^{k+1},y^{k+1})=p_{\lambda_k}(X^k,y^k), \tag{62}$$

where $p_{\lambda_k}(X^k,y^k)$ is defined by

$$\begin{aligned} p_{\lambda_k}(X^k,y^k) &= (I+\lambda_k\mathcal{T}_l)^{-1}(X^k,y^k) \\ &= \arg\min_{X\in\Re^{n_1\times n_2}}\max_{y\in\Re^m}\left\{l(X,y)+\frac{1}{2\lambda_k}\|X-X^k\|^2-\frac{1}{2\lambda_k}\|y-y^k\|^2\right\}, \end{aligned} \tag{63}$$

and the sequence $\{\lambda_k\}$ satisfying (9) is given.

We see that in the k-th step of the primal-dual PPAs, one needs to obtain the saddle point of $l_k(X,y)$, where $l_k(X,y)$ is defined by

$$l_k(X,y):=l(X,y)+\frac{1}{2\lambda_k}\|X-X^k\|^2-\frac{1}{2\lambda_k}\|y-y^k\|^2.$$

By the Saddle Point Theorem, it can be easily verified that in order that (X^{k+1}, y^{k+1}) is the saddle point of $l_k(X, y)$, it is sufficient and necessary that one of the following statements is valid:

(i). $X^{k+1} = \mathcal{P}_{\lambda_k}[X + \lambda_k \mathcal{A}^*(y^{k+1})]$, where y^{k+1} satisfies

$$y^{k+1} = \arg\max_{y \in \mathcal{Q}^*} \Big\{ \Theta_{\lambda_k}(y; X^k) - \frac{1}{2\lambda_k}\|y - y^k\|^2 \Big\}.$$

(ii). $y^{k+1} = \Pi_{\mathcal{Q}^*}[y^k + \lambda_k(b - \mathcal{A}(X^{k+1}))]$, where X^{k+1} satisfies

$$X^{k+1} = \arg\min_{X \in \Re^{n_1 \times n_2}} \Big\{ \|X\|_* + \Psi_{\lambda_k}(X; y^k) + \frac{1}{2\lambda_k}\|X - X^k\|^2 \Big\}.$$

The above results lead to two versions of the inexact primal-dual PPA. The first version of the inexact primal-dual PPA based on part (i) can be stated as follows :

The Primal-Dual PPA-I. Given a tolerance $\varepsilon > 0$. Input $X^0 \in \Re^{n_1 \times n_2}$, $y^0 \in \Re^m$, and $\lambda_0 > 0$. Set $k := 0$. Iterate:

Step 1. Approximately find the unique maximizer

$$\mathcal{Q}^* \ni y^{k+1} \approx \arg\max_{y \in \Re^m} \Big\{ \theta_k(y) := \Theta_{\lambda_k}(y; X^k) - \chi_{\mathcal{Q}^*}(y) - \frac{1}{2\lambda_k}\|y - y^k\|^2 \Big\}. \tag{64}$$

Step 2. Compute

$$X^{k+1} \;=\; \mathcal{P}_{\lambda_k}[X^k + \lambda_k \mathcal{A}^*(y^{k+1})].$$

Step 3. If $\|(X^k - X^{k+1})/\lambda_k\| \le \varepsilon$; stop; else; update λ_k such that (9); end.

In the primal-dual PPA-I stated above, one does not need to solve problem (64) exactly. Two stopping criteria to terminate them are treated as follows:

$$\mathrm{dist}(0, \partial\theta_k(y^{k+1})) \;\le\; \frac{\varepsilon_k}{\lambda_k}, \quad \varepsilon_k > 0, \; \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{65a}$$

$$\mathrm{dist}(0, \partial\theta_k(y^{k+1})) \;\le\; \frac{\delta_k}{\lambda_k}\|(X^{k+1}, y^{k+1}) - (X^k, y^k)\|, \quad \delta_k > 0, \; \sum_{k=0}^{\infty} \delta_k < \infty. \tag{65b}$$

We next apply the general convergence results [42] to the primal-dual PPA-I. The following proposition is crucial for this purpose.

Proposition 3.2. *Let p_{λ_k} be given by (63) and $X^{k+1} = \mathcal{P}_{\lambda_k}[X^k + \lambda_k \mathcal{A}^*(y^{k+1})]$. Then, one has*

$$\|(X^{k+1}, y^{k+1}) - p_{\lambda_k}(X^k, y^k)\| \le \lambda_k \mathrm{dist}(0, \partial\theta_k(y^{k+1})). \tag{66}$$

Proof. We first note that

$$\partial\theta_k(y^{k+1}) = \partial\phi_k(y^{k+1}) - \lambda_k^{-1}(y^{k+1} - y^k),$$

where $\phi_k(y) = \Theta_{\lambda_k}(y; X^k) - \chi_{\mathcal{Q}^*}(y)$. Therefore, for any $w \in \partial\theta_k(y^{k+1})$, one has $w + \lambda_k^{-1}(y^{k+1} - y^k) \in \partial\phi_k(y^{k+1})$, and hence $w + \lambda_k^{-1}(y^{k+1} - y^k) \in \partial_y l(X^{k+1}, y^{k+1})$. On the other hand, from (23) and (31), we have that $\lambda_k^{-1}(X^k - X^{k+1}) \in \partial_X l(X^{k+1}, y^{k+1})$. Consequently, we obtain that

$$(\lambda_k^{-1}(X^k - X^{k+1}), -w + \lambda_k^{-1}(y^k - y^{k+1})) \in \mathcal{T}_l(X^{k+1}, y^{k+1}),$$

or equivalently, $(X^k, -\lambda_k w + y^k) \in (I + \lambda_k \mathcal{T}_l)(X^{k+1}, y^{k+1})$, which implies that $(X^{k+1}, y^{k+1}) = p_{\lambda_k}(X^k, -\lambda_k w + y^k)$. Since p_{λ_k} is nonexpansive [42], we have

$$\|(X^{k+1}, y^{k+1}) - p_{\lambda_k}(X^k, y^k)\| \le \|(X^k, -\lambda_k w + y^k) - (X^k, y^k)\| \le \lambda_k \|w\|.$$

Since this holds for any $w \in \partial\theta_k(y^{k+1})$, we obtain the estimate (66). This completes the proof. □

We are ready to state the convergence results for the primal-dual PPA-I.

Theorem 3.5. (**Global Convergence**) *Assume that $\mathcal{F}_P \neq \emptyset$ and condition (51) holds for problem (2). Let the primal-dual PPA-I be executed with stopping criterion (65a). Then, the generated sequence $\{(X^k, y^k)\} \subset \Re^{n_1 \times n_2} \times \mathcal{Q}^*$ is bounded, and $(X^k, y^k) \to (\overline{X}, \overline{y})$, where $\overline{X}$ is an optimal solution to problem (2) and $\overline{y}$ is an optimal solution to problem (11).*

Proof. Combining Proposition 3.2 with [42, Theorem 1], we know that (X^k, y^k) converges to some $(\overline{X}, \overline{y})$ such that $(0, 0) \in \mathcal{T}_l(\overline{X}, \overline{y})$, which means that $(\overline{X}, \overline{y})$ is a saddle point of the Lagrangian function l and hence $\overline{X}$ is an optimal solution to problem (2) and $\overline{y}$ is an optimal solution to problem (11). This completes the proof. □

Theorem 3.6. (**Local Convergence**) *Assume that $\mathcal{F}_P \neq \emptyset$ and condition (51) holds for problem (2). Let the primal-dual PPA-I be executed with stopping criterion (65a) and (65b). If $\mathcal{T}_l^{-1}$ is Lipschitz continuous at the origin with modulus $a_l > 0$, then $\{(X^k, y^k)\}$ is bounded and $(X^k, y^k) \to (\overline{X}, \overline{y})$, where $\overline{X}$ is the unique optimal solution to problem (2) and $\overline{y}$ is the unique optimal solution to problem (11). Furthermore, one has*

$$\|(X^{k+1} - y^{k+1}) - (\overline{X}, \overline{y})\| \le \eta_k \|(X^k, y^k) - (\overline{X}, \overline{y})\|, \text{ for all } k \text{ sufficiently large,}$$

where

$$\eta_k = [a_l(a_l^2 + \lambda_k^2)^{-1/2} + \delta_k](1 - \delta_k)^{-1} \to \eta_\infty = a_l(a_l^2 + \lambda_\infty^2)^{-1/2} < 1.$$

Proof. By using Proposition 3.2, we get the conclusions from [42, Theorem 2]. □

The second version of the inexact primal-dual PPA based on part (ii) takes the following form:

The Primal-Dual PPA-II. Given a tolerance $\varepsilon > 0$. Input $X^0 \in \Re^{n_1 \times n_2}$, $y^0 \in \Re^m$, and $\lambda_0 > 0$. Set $k := 0$. Iterate:

Step 1. Approximately find the unique minimizer

$$X^{k+1} \approx \arg\min_{X \in \Re^{n_1 \times n_2}} \Big\{ \psi_k(X) := \|X\|_* + \Psi_{\lambda_k}(X; y^k) + \frac{1}{2\lambda_k}\|X - X^k\|^2 \Big\}. \tag{67}$$

Step 2. Compute

$$y^{k+1} \;=\; \Pi_{\mathcal{Q}^*}[y^k + \lambda_k(b - \mathcal{A}(X^{k+1}))].$$

Step 3. If $\|(y^k - y^{k+1})/\lambda_k\| \le \varepsilon$; stop; else; update λ_k such that (9); end.

From the computational point of view, in the primal-dual PPA-II, one only needs to approximately solve (67). Two implementable stopping criteria to terminate them are suggested here:

$$\psi_k(X^{k+1}) - \check{\psi}_k \;\le\; \frac{\varepsilon_k^2}{4\lambda_k}, \quad \varepsilon_k > 0, \; \sum_{k=0}^{\infty} \varepsilon_k < \infty, \tag{68a}$$

$$\psi_k(X^{k+1}) - \check{\psi}_k \;\le\; \frac{\delta_k^2}{4\lambda_k}\|(X^{k+1}, y^{k+1}) - (X^k, y^k)\|, \quad \delta_k > 0, \; \sum_{k=0}^{\infty} \delta_k < \infty, \tag{68b}$$

where $\check{\psi}_k := \Theta_{\lambda_k}(y^{k+1}; X^k) - (1/2\lambda_k)\|y^{k+1} - y^k\|^2$. Note that

$$\check{\psi}_k \le \max_{y \in \mathcal{Q}^*} \Big\{ \Theta_{\lambda_k}(y; X^k) - \frac{1}{2\lambda_k}\|y - y^k\|^2 \Big\} = \inf \psi_k. \tag{69}$$

Remark 3.6. *Note that the primal-dual PPA-II is actually the proximal method of multipliers developed in [43, Section 5] applied to problem (2).*

Before applying the general convergence results of the proximal point method [42] to the primal-dual PPA-II, we need the following property.

Proposition 3.3. *Let p_{λ_k} be given as in (63) and $y^{k+1} = \Pi_{\mathcal{Q}^*}[y^k + \lambda_k(b - \mathcal{A}(X^{k+1}))]$. Then, one has*

$$\|(X^{k+1}, y^{k+1}) - p_{\lambda_k}(X^k, y^k)\|^2/(4\lambda_k) \le \psi_k(X^{k+1}) - \inf \psi_k. \tag{70}$$

Proof. Let us denote $p_{\lambda_k}(X^k, y^k)$ by $(\overline{X}^{k+1}, \overline{y}^{k+1})$. Then, the same argument as in [43, Proposition 6] or in Proposition 3.1 implies that

$$\|y^{k+1} - \overline{y}^{k+1}\|^2/(2\lambda_k) \le \psi_k(X^{k+1}) - \inf \psi_k. \tag{71}$$

Since ψ_k is strongly convex in X with modulus $1/\lambda_k$, we know (see, e.g., [42, Proposition 6]) that

$$\|X^{k+1} - \overline{X}^{k+1}\|^2/(2\lambda_k) \leq \psi_k(X^{k+1}) - \inf \psi_k,$$

which, together with (71), yields (70). This completes the proof. □

Theorem 3.7. (**Global Convergence**) *Assume that $\mathcal{F}_P \neq \emptyset$ and condition (51) holds for (2). Let the primal-dual PPA-II be executed with stopping criterion (68a). Then, the generated sequence $\{(X^k, y^k)\} \subset \Re^{n_1 \times n_2} \times \Re^m$ is bounded, and $(X^k, y^k) \to (\overline{X}, \overline{y})$, where $\overline{X}$ is an optimal solution to problem (2) and $\overline{y}$ is an optimal solution to problem (11).*

Proof. Combining Proposition 3.3 with [42, Theorem 1], we know that (X^k, y^k) converges to some $(\overline{X}, \overline{y})$ such that $(0, 0) \in \mathcal{T}_l(\overline{X}, \overline{y})$, which means that $(\overline{X}, \overline{y})$ is a saddle point of the Lagrangian function l and hence $\overline{X}$ is an optimal solution to problem (2) and $\overline{y}$ is an optimal solution to problem (11). This completes the proof. □

Theorem 3.8. (**Local Convergence**) *Assume that $\mathcal{F}_P \neq \emptyset$ and condition (51) holds for problem (2). Let the primal-dual PPA-II be executed with stopping criterion (68a) and (68b). If $\mathcal{T}_l^{-1}$ is Lipschitz continuous at the origin with modulus $a_l > 0$, then $\{(X^k, y^k)\}$ is bounded and $(X^k, y^k) \to (\overline{X}, \overline{y})$, where $\overline{X}$ is the unique optimal solution to problem (2) and $\overline{y}$ is the unique optimal solution to problem (11). Furthermore, one has*

$$\|(X^{k+1} - y^{k+1}) - (\overline{X}, \overline{y})\| \leq \eta_k \|(X^k, y^k) - (\overline{X}, \overline{y})\|, \text{ for all } k \text{ sufficiently large,}$$

where

$$\eta_k = [a_l(a_l^2 + \lambda_k^2)^{-1/2} + \delta_k](1 - \delta_k)^{-1} \to \eta_\infty = a_l(a_l^2 + \lambda_\infty^2)^{-1/2} < 1.$$

Proof. By virtue of (68a) and (68b), from Proposition 3.3, we can easily obtain the conclusions by applying the convergence rate of the general proximal point method [42, Theorem 2] to the case of $\mathcal{T}_l$. □

4 Implementation issues

For the PPAs to be practical, we need to be able to solve the inner sub-problems and evaluate the proximal mapping $\mathcal{P}_\lambda(\cdot)$ efficiently. Here we describe our implementations to achieve these goals.

4.1 First-order methods for the inner sub-problems

In this subsection, we describe the application of first-order methods to solve the inner sub-problems in the PPAs. In particular, we propose to solve the inner sub-problems of the primal PPA and the primal-dual PPA-I by the gradient projection method, and the inner sub-problems of the dual PPA and the primal-dual PPA-II by an accelerated proximal gradient method.

First, we consider the gradient projection method to solve the inner sub-problems of the primal PPA and the primal-dual PPA-I. For some fixed $X \in \Re^{n_1 \times n_2}$, $z \in \Re^m$, and $\lambda > 0$, the inner sub-problems in these PPAs have the following form:

$$\min \Big\{ h(y) \ : \ y \in \mathcal{Q}^* \Big\}, \tag{72}$$

where h is continuously differentiable and its gradient is Lipshitz continuous with modulus $L > 0$. Actually, from (43) and (64), we know that for the primal PPA, $h(y) = -\Theta_\lambda(y; X)$, $\nabla h(y) = \mathcal{A}\mathcal{P}_\lambda[X + \lambda\mathcal{A}^*(y)] - b$ and $L = \lambda\|\mathcal{A}\|_2^2$; and for the primal-dual PPA-I, $h(y) = -\Theta_\lambda(y; X) + \|y - z\|^2/(2\lambda)$, $\nabla h(y) = \mathcal{A}\mathcal{P}_\lambda[X + \lambda\mathcal{A}^*(y)] - b + (y - z)/\lambda$ and $L = \lambda\|\mathcal{A}\|_2^2 + \frac{1}{\lambda}$, where $\|\mathcal{A}\|_2$ denotes the operator norm of $\mathcal{A}$.

One of the simplest methods for solving (72) is the following gradient projection (GP) method:

$$y^{j+1} = \Pi_{\mathcal{Q}^*}[y^j - \alpha_j \nabla h(y^j)], \tag{73}$$

where $y^0 \in \mathcal{Q}^*$ is given, and $\alpha_j > 0$ is the steplength which can be determined by various rules, e.g., the Armijo line search rule. In particular, if $y^j - \alpha_j \nabla h(y^j)$ is feasible, the GP iteration reduces to the standard steepest descent iteration. Let $s > 0$, $\rho \in (0, 1)$, and $\gamma \in (0, 1)$ be given. The Armijo line search rule is to choose $\alpha_j = s\rho^{i_j}$, where i_j is the smallest nonnegative integer i such that

$$h(\Pi_{\mathcal{Q}^*}[y^j - s\rho^i \nabla h(y^j)]) - h(y^j) \ \le \ \gamma \langle \nabla h(y^j), \Pi_{\mathcal{Q}^*}[y^j - s\rho^i \nabla h(y^j)] - y^j \rangle. \tag{74}$$

Alternatively, since ∇h is Lipschitz continuous with modulus L, one can choose the constant steplength rule

$$\alpha_j = s \quad \text{with } s \in (0, 2/L), \tag{75}$$

which was first proposed by Goldstein [22] and Levitin and Poljak [26]. The constant steplength choice is, however, too conservative and the convergence is typically slow. In our implementation, we use the Armijo line search rule, which is shown to be better than the constant steplength rule (see Subsection 5.1). The global convergence of the GP method with the Armijo line search rule (74) was originally shown by Bertsekas [6] for (72) in which h is continuously differentiable and $\mathcal{Q}^*$ is replaced by bound constraints. In 1984, Gafni and Bertsekas [20] proved the global convergence of the GP method with the Armijo line search rule (74) for a general closed convex set. The following theorem gives the results on the complexity iteration of the GP method with the constant steplength rule. For the details, see, e.g., [36, Theorem 2.2.14].

Theorem 4.1. *Let $\{y^j\}$ be generated by the GP method with the steplength α_j chosen by the constant steplength rule (75). Then, for every $j \ge 1$, one has*

$$h(y^j) - \inf h \le O(L/j),$$

and hence $O(L/\varepsilon_{sub})$ iterations suffice to achieve within $\varepsilon_{sub} > 0$ of the optimal value.

Note that theoretically, the accuracy tolerance ε_{sub} needed for solving the inner subproblem (72) should depend on the stopping conditions (46a), (46b) and (45). In our practical implementation of the primal PPA, for the subproblem at the kth iteration, we find that choosing ε_{sub} to be $10^{-2}\|(X^{k+1} - X^k)/\lambda_k\|$ is usually good enough for the overall algorithm to attain the required accuracy.

For a comprehensive study on the GP methods in general, we refer to Bertsekas [7, Chapter 2] and references therein.

Remark 4.1. *In our implementation of the primal PPA and the primal-dual PPA-I where the inner sub-problems are solved by the GP method with Armijo line search, the initial steplength estimate s in (74) at the j iteration is chosen as follows:*

$$s = \begin{cases} 1.11\,\alpha_{j-1} & \text{if } i_{j-1} = 0, \\ \alpha_{j-1} & \text{otherwise.} \end{cases}$$

In our numerical experiments of the primal PPA and the primal-dual PPA-I on NNM problems arising from random matrix completion problems, such an initial estimate is typically accepted as the steplength α_j.

Next, we turn to consider an accelerated proximal gradient method to solve the inner sub-problems of the dual PPA and the primal-dual PPA-II. For some fixed $Z \in \Re^{n_1 \times n_2}$, $y \in \Re^m$, and $\lambda > 0$, in the dual PPA and the primal-dual PPA-II, the inner sub-problems have the following form:

$$\min \Big\{ H(X) := \|X\|_* + h(X) \ : \ X \in \Re^{n_1 \times n_2} \Big\}. \tag{76}$$

It is readily seen from (55) and (64) that h is proper, convex, continuously differentiable on $\Re^{n_1 \times n_2}$, and ∇h is globally Lipschitz continuous with (different) modulus $L > 0$. In fact, for the dual PPA, $h(X) = \Psi_\lambda(X; y)$, $\nabla h(X) = -\mathcal{A}^* \Pi_{\mathcal{Q}^*}[y + \lambda(b - \mathcal{A}(X))]$ and $L = \lambda \|\mathcal{A}\|_2^2$; and for the primal-dual PPA-II, $h(X) = \Psi_\lambda(X; y) + \|X - Z\|^2/(2\lambda)$, $\nabla h(X) = -\mathcal{A}^* \Pi_{\mathcal{Q}^*}[y + \lambda(b - \mathcal{A}(X))] + (X - Z)/\lambda$ and $L = \lambda \|\mathcal{A}\|_2^2 + \frac{1}{\lambda}$.

Recently, Toh and Yun [45] proposed an accelerated proximal gradient (APG) algorithm for solving a more general form of (76) and reported good performances of the APG algorithm on large scale matrix completion problems. (The APG algorithm is in the class of accelerated first-order methods studied by Nesterov, Nemirovski, and others; see [34, 35, 36, 37, 38, 46] and references therein.) A few recent papers have also reported promising numerical results using improved variants of the APG method for some large scale convex optimization problems, see, e.g., [5, 30, 45] and related works. This motivates us to consider APG methods for solving (76).

For given $\tau_0 = \tau_{-1} = 1$ and $X^0 = X^{-1} \in \Re^{n_1 \times n_2}$, the APG algorithm applied to

solving (76) can be expressed as:

$$\begin{cases} Y^j &= X^j + \tau_j^{-1}(\tau_{j-1} - 1)(X^j - X^{j-1}), \\ X^{j+1} &= \mathcal{P}_{L^{-1}}[Y^j - L^{-1}\nabla h(Y^j)], \\ \tau_{j+1} &= (\sqrt{1 + 4\tau_j^2} + 1)/2, \end{cases} \tag{77}$$

where L is the Lipschitz modulus of ∇h.

The following theorem shows that the APG algorithm given in (77) has an attractive iteration complexity of $O(\sqrt{L/\varepsilon_{sub}})$ for achieving ε_{sub}-optimality for any $\varepsilon_{sub} > 0$. For the more general discussions, see, e.g., [46, Corollary 2].

Theorem 4.2. *Let $\{Y^j\}, \{X^j\}, \{\tau_j\}$ be generated by the APG algorithm (77). Then, for any $X \in \Re^{n_1 \times n_2}$ such that $H(X) \le \inf_{X \in \Re^{n_1 \times n_2}}\{H(X)\} + \varepsilon_{sub}$, we have*

$$\min_{i=0,1,\ldots,j+1}\{H(X^i)\} \le H(X) + \varepsilon_{sub} \quad \textit{whenever} \quad j \ge \sqrt{\frac{4L\|X - X^0\|^2}{\varepsilon_{sub}}} - 2.$$

Remark 4.2. *Notice that L^{-1} in the second step of (77) plays the role of the steplength, and the default steplength of L^{-1} could be too conservative. The APG method in (77) can generally be accelerated by using a smaller L. As explained in [46], one chooses an initial under estimate of L and increasing the estimate by a pre-specified constant factor and repeating the iteration whenever the following condition is violated:*

$$h(X^{j+1}) \le h(Y^j) + \langle \nabla h(Y^j), X^{j+1} - Y^j \rangle + \frac{L}{2}\|X^{j+1} - Y^j\|^2. \tag{78}$$

We use the linesearch-like scheme stated above in our implementation, which has been shown in [45] to greatly accelerate the convergence of the APG algorithm.

Again, the accuracy tolerance ε_{sub} in solving the subproblem (76) should theoretically be dependent on the stopping conditions (58a), (58b) and (57). In our practical implementation of the dual PPA, we choose $\varepsilon_{sub} = 2 \times 10^{-2}\|(y^k - y^{k+1})/\lambda_k\|$ when solving the subproblem at the k iteration.

Remark 4.3. *We should mention that the APG method (see, e.g., [46, Algorithm 2]) applied to the inner sub-problems of the primal PPA and the primal-dual PPA-I requires two SVDs per iteration if the linesearch-like strategy in (78) is employed. Since the iteration complexities of the GP method and the APG method are $O(L/\varepsilon_{sub})$ and $O(\sqrt{L/\varepsilon_{sub}})$, respectively, it seems that one may still benefit more from the APG method than from the GP method. However, our numerical results show that the number of iterations taken by the GP method is at most twice that of iterations taken by the APG method[3]. Therefore, the total computational cost consumed by the APG method is more than that consumed by the GP method since the latter (with the line search) only requires slightly more than one SVD per iteration on the average.*

[3]Such an observation is possible because the $O(L/\varepsilon_{sub})$ and $O(\sqrt{L/\varepsilon_{sub}})$ iteration complexities are for the worst cases.

4.2 Evaluation of singular value decompositions

The main computational cost at each iteration of the GP method and the APG algorithm is to compute a partial SVD (see (19) and (20)) so as to compute $\mathcal{P}_\lambda[X + \lambda\mathcal{A}^*(y^j)]$ or $\mathcal{P}_{L^{-1}}[Y^j - L^{-1}\nabla h(Y^j)]$. In particular, in the j-th iteration of the GP method, for given X and λ, we need to know those singular values of $X + \lambda\mathcal{A}^*(y^j)$ exceeding λ and their corresponding singular vectors; and in the j-th iteration of the APG algorithm, for given y and λ, we need to know those singular values of $Y^j - L^{-1}\nabla h(Y^j)$ exceeding L^{-1} and their corresponding singular vectors.

As in [10, 45], we use the PROPACK package (see [24]) based on the Lanczos bidiagonalization algorithm with partial reorthogonalization to compute a partial SVD. Note that PROPACK cannot automatically compute only those singular values of a matrix greater than a given constant but it can compute a specified number sv of the largest or smallest singular values and their corresponding singular vectors. Hence, we must specify the number sv^k of the largest singular values to compute beforehand at the k-th iteration. We use the following procedure given in [45] to update sv^k. Input $\mathrm{sv}^0 = 5$, for $k = 0, 1, \ldots$, update sv^{k+1} by

$$\mathrm{sv}^{k+1} = \begin{cases} \mathrm{svp}^k + 1 & \text{if } \mathrm{svp}^k < \mathrm{sv}^k, \\ \mathrm{svp}^k + 5 & \text{if } \mathrm{svp}^k = \mathrm{sv}^k, \end{cases}$$

where svp^k is the rank of X^k. In our experiments, the above procedure appears to work well.

In addition, we use the truncation technique introduced in [45] in the implementation of the GP method and the APG algorithm. For the details on the description of the truncation technique, see [45, Section 3.4]. The benefit of using the truncation technique is that the rank of the iterate X^k is kept as low as possible without severely affecting the convergence of the algorithms. The main motivation for keeping the rank of X^k low is to reduce the cost of computing the partial SVD of $X^k + \lambda\mathcal{A}^*(y^k)$ or $Y^k - L^{-1}\nabla h(Y^k)$, where Y^k is linear combination of X^k and X^{k-1}.

5 Numerical experiments

In this section, we report some numerical results on the application of the PPAs to NNM problems arising in minimum rank matrix completion problems.

In the matrix completion problem, the goal is to recover an unknown matrix from a sampling of its entries by solving the following problem:

$$\min \Big\{ \mathrm{rank}(X) \,:\, X_{ij} = M_{ij},\ (i,j) \in \Omega \Big\}, \tag{79}$$

where X is the decision variable, M is the unknown matrix with m available sampled entries, and Ω is the set of indices of the observed entries. This is a special case of the

rank minimization problem (3) for which one has

$$\mathcal{A}(X) = X_\Omega, \tag{80}$$

where $X_\Omega \in \Re^{|\Omega|}$ is the vector consisting of elements selected from X whose indices are in Ω.

We have implemented the primal PPA, the dual PPA and the primal-dual PPA-I (-II) in MATLAB, using PROPACK package to evaluate partial SVDs. All runs are performed on an Intel Xeon 3.20GHz PC with 4GB memory, running Linux and MATLAB (Version 7.6). In our experiments, the initial point for the primal PPA, the dual PPA and the primal-dual PPA-I (-II) is set to be $X^0 = 0, y^0 = 0, (X^0, y^0) = (0, 0)$, respectively.

We first consider random matrix completion problems, which are generated as in [14]. For each triple (n, r, m), where n (we set $n_1 = n_2 = n$) is the dimension of matrix, r is the predetermined rank, and m is the number of sampled entries, we first generate $M = M_L M_R^T$, where M_L and M_R are $n \times r$ matrices with i.i.d. standard Gaussian entries. Then we select a subset Ω uniformly at random among all sets of cardinality m. Note that from (80), we have $b = \mathcal{A}(M)$.

We also consider random matrix completion problems with noisy sampled entries. For the random matrix completion problem with noisy data, the matrix M is contaminated with a noisy matrix Ξ, and

$$b = \mathcal{A}(M + \omega\Xi),$$

where Ξ is a matrix with i.i.d. standard Gaussian random entries and ω is set to be

$$\omega = \kappa \frac{\|\mathcal{A}(M)\|}{\|\mathcal{A}(\Xi)\|},$$

and κ is a given noise factor.

In our experiments, the primal PPA or the primal-dual PPA-I is stopped when any of the following conditions is satisfied:

$$(i) \quad \frac{\|b - \mathcal{A}(X^k)\|}{\max\{1, \|b\|\}} < \text{Tol},$$

$$(ii) \quad \left| \frac{\|b - \mathcal{A}(X^k)\|}{\|b - \mathcal{A}(X^{k-1})\|} - 1 \right| < 10^{-2} \text{ and } \frac{\|X^k - X^{k-1}\|}{\max\{1, \|X^k\|\}} < 10^{-1}.$$

where Tol is a given tolerance. The dual PPA or the primal-dual PPA-II is stopped when the following condition is satisfied:

$$\frac{1}{\lambda_k}\|y^k - y^{k-1}\| < \text{Tol}.$$

Unless otherwise specified, in our experiments, Tol is set to be 10^{-4}. In addition, the accuracy of the recovery solution X^{sol} of the PPAs is measured by the relative error

defined by:

$$\text{error} := \frac{\|X^{\text{sol}} - M\|}{\|M\|}, \tag{81}$$

where M is the pre-generated low-rank matrix.

5.1 Sensitivity of the primal PPA to the parameter λ

Here we investigate the benefits of the primal PPA for solving (2) as opposed to the SVT algorithm in [10] applied to the regularized problem (52). For simplicity, we only consider the case without second order cone constraints, i.e., $\mathcal{Q} = \{0\}^{m_1}$.

In this experiment, we use the stopping criterion in the SVT algorithm (downloaded from [12] in April 2009) for the primal PPA, i.e.,

$$\frac{\|b - \mathcal{A}(X^k)\|}{\max\{1, \|b\|\}} < 10^{-4}.$$

Table 1 reports the number of iterations for one random instance without noise and gives the ratio (m/d_r) between the number of sampled entries (m) and the degrees of freedom $(d_r := r(2n-r))$ of an $n \times n$ rank-r matrix. Table 1 also presents the results on the performance of the SVT algorithm with different constant steplengths $\delta = 1.0/p, 1.2/p, 1.5/p$, where $p := m/n^2$ is the proportion of observed entries. From Table 1, we can see that when λ is set to $n/2$ or n, the primal PPA recovers the matrix M, whereas the SVT algorithm fails to recover it. In addition, the number of iterations of the SVT algorithm varies greatly with the constant steplength δ. This behavior is consistent with the fact that the SVT algorithm used a heuristic choice of constant steplength which may be overly optimistic and it has no known convergence guarantee, whereas the primal PPA incorporated the Armijo line search rule to guarantee convergence by ensuring sufficient descent in the objective function at each iteration.

Table 1: Numerical results for the primal PPA versus the SVT algorithm.

$\mathbf{n/r/(}m/d_r\mathbf{)}$	method	$\lambda = n/2$	$\lambda = n$	$\lambda = 5n$	$\lambda = 10n$
1000/50/4	PPA	64	60	88	169
	SVT ($\delta = 1.0/p$)	fail	fail	135	250
	SVT ($\delta = 1.2/p$)	fail	fail	112	208
	SVT ($\delta = 1.5/p$)	fail	fail	89	165
5000/50/5	PPA	70	72	86	141
	SVT ($\delta = 1.0/p$)	fail	fail	129	239
	SVT ($\delta = 1.2/p$)	fail	fail	108	199
	SVT ($\delta = 1.5/p$)	fail	fail	86	159

5.2 Performance of PPAs on random matrix completion problems

In this section, we report the performance of the PPAs for solving randomly generated matrix completion problems without and with noise.

Notice that for the primal PPA, we fixed the parameter λ to be $\lambda = \max\{10^3, n\}$. The performance of the primal PPA on random matrix completion problems without noise is displayed in Table 2. In this table, we give the ratio (m/d_r), the mean value of the parameter (λ), the average number of iterations (iter), the average number of positive singular values of the recovered matrix (#sv), the average CPU time (in seconds), and the average error (as defined in (81)) of the recovered matrix, of five runs. As can be seen from Table 2, the maximum average number of iterations is 99 for the case of $n = 10^5$ and $r = 10$, and for all the other cases, the average number of iterations are at most 83. Notice that all the errors are smaller than or roughly equal to 10^{-4}, except for the case $n = 50000$ and $r = 10$. In addition, the primal PPA can recover a $100,000 \times 100,000$ matrix of rank 10 from about 0.12% of the sampled entries in less than 1000 seconds with an error of 8.58×10^{-5}.

Table 2: Numerical results for the **primal PPA** on random matrix completion problems without noise.

Unknown M					**Results**			
n	m	r	m/d_r	λ^{-1}	iter	#sv	time	error
1000	119560	10	6	1.00e-03	54	10	5.77e+00	7.02e-05
	389638	50	4	1.00e-03	61	50	3.19e+01	7.42e-05
	569896	100	3	1.00e-03	77	100	1.03e+02	5.34e-05
5000	599936	10	6	2.00e-04	57	10	2.23e+01	6.11e-05
	2487739	50	5	2.00e-04	72	50	2.13e+02	4.12e-05
	3960882	100	4	2.00e-04	83	100	6.94e+02	1.04e-04
10000	1200730	10	6	1.00e-04	52	10	4.40e+01	1.43e-04
	4985869	50	5	1.00e-04	81	50	5.61e+02	3.05e-05
	7959722	100	4	1.00e-04	82	100	1.48e+03	8.35e-05
20000	2400447	10	6	5.00e-05	66	10	1.07e+02	1.20e-04
30000	3599590	10	6	3.33e-05	72	10	1.86e+02	5.90e-05
50000	5995467	10	6	2.00e-05	70	10	3.49e+02	5.59e-04
100000	11994813	10	6	1.00e-05	99	10	9.85e+02	8.58e-05

The performance of the primal PPA on random matrix completion problems with

noise is displayed in Table 3. We report the same results as in Table 2. As can be seen from Table 3, the primal PPA takes at most 67 iterations on the average to recover the unknown matrices. More importantly, the relative errors are all smaller than 7.94×10^{-2}, which is smaller than the given noise level of $\kappa = 0.1$.

Table 3: Numerical results for the **primal PPA** on random matrix completion problems with noise. The noise factor κ is set to 0.1.

Unknown M					**Results**			
n /κ	m	r	m/d_r	λ^{-1}	iter	#sv	time	error
1000 /0.10	119560	10	6	1.00e-03	38	10	5.10e+00	5.62e-02
	389638	50	4	1.00e-03	47	51	3.16e+01	7.74e-02
	569896	100	3	1.00e-03	45	100	5.80e+01	7.94e-02
5000 /0.10	599936	10	6	2.00e-04	45	10	2.38e+01	5.02e-02
	2487739	50	5	2.00e-04	53	50	2.23e+02	5.93e-02
	3960882	100	4	2.00e-04	47	100	4.93e+02	7.72e-02
10000 /0.10	1200730	10	6	1.00e-04	45	10	5.59e+01	4.89e-02
	4985869	50	5	1.00e-04	36	50	3.62e+02	5.84e-02
	7959722	100	4	1.00e-04	57	100	1.24e+03	6.82e-02
20000 /0.10	2400447	10	6	5.00e-05	47	10	9.32e+01	5.60e-02
30000 /0.10	3599590	10	6	3.33e-05	53	10	1.69e+02	4.80e-02
50000 /0.10	5995467	10	6	2.00e-05	58	10	3.33e+02	5.24e-02
100000 /0.10	11994813	10	6	1.00e-05	67	10	7.53e+02	5.42e-02

In Table 4 and Table 5, we report the performance of the dual PPA on random matrix completion problems without and with noise, respectively. Note that for the dual PPA, we fixed λ to be $\lambda = 10^4/\|\mathcal{A}^*(b)\|_2$. As shown in the tables, we can see that the dual PPA works well with relatively small values of λ.

Comparing the performance of the primal PPA and the dual PPA on random matrix completion problems without/with noise, we observe that the dual PPA outperforms the primal PPA[4]. For the case $n = 10^5$ and $r = 10$ without noise, the dual PPA solves the problem in 519 seconds whereas the primal PPA takes 985 seconds. There are two possible reasons to explain this difference. First, the inner sub-problems of the dual PPA are solved by an APG method, while those of the primal PPA are solved by a gradient projection method. Second, the former works well with relatively small values of λ, while the latter requires larger values of λ. However, a larger value of λ often leads to a slower rate of convergence for the outer iteration in the PPA.

[4]Here our conclusion is based on using the gradient-type methods to solve the corresponding sub-problems.

Table 4: Numerical results for the **dual PPA** on random matrix completion problems without noise.

Unknown M					**Results**			
n	m	r	m/d_r	λ^{-1}	iter	#sv	time	error
1000	119560	10	6	1.44e-02	35	10	3.90e+00	1.05e-04
	389638	50	4	5.37e-02	51	50	2.95e+01	6.21e-05
	569896	100	3	8.66e-02	56	100	7.78e+01	2.41e-05
5000	599936	10	6	1.38e-02	42	10	1.71e+01	7.34e-05
	2487739	50	5	6.08e-02	50	50	1.47e+02	6.50e-05
	3960882	100	4	1.02e-01	56	100	4.32e+02	9.68e-05
10000	1200730	10	6	1.37e-02	40	10	2.96e+01	1.40e-04
	4985869	50	5	5.93e-02	51	50	3.19e+02	6.54e-05
	7959722	100	4	9.88e-02	56	100	9.05e+02	1.04e-04
20000	2400447	10	6	1.35e-02	45	10	6.72e+01	1.50e-04
30000	3599590	10	6	1.35e-02	54	10	1.21e+02	1.41e-04
50000	5995467	10	6	1.34e-02	58	10	2.46e+02	4.83e-05
100000	11994813	10	6	1.34e-02	55	10	5.19e+02	1.04e-04

Table 5: Numerical results for the **dual PPA** on random matrix completion problems with noise. The noise factor κ is set to 0.1.

Unknown M					**Results**			
n /κ	m	r	m/d_r	λ^{-1}	iter	#sv	time	error
1000 /0.10	119560	10	6	1.44e-02	29	10	3.95e+00	4.49e-02
	389638	50	4	5.37e-02	31	50	1.52e+01	5.49e-02
	569896	100	3	8.67e-02	39	100	4.36e+01	6.39e-02
5000 /0.10	599936	10	6	1.38e-02	39	10	2.20e+01	4.51e-02
	2487739	50	5	6.08e-02	39	50	1.09e+02	4.96e-02
	3960882	100	4	1.02e-01	41	100	2.71e+02	5.67e-02
10000 /0.10	1200730	10	6	1.37e-02	44	10	4.73e+01	4.53e-02
	4985869	50	5	5.93e-02	39	50	2.26e+02	4.99e-02
	7959722	100	4	9.89e-02	47	100	6.92e+02	5.73e-02
20000 /0.10	2400447	10	6	1.35e-02	44	10	9.65e+01	4.52e-02
30000 /0.10	3599590	10	6	1.35e-02	45	10	1.45e+02	4.53e-02
50000 /0.10	5995467	10	6	1.34e-02	47	10	2.70e+02	4.53e-02
100000 /0.10	11994813	10	6	1.34e-02	43	10	5.42e+02	4.53e-02

Remark 5.1. *Here we do not report the numerical results for the primal-dual PPA-I and PPA-II for the sake of saving some space. Indeed, in our experiments, we observe that*

the performance of the primal-dual PPA-I is similar to that of the primal PPA, and the performance of the primal-dual PPA-II is similar to that of the dual PPA.

5.3 Performance of the dual PPA on real matrix completion problems

Now we consider the well-known matrix completion problem in the Netflix Prize Contest [39]. Three data sets are provided in the Contest.

1. `training set`: consists of about 100 million ratings from 480189 randomly chosen users on 17770 movie titles. The ratings are integers on a scale from 1 to 5.

2. `qualifying set`: contains over 2.8 million user/movie pairs but with the ratings withheld. The `qualifying set` is further randomly divided into two disjoint subsets called `quiz` and `test` subsets.

3. `probe set`: this is a subset of the `training set` consisting of about 1.4 million user/movie pairs with known ratings. This subset is constructed to have similar properties as the `qualifying set`.

For convenience, we assume that the users are enumerated from 1 to 480189, and the movies are enumerated from 1 to 17770. We define

$$\begin{aligned}
\Omega_t &= \big\{(i,j) \,:\, \text{user } i \text{ has rated movie } j \text{ in the } \texttt{training set}\big\},\\
\Omega_q &= \big\{(i,j) \,:\, \text{user } i \text{ has rated movie } j \text{ in the } \texttt{qualifying set}\big\},\\
\Omega_p &= \big\{(i,j) \,:\, \text{user } i \text{ has rated movie } j \text{ in the } \texttt{probe set}\big\}.
\end{aligned}$$

The Netflix Prize Contest solicits algorithms that can make predictions for all the withheld ratings for the user/movie pairs in the `qualifying set`. The quality of the predictions is measured by the root mean squared error:

$$\text{RMSE} = \left[\frac{1}{|\Omega_q|}\sum_{(k,j)\in\Omega_q}(x_{kj}^{\text{pred}} - x_{kj}^{\text{true}})^2\right]^{1/2},$$

where x_{kj}^{pred}, x_{kj}^{true} are the predicted and actual ratings for the k-th user on the j-th movie. For any predictions submitted to the Contest, the RMSE for the `quiz subset` will be reported publicly on [39] whereas the RMSE for the `test subset` is withheld but will be employed for the purpose of selecting the winner in the Contest. At the start of the Contest, the RMSE of Netflix's proprietary Cinematch algorithm on the `quiz` and `test` subsets, based on the training data set alone, were 0.9514 and 0.9525, respectively. The RMSE obtained by the Cinematch algorithm on the `probe set` is 0.9474.

Due to memory constraint, in our numerical experiment, we divide the `training set` and `probe set` respectively into 5 disjoint subsets according to the users' id as follows:

$$\begin{aligned} \texttt{training-k} &= \big\{(i,j) \in \Omega_t \backslash \Omega_p \ : \ (k-1)100,000 < i \le (k+1)100,000\big\}, \\ \texttt{probe-k} &= \big\{(i,j) \in \Omega_p \ : \ (k-1)100,000 < i \le (k+1)100,000\big\}, \quad k = 1, \ldots, 5. \end{aligned}$$

Note that we removed the data in the `probe set` from the `training set` in the experiments.

We apply the dual PPA to (4) for all the 5 subsets to predict the ratings of all the users on all the movies. As the noise level δ for these problems are not known, we estimate δ dynamically from (outer) iteration to iteration. That is, for the k-th outer iteration in the dual PPA, we set $\delta = 0.5\|b - \mathcal{A}(X^k)\|$. In addition, as the optimal solutions of these problems are not necessarily low-rank, we truncate the rank of $X^{j+1} = \mathcal{P}_{L^{-1}}[Y^j - L^{-1}\nabla h(Y^j)]$ in the APG algorithm (77) to 10 in each iteration of the APG algorithm. We have tested truncating the rank to 50, but the results were slightly worse.

For each of the subsets `training-k`, we compute the RMSE for the corresponding probe subsets `probe-k`. Table 6 shows the results we obtained. We should note that in our experiment, we do not preprocess the data sets via any statistical means, except to center the partially observed matrix M^k corresponding to `training-k` such that the modified matrix $\bar{M}^k$ has all its rows and columns each having zero sum. That is,

$$\bar{M}^k_{ij} = M^k_{ij} - d_i - f_j, \quad \forall\, i, j$$

and d_i, f_j are determined so that $\sum_j \bar{M}^k_{ij} = 0$ and $\sum_i \bar{M}^k_{ij} = 0$ for all i and j.

Table 6: Numerical results for the **dual PPA** on matrix completion problems arising from Netflix Contest. The number of movies is 17770.

Unknown M				**Results**				
	n	m	λ^{-1}	iter	#sv	time	`training` RMSE	`probe` RMSE
`training-1`	100000	2.08e+07	2.92e+00	35	10	3.9e+02	0.8148	0.9309
`training-2`	100000	2.08e+07	2.93e+00	35	10	4.0e+02	0.8126	0.9292
`training-3`	100000	2.09e+07	2.93e+00	35	10	4.0e+02	0.8131	0.9278
`training-4`	100000	2.07e+07	2.94e+00	35	10	3.9e+02	0.8152	0.9331
`training-5`	80189	1.66e+07	2.64e+00	35	10	2.9e+02	0.8136	0.9366
`training\probe`	160378	9.98e+07				1.9e+03	0.8139	0.9313

As we can observe from Table 6, the `training set` (with `probe set` removed) RMSE is much lower than the `probe set` RMSE, and this reflects that the dual PPA on (4) over trains the data. Despite that, the `probe set` RMSE of 0.9313 we obtained is better than that obtained by Netflix's Cinematch algorithm. The computed RMSE for the `quiz subset` is 0.9416.

6 Conclusions and discussions

In this paper, we have proposed implementable proximal point algorithms in the primal, dual and primal-dual forms for solving the nuclear norm minimization problem with linear equality and second order cone constraints, and presented comprehensive convergence results. These algorithms are efficient and competitive to state-of-the-art alternatives when the inner sub-problems of these algorithms are solved by either the gradient projection method or the accelerated proximal point method.

Before closing this paper, we would like to discuss future research directions related to this work. Firstly, our algorithms achieve linear rate of convergence under the condition that $\mathcal{T}_f^{-1}$ or $\mathcal{T}_g^{-1}$ or $\mathcal{T}_l^{-1}$ is Lipschitz continuous at the origin. It is then interesting to know whether one can characterize these conditions as in [50]. Secondly, it would be worth exploring the performance of these algorithms in which the inner sub-problems are solved by second-order methods such as semismooth Newton and smoothing Newton methods, where applicable. Finally, we plan to study how the general framework presented in this paper can help solve more general nuclear norm optimization problems, see, e.g., [49] for the nuclear norm constrained optimization problem.

Acknowledgements

We are grateful to the two anonymous referees and the associate editor for their helpful comments and suggestions which helped improve the quality of this paper. Thanks also go to Zhaosong Lu at Simon Fraser University for bringing the two papers [30, 49] to our attention.

References

[1] Alfakih, A.Y., Khandani, A. and Wolkowicz, H., *Solving Euclidean distance matrix completion problems via semidefinite programming*, Comp. Optim. Appl. 12 (1999), 13–30.

[2] Alizadeh, F. and Goldfarb, D., *Second-order cone programming*, Math. Program. 95 (2003), 3–51.

[3] Ames, B.P.W. and Vavasis, S.A., *Nuclear norm minimization for the planted clique and biclique problems*, preprint, 2009.

[4] Barvinok, A., *Problems of distance geometry and convex properties of quadratic maps*, Discrete Comput. Geom. 13 (1995), 189–202.

[5] Beck, A. and Teboulle, M., *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sciences 1 (2009), 183–202.

[6] Bertsekas, D.P., *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control 21 (1976), 174–184.

[7] Bertsekas, D.P., Nonlinear Programming, 2nd edition, Athena Scientific, Belmont, 1999.

[8] Burer, S. and Monteiro, R.D.C., *A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization*, Math. Program. 95 (2003), 329–357.

[9] Burer, S. and Monteiro, R.D.C., *Local minima and convergence in low-rank semidefinite programs*, Math. Program. 103 (2005), 427–444.

[10] Cai, J.-F., Candès, E.J. and Shen, Z.W., *A singular value thresholding algorithm for matrix completion*, to appear in SIAM Journal on Optimization.

[11] Candès, E.J., *Compressive sampling*, In International Congress of Mathematicians. Vol. III, Eur. Math. Soc., Zöurich, 1433–1452, 2006.

[12] Candès, E.J. and Becker, S., *Singular value thresholding – codes for the SVT algorithm to minimize the nuclear norm of a matrix, subject to linear constraints*, April 2009. Available at `http://svt.caltech.edu/code.html`.

[13] Candès, E.J. and Tao, T., *Nearly optimal signal recovery from random projections: Universal encoding strategies*, IEEE Trans. Info. Theory 52 (2006), 5406–5425.

[14] Candès, E.J. and Recht, B., *Exact matrix completion via convex optimization*, Found. Comput. Math. 9 (2009), 712–772.

[15] Donoho, D., *Compressed sensing*, IEEE Trans. Info. Theory 52 (2006), 1289–1306.

[16] Faraut, U. and Korányi, A., Analysis on Symmetric Cones, Oxford Mathematical Monographs, Oxford University Press, New York, 1994.

[17] Fazel, M., Matrix rank minimization with applications, Ph.D. thesis, Stanford University, 2002.

[18] Fazel, M., Hindi, H. and Boyd, S., *A rank minimization heuristic with application to minimum order system approximation*, In Proceedings of the American Control Conference, 2001.

[19] Fazel, M., Hindi, H. and Boyd, S., *Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices*, In Proceedings of the American Control Conference, 2003.

[20] Gafni, E.H. and Bertsekas, D.P., *Two-metric projection methods for constrained optimization*, SIAM J. Control Optim. 22 (1984), 936–964.

[21] Ghaoui, L.E. and Gahinet, P., *Rank minimiztion under LMI constraints: A framework for output feedback problems*, In Proceedings of the European Control Conference, 1993.

[22] Goldstein, A.A., *Convex programming in Hilbert space*, Bull. Amer. Math. Soc. 70 (1964), 709–710.

[23] Hiriart-Urruty, J.-B. and Lemaréchal, C., *Convex Analysis and Minimization Algorithms*, Vols. 1 and 2, Springer-Verlag, Berlin, Heidelberg, New York, 1993.

[24] Larsen, R.M., *PROPACK–Software for large and sparse SVD calculations*, Available from `http://sun.stanford.edu/~rmunk/PROPACK/`.

[25] Lemaréchal, C. and Sagastizábal, C., *Practical aspects of the Moreau-Yosida regularization I: theoretical preliminaries*, SIAM J. Optim. 7 (1997), 367–385.

[26] Levitin, E.S. and Polyak, B.T., *Constrained minimization problems*, USSR Computational Mathematics and Mathematical Physics 6 (1966), 1–50.

[27] Linial, N., London, E. and Rabinovich, Y., *The geometry of graphs and some of its algorithmic applications*, Combinatorica 15 (1995), 215–245.

[28] Liu, Z. and Vandenberghe, L., *Interior-point method for nuclear norm approximation with application to system identification*, SIAM J. Matrix Anal. Appl. 31 (2009), 1235–1256.

[29] Lu, F., Keles, S., Wright, S.J. and Wahba, G., *A framework for kernel regularization with application to protein clustering*, Proceedings of the National Academy of Sciences, 102 (2005), 12332–12337.

[30] Lu, Z., Monteiro, R.D.C. and Yuan, M., *Convex optimization methods for dimension reduction and coefficient estimation in multivariate linear regression*, preprint, 2009.

[31] Ma, S.Q., Goldfarb, D. and Chen, L.F., *Fixed point and Bregman iterative methods for matrix rank minimization*, to appear in Mathematical Programming.

[32] Martinet, B., *Régularisation d'inédquations variationelles par approximations successives*, Rev. Francaise Inf. Rech. Oper., 4 (1970), 154–159.

[33] Moreau, J.J., *Proximite et dualite dans un espace hilbertien*, Bulletin de la Societe Mathematique de France, 93 (1965), 273–299.

[34] Nemirovski, A., *Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, SIAM J. Optim. 15 (2005), 229–251.

[35] Nesterov, Y.E., *A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$*, Doklady AN SSSR, 269 (1983), 543–547.

[36] Nesterov, Y.E., Introductory Lectures on Convex Optimization: A Basic Course, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2004.

[37] Nesterov, Y.E., *Smooth minimization of nonsmooth functions*, Math. Program. 103 (2005), 127–152.

[38] Nesterov, Y.E., *Gradient methods for minimizing composite objective function*, Report, CORE, Catholic University of Louvain, Louvain-la-Neuve, Belgium, September, 2007.

[39] Netflix Prize: `http://www.netflixprize.com/`.

[40] Recht, B., Fazel, M. and Parrilo, P.A., *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, to appear in SIAM Review.

[41] Rockafellar, R.T., Convex Analysis, Princeton University Press, Princeton, 1970.

[42] Rockafellar, R.T., *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim. 14 (1976), 877–898.

[43] Rockafellar, R.T., *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res. 1 (1976), 97–116.

[44] Sturm, J.F., *Using SeDuMi 1.02, a Matlab toolbox for optimization over symmetric cones*, Optim. Methods Softw. 11-12 (1999), 625–653.

[45] Toh, K.C. and Yun, S.W., *An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems*, to appear in Pacific Journal of Optimization.

[46] Tseng, P., *On accelerated proximal gradient methods for convex-concave optimization*, preprint, University of Washington, 2008.

[47] Tütüncü, R.H., Toh, K.C. and Todd, M.J., *Solving semidefinite-quadratic-linear programs using SDPT3*, Math. Program. 95 (2003), 189–217.

[48] Yosida, K., Functional Analysis, Springer Verlag, Berlin, 1964.

[49] Yuan, M., Ekici, A., Lu, Z. and Monteiro. R.D.C., *Dimension reduction and coefficient estimation in multivariate linear regression*, J. R. Statist. Soc. B 69 (2007), 329–346.

[50] Zhao, X.Y., Sun, D.F. and Toh, K.C., *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim. 20 (2010), 1737–1765.