Computational Optimization and Applications (2021) 79:273–300
https://doi.org/10.1007/s10589-021-00276-5

Low-rank factorization for rank minimization with nonconvex regularizers

April Sagan[1] · **John E. Mitchell**[1]

Received: 14 October 2020 / Accepted: 6 April 2021 / Published online: 24 April 2021

Abstract
Rank minimization is of interest in machine learning applications such as recommender systems and robust principal component analysis. Minimizing the convex relaxation to the rank minimization problem, the nuclear norm, is an effective technique to solve the problem with strong performance guarantees. However, nonconvex relaxations have less estimation bias than the nuclear norm and can more accurately reduce the effect of noise on the measurements. We develop efficient algorithms based on iteratively reweighted nuclear norm schemes, while also utilizing the low rank factorization for semidefinite programs put forth by Burer and Monteiro. We prove convergence and computationally show the advantages over convex relaxations and alternating minimization methods. Additionally, the computational complexity of each iteration of our algorithm is on par with other state of the art algorithms, allowing us to quickly find solutions to the rank minimization problem for large matrices.

Keywords Rank minimization · Matrix completion · Nonconvex regularizers · Semidefinite programming

1 Introduction

We consider the rank minimization problem with linear constraints formulated as

This work was supported in part by National Science Foundation under Grant Number DMS-1736326.

✉ April Sagan
aprilsagan1729@gmail.com

John E. Mitchell
mitchj@rpi.edu

[1] Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, New York 12180, USA

$$\begin{aligned} \min_{X \in \mathcal{S}^n} \quad & \text{rank}(X) + \phi(X) \\ \text{subject to} \quad & \mathcal{A}(X) = b \\ & X \succeq 0 \end{aligned}$$

where $\mathcal{S}^n$ denotes the set of symmetric $n \times n$ matrices, $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$ is an linear map, $b \in \mathbb{R}^m$ is the measurement vector, and $\phi(X)$ is an L- smooth function. A common example is matrix completion, in which the linear constraint is $P_\Omega(M) = P_\Omega(X)$, where Ω is the set of indices (i, j) of known points in the matrix, and $P_\Omega : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ is the projection onto the set of matrices which the entry (i, j) vanishes for all $(i, j) \notin \Omega$. Formally, we define P_Ω as

$$P_\Omega(X)_{ij} = \begin{cases} 0 & (i,j) \notin \Omega \\ X_{ij} & (i,j) \in \Omega \end{cases}$$

Additionally, in the presence of noise, we can penalize the constraint by adding $\phi(X) = \frac{\beta}{2}||P_\Omega(X - M)||_F^2$ to the objective function, with a parameter β. Solving the rank minimization problem directly is impractical due to the rank function being non-convex and highly discontinuous. In practice, it is common to instead minimize the convex relaxation to the rank function known as the nuclear norm, which is defined as the sum of the singular values of the matrix, or in the case of positive semidefinite matrices, the trace.

$$\begin{aligned} \min_{X} \quad & \text{trace}(X) \\ \text{subject to} \quad & \mathcal{A}(X) = b \\ & X \succeq 0 \end{aligned}$$

The nuclear norm, denoted by $||X||_* = \sum_{i=1}^{n} \sigma_i(X)$ where $\sigma_i(X)$ is the i^{th} singular value of X, is the tightest convex relaxation, and in the case of matrix completion on an n by n matrix known to be at most rank r, it has been shown to exactly recover the original matrix with high probability if at least $Cnr\log(n)$ entries are observed, for an absolute constant C, under the assumption that the original matrix satisfies the incoherence property [5]. However, minimizing the nuclear norm is not always the best approach. As observed in the similar problem of l_0 norm minimization, the convex relaxation, the l_1 norm, introduces an estimation bias [33]. Consider the following rank minimization problem:

$$\min_{X \in \mathbb{R}^{m \times n}} ||X||_* + \frac{\beta}{2}||P_\Omega(\tilde{M} - X)||_F^2$$

where $\tilde{M}$ is a low rank matrix, M, plus Gaussian noise. As we show in Sect. 2, the minimizer to the expected value of the nuclear norm regularized formulation is $\frac{p\beta}{p\beta+1}M$, where $p = \frac{|\Omega|}{mn}$. The bias of this formulation comes from the nuclear norm not only minimizing the smallest singular values, which correspond to the noise, but also the largest singular values, which correspond to the signal.

Another common approach to fitting a low rank matrix to a set of measurements is rank constrained optimization, wherein one attempts to find a rank r matrix that minimizes an objective function.

$$\min_{X\in\mathbb{R}^{m\times n}} ||\mathcal{A}(X)-b||^2 \text{ subject to } \operatorname{rank}(X)=r$$

The most common approach utilizes the low rank factorization $X=UV^T$ for $U\in\mathbb{R}^{m\times r}$ and $V\in\mathbb{R}^{n\times r}$

$$\min_{U\in\mathbb{R}^{m\times r},V\in\mathbb{R}^{n\times r}} ||\mathcal{A}(UV^T)-b||^2$$

Because r is typically much smaller than the size of the matrix, this greatly reduces the number of variables.

In addition to finding a matrix of a given rank, this technique can be used in nuclear norm minimization as well [18, 22, 23]. The nuclear norm can be characterized as follows:

$$\begin{aligned} ||X||_* = \quad & \min_{U,V} \quad \tfrac{1}{2}\left(||U||_F^2+||V||_F^2\right) \\ & \text{subject to} \quad X=UV^T \end{aligned}$$

and so, to minimize a weighted sum of the nuclear norm and a quadratic loss function, we can minimize the following

$$\min_{U\in\mathbb{R}^{m\times r},V\in\mathbb{R}^{n\times r}} \frac{1}{2}\left(||U||_F^2+||V||_F^2\right)+\frac{\beta}{2}||\mathcal{A}(UV^T)-b||^2$$

1.1 Contributions

In this paper, we consider the following general relaxation to the rank minimization

$$\begin{aligned} \min_X \quad & \sum_{i=1}^{n}\rho(\lambda_i(X))+\phi(X) \\ \text{subject to} \quad & \mathcal{A}(X)=b \\ & X\succeq 0 \end{aligned} \tag{1}$$

where $\lambda_i(X)$ denotes the i^{th} eigenvalue of X. We impose the following assumptions on all ρ throughout the paper.

Assumption 1 For a function $\rho:[0,\infty)\to[0,\infty)$,

(i) ρ is concave
(ii) ρ is monotonically increasing
(iii) $\rho(0)=0$
(iv) For all $x\in[0,\infty)$, every subgradient of ρ is finite. Because ρ is concave, it is sufficient to say

$$\lim_{x \to 0^+} \sup_{w \in \partial\rho(x)} w = \kappa < +\infty$$

Additionally, we may also impose one or both of the following two assumptions:

Assumption 2 The function $\rho(x)$ is strictly concave on $[0, \infty)$.

Assumption 3 The function $\rho(x)$ is differentiable on $[0, \infty)$.

Examples of functions meeting these assumptions that are commonly used as surrogates to the l_0 norm are shown in Table 1. For each of the functions listed with the exception of the Shatten-p norm and the LogDet relaxation, the derivative approaches 0 for large values of x, which would expect to greatly reduce the estimation bias.

To simplify notation, when applied to a positive semidefinite matrix, the function $\rho : \mathcal{S}^n_+ \to \mathbb{R}_+$ is the sum of the regularizer ρ applied to the eigenvalues of the matrix. That is,

$$\rho(X) = \sum_i^n \rho(\lambda_i(X))$$

In this paper, we show by construction that for any regularizer meeting Assumption 1, the optimization problem (1) can be posed as a bi-convex optimization problem. Our bi-convex formulation serves as an abstraction of that presented by Mohan and Fazel [17], and can be used to derive similar iterative reweighted problems. Using our abstraction, we are able to utilize the low-rank factorization method for

Table 1 Examples of typical concave relaxations used in sparse optimization and their supergradients. For each regularizer, γ is a positive parameter. For SCAD, we take $\beta > 1$, and for the Schatten-p norm, $0 < p \leq 2$. Each of these functions satisfies Assumption 1

	$\rho(x)$	$\partial\rho(x)$
Trace inverse [8]	$1 - \frac{\gamma}{\gamma+x}$	$\frac{\gamma}{(\gamma+x)^2}$
Capped l_1 norm [29]	$\min(\gamma x, 1)$	$\begin{cases} \gamma, & x < \frac{1}{\gamma} \\ [0, \gamma] & x = \frac{1}{\gamma} \\ 0, & x \geq \frac{1}{\gamma} \end{cases}$
LogDet [7, 17]	$\log(x + \gamma)$	$\frac{\gamma}{\gamma+x}$
Schatten-p norm [12]	$(x + \gamma)^{\frac{p}{2}}$	$\frac{p}{2\lambda}(x + \gamma)^{\frac{p}{2}-1}$
SCAD [6]	$\begin{cases} \gamma x & x \leq \gamma \\ \frac{-x^2+2\gamma\alpha x-\gamma^2}{2(\alpha-1)} & \gamma \leq x \leq \alpha\gamma \\ \frac{\gamma^2(\alpha+1)}{2} & x > \alpha\gamma \end{cases}$	$\begin{cases} \gamma & x \leq \gamma \\ \frac{\alpha\gamma-x}{(\alpha-1)} & \gamma \leq x \leq \beta\gamma \\ 0 & x > \alpha\gamma \end{cases}$
Laplace [26]	$1 - e^{-\gamma x}$	$\gamma e^{-\gamma x}$

solving SDPs proposed by Burer and Monteiro [3] in order to reduce the number of variables to $O(nr)$ where r is an upper bound on the rank of the matrix, and extend the results to rectangular matrices as well. We derive algorithms based on the low rank factorization and prove convergence.

1.2 Previous works on nonconvex approaches to rank minimization

In order to more closely approximate the rank of a matrix, Fazel et al. proposed the LogDet heuristic for positive semidefinite rank minimization [7]. Instead of a convex function, the authors use the following smooth, concave function as a surrogate for the rank function.

$$\log(\det(X+\gamma I)) = \sum_{i=1}^{n} \log(\lambda_i(X)+\gamma)$$

where γ is a positive parameter. While nonconvex, the authors put forwards a Majorize-Minimization (MM) algorithm to find a local optimum. At each iteration, the first order Taylor expansion centered at the previous iterate is solved as a surrogate function. The algorithm is simplified to solving the following SDP at each iteration.

$$\begin{aligned} X^{(\mathrm{k}+1)} = \underset{X}{\operatorname{argmin}} \ & \left\langle W^{(k)}, X \right\rangle \\ \text{subjected to} \quad & \mathcal{A}(X) = b \\ & X \succeq 0 \end{aligned}$$

where $W^{(k)} = (X^{(k-1)} + \delta I)^{-1}$. We can view this algorithm as an iterative reweighting of the nuclear norm. The iterative reweighted scheme was later generalized by Mohan and Fazel [17] to minimize a class of surrogate functions known as the smooth Schatten-p function, defined as

$$f_q(X) = \mathrm{Tr}(X+\gamma I)^{\frac{p}{2}} = \sum_{i=1}^{n} (\lambda_i(X)+\gamma)^{\frac{p}{2}}$$

for $0 < p \leq 2$. The weight matrix for the Schatten-p function is $W^{(k)} = (X^{(k-1)} + \gamma I)^{\frac{p}{2}-1}$. Mohan and Fazel extend the algorithm for non square matrices by solving

$$\begin{aligned} X^{(k+1)} = \underset{X}{\operatorname{argmin}} \ & \langle W^{(k)}, X^T X \rangle \\ \text{subject to} \quad & \mathcal{A}(X) = b \end{aligned} \tag{2}$$

where $W^{(k)} = (X^{(k-1)^T} X^{(k-1)} + \gamma I)^{-1}$ at each iteration. The authors prove asymptotic convergence of the iterative reweighted algorithm for $0 \leq p \leq 1$. While this algorithm does give superior computational results, it can be very time consuming in the positive semidefinite case and will not scale well for large problems. We show in Sect. 5 how this can be improved by taking advantage of the low rank property of X.

In recent years, many functions have been proposed as alternative non-convex surrogates to the rank function in addition to the logdet heuristic. Zhang et al. [34]

proposed minimizing the truncated nuclear norm for a general matrix $X \in \mathbb{R}^{m \times n}$, defined for a fixed constant r as

$$||X||_{r,*} = \sum_{i=r+1}^{\min(m,n)} \sigma_i(X)$$

where $\sigma_i(X)$ denotes the i^{th} largest singular value. If we consider the large singular values to represent the signal and the small singular values the noise, as in the case of noisy image reconstruction, then this minimizes only the noise.

The idea of minimizing a concave function of the eigenvalues has been generalized by Lu et al. [4, 15], to any monotonically increasing and Lipschitz differentiable function. These works consider an unconstrained problem with a general loss function $\phi(X)$.

$$\min_X \sum_{i=1}^{min(m,n)} \rho(\sigma_i(X)) + \phi(X)$$

As with the LogDet algorithm, one can derive an MM algorithm using the first order Taylor expansion about the objective function. The authors include a proximal term. At each iteration, the authors propose solving the following problem

$$\begin{aligned} X^{k+1} =& \min \sum_{i=1}^{\min(m,n)} w_i \sigma_i(X) + \langle \nabla \phi(X^k), X - X^k \rangle + \frac{\mu}{2} ||X - X^k|| \\ =& \min \sum_{i=1}^{\min(m,n)} w_i \sigma_i(X) + \frac{\mu}{2} ||X - Y|| \end{aligned}$$

where $Y = X^k - \nabla \phi(X^k)$ and $w_i = \rho'_\gamma(\sigma_i(X^k))$. Much like the popular Singular Value Thresholding method put forth by Cai, Candès, and Shen [11], this has a closed form involving the shrinkage operator defined as $\mathcal{S}_t(\Sigma) = \text{Diag}(\Sigma_{ii} - t_i)_+$. The authors prove that the subproblem has a closed form solution

$$X^{k+1} = U \mathcal{S}_{\gamma w}(\Sigma) V^T$$

where $U \Sigma V^T$ is the singular value decomposition of Y.

The shrinkage operator, however, requires computing the singular value decompositon of a possibly very large matrix, which can be time consuming and inefficent even when only the top few singular values are needed. Similar algorithms presented by Yao et al. address this problem by showing one only needs to find the singular value decomposition of a much smaller matrix, making the method suitable for large scale problems [31, 32].

2 Equivalent biconvex formulation

It was shown by Mohan and Fazel [17] that the LogDet heuristic can be reformulated as a bi-convex problem with an additional variable W as follows

$$\begin{aligned} \min_{X,W} \quad & \langle X, W\rangle + \gamma \text{trace}(W) - \log\det(W) \\ \text{subject to } & A(X) = b \\ & X \succeq 0 \\ & I \succeq W \succeq 0 \end{aligned} \tag{3}$$

This allowed the authors to reformulate the MM algorithm outlined in Eq. (2) as an alternating method, which was of use when showing convergence of the algorithm. We now show that an extension of this reformulation can be used for any surrogate to the rank function satisfying Assumption 1.

Proposition 1 *For a function ρ satisfying Assumption 1, consider the following biconvex semidefinite program*

$$\begin{aligned} \min_{X,W} \quad & \langle X, W\rangle + G(W) + \phi(W) \\ \text{subject to } & \mathcal{A}(X) = b \\ & X \succeq 0 \\ & \kappa I \succeq W \succeq 0 \end{aligned} \tag{4}$$

where $\kappa = \sup \partial\rho(0)$, the function $G : \mathbb{S}^n_+ \to \mathbb{R}$ defined as $G(W) = \sum g(\lambda_i(W))$ satisfies the following condition:

$$\partial g(w) = \{-x : w \in \partial\rho(x)\}. \tag{5}$$

Any KKT point X^* of the general nonconvex relaxation (1) can be used to construct a KKT point (X^*, W^*) of (4) where $W^* \in \partial\rho(X^*)$. Likewise, for any (X^*, W^*) pair that is a KKT point of (4), X^* is a KKT point of (1) and $W^* \in \partial\rho(X^*)$.

Remark 1 In previous works, it has been shown that the rank minimization problem (1) is equivalent to the following semidefinite program with complementarity constraints:

$$\begin{aligned} \min_{X,U} \quad & n - \text{trace}(U) + \phi(X) \\ \text{subject to} \quad & \langle X, U\rangle = 0 \\ & \mathcal{A}(X) = b \\ & X \succeq 0 \\ & 0 \preceq U \preceq I \end{aligned} \tag{6}$$

Intuitively, the eigenvalues of the matrix $I - U$ are the l_0 norm of the eigenvalues of X, which implies that $n - \text{trace}(U)$ is the rank of X [13, 20, 21]. Shen and Mitchell [21] studied the problem when the complementarity constraint is relaxed as a penalty term.

$$\begin{array}{ll} \min\limits_{X,U} & n - \text{trace}(U) + \gamma\langle X, U\rangle + \phi(X) \\ \text{subject to} & \mathcal{A}(X) = b \\ & X \succeq 0 \\ & 0 \preceq U \preceq I \end{array} \tag{7}$$

The penalty formulation is a biconvex semidefinite program in the form of (4), with $W = \frac{1}{\gamma}U$ and $G(W) = -\frac{1}{\gamma}\text{trace}(W)$. This is equivalent to the semidefinite program (1) with $\rho(x)$ being the capped l_1 norm, $min(\frac{1}{\gamma}x, 1)$

We want to work with the derivative of the inverse of the derivative of $\rho(x)$, but this is only defined as stated if $\rho_\gamma(x)$ satisfies Assumptions 2 and 3. Under only Assumption 1, we define the function

$$q(t) \ := \ \inf\{x \in [0, \infty) \ : \ t \in \partial\rho_\gamma(x)\}. \tag{8}$$

Note that if $t \geq \kappa$ then $t \in \partial\rho(0)$, so $q(t) = 0$ for $t \geq \kappa$. The function $q(t)$ is defined for $t > \beta$, since $\rho(x)$ is concave; $q(\beta)$ is also defined if β is attained. We let J denote the domain of $q(t)$ and $\bar{J} := \{w \in J \ : \ w \leq \kappa\}$. Note that $q(t)$ is lower semicontinuous; it is continuous if Assumptions 2 and 3 hold, in which case it is the inverse function of the derivative of $\rho(x)$ for $t \in \bar{J}$. We can now define the function $g : J \to [0, \infty)$ as

$$g(w) \ := \ \int_w^\kappa q(t)dt. \tag{9}$$

Lemma 1 *The function $g(w)$ is decreasing and convex on its domain J. It is strictly convex for $w \leq \kappa$ if Assumption 3 holds. It is differentiable for $w \leq \kappa$ if Assumption 2 holds.*

Lemma 2 *For each $x \in [0, \infty)$, there exists $w \in \partial\rho(x)$ such that*

$$-x \ \in \ \partial g(w). \tag{10}$$

Further, the subdifferential is given by

$$\partial g(w) \ = \ \{-x \ : \ w \in \partial\rho(x)\}. \tag{11}$$

If $\rho(x)$ also satisfies Assumptions 2 and 3 then

$$g'((\rho')^{-1}(x)) \ = \ -x. \tag{12}$$

Example 1 Let $\rho(x)$ be the continuous nondifferentiable function

$$\rho(x) \ = \ \begin{cases} 4x & \text{if } 0 \leq x \leq 2 \\ 6x - x^2 & \text{if } 2 \leq x \leq 3 \\ 9 & \text{if } x \geq 3 \end{cases}$$

which is nondifferentiable at $x = 2$ and is only strictly concave for $x \in [2, 3]$. We have $\beta = 0$ and $\kappa = 4$. Then

$$q(t) = \begin{cases} 3 - \frac{1}{2}t & \text{if } 0 \le t \le 2 \\ 2 & \text{if } 2 \le t < 4 \\ 0 & \text{if } t \ge 4 \end{cases}$$

and

$$g(t) = \begin{cases} 9 + \frac{1}{4}t^2 - 3t & \text{if } 0 \le t \le 2 \\ 2(4 - t) & \text{if } 2 \le t \le 4 \\ 0 & \text{if } t \ge 4 \end{cases}$$

Further,

$$\partial g(w) = \begin{cases} [-\infty, -3] & \text{if } w = 0 \\ \{\frac{1}{2}w - 3\} & \text{if } 0 < w \le 2 \\ \{-2\} & \text{if } 2 \le w < 4 \\ [-2, 0] & \text{if } w = 4 \\ \{0\} & \text{if } w > 4 \end{cases}$$

The lack of strict concavity on the two line segments leads to the two intervals of subgradients $\partial g(w)$ for $w = 0$ and $w = 4$. The nondifferentiability at $x = 2$ leads to multiple values of w having the same set of subgradients $\partial g(w)$, namely $\{2\}$ for $2 \le w < 4$.

Proofs of lemmas

Proof Proof of Lemma 1:

Monotonicity of $g(w)$ follows from the nonnegativity of $q(t)$.

To show convexity, we consider $w_1 < w_2$, with $w_1, w_2 \in J$, and $0 \le \lambda \le 1$. We have

$$\begin{aligned} g(\lambda w_1 + (1 - \lambda)w_2) &= \textstyle\int_{\lambda w_1 + (1-\lambda)w_2}^{\kappa} q(t)dt \\ &= \lambda \textstyle\int_{w_1}^{\kappa} q(t)dt + (1 - \lambda) \int_{w_2}^{\kappa} q(t)dt - \lambda \int_{w_1}^{\lambda w_1 + (1-\lambda)w_2} q(t)dt \\ &\qquad + (1 - \lambda) \textstyle\int_{\lambda w_1 + (1-\lambda)w_2}^{w_2} q(t)dt \\ &\le \lambda g(w_1) + (1 - \lambda)g(w_2) \\ &\qquad - \lambda(\lambda w_1 + (1 - \lambda)w_2 - w_1)\, g(\lambda w_1 + (1 - \lambda)w_2) \\ &\qquad + (1 - \lambda)(w_2 - \lambda w_1 + (1 - \lambda)w_2)\, g(\lambda w_1 + (1 - \lambda)w_2) \\ &\qquad \text{from monotonicity of } q(t) \\ &= \lambda g(w_1) + (1 - \lambda)g(w_2), \end{aligned}$$

so $g(w)$ is convex.

If Assumption 3 holds then $q(t)$ is strictly decreasing for $\beta < w_1 \le \kappa$, so the inequality above holds strictly, so $g(w)$ is strictly convex.

If Assumption 2 holds then $q(t)$ is continuous on J, so $g(w)$ is differentiable. □

Proof Proof of Lemma 2:

Since $g(w)$ is convex, the subdifferential of $g(w)$ for a slope $w \in J$ is defined as

$$\begin{aligned}
\partial g(w) &= \{\xi \,:\, \xi h \leq g(w+h) - g(w)\ \forall\, w+h \in J\} \\
&= \{\xi \,:\, \xi h \leq -\textstyle\int_w^{w+h} q(t)dt\ \forall\, w+h \in J\} \\
&= \{\xi \,:\, \xi h \leq -hq(w+h)\, \forall\, w+h \in J\} \\
&\qquad \text{from monotonicity of } q(t) \\
&= \{\xi \,:\, \xi \leq -q(w+h)\, \forall\, h > 0,\ w+h \in J\} \\
&\qquad \cap\ \{\xi \,:\, \xi \geq -q(w+h)\, \forall\, h < 0,\ w+h \in J\} \\
&= \{\xi \,:\, \xi \leq -x\ \forall\, x \in [0,\infty) \text{ with } w+h \in J \cap \partial f(x),\ h > 0\} \\
&\qquad \cap\ \{\xi \,:\, \xi \geq -x\ \forall\, x \in [0,\infty) \text{ with } w+h \in J \cap \partial f(x),\ h < 0\} \\
&= \{-x \,:\, w \in \partial f(x)\} \quad \text{from concavity of } \rho(x).
\end{aligned}$$

It follows that given $x \in [0,\infty)$, we can choose $\bar{w} \in \partial f(x)$, and we will have $-x \in \partial g(\bar{w})$.

If Assumptions 2 and 3 hold then $\partial\rho(x) = \{\rho'(x)\}$, x is the unique point with derivative $\rho'(x)$, and $g(w)$ is differentiable from Lemma 1. Setting $\bar{w} = \rho'(x)$, the Fundamental Theorem of Calculus implies that

$$g'(\rho'(x)) = -q(\rho'(x)) = -x,$$

as required. □

Before proving Proposition 1, we consider the following lemma.

Lemma 3 *Let X be a positive definite matrix. Let $G(W) = \sum_{i=1}^n g(\lambda_i(W))$ be a convex function for any matrix $W \in \mathbb{S}_+^n$. Let κ be a positive constant. If $\tilde{W}$ is a minimizer of*:

$$\begin{array}{ll} \min_{W \in \mathbb{S}_+^n} & \langle X, W\rangle + G(W) \\ \text{subject to} & 0 \preceq W \preceq \kappa I \end{array}$$

Then $\hat{W}$ is also a minimizer with the same objective value, where

$$\hat{W} = \sum_{i=1}^n \lambda_{n-i+1}(\tilde{W}) v_i v_i^T$$

and v_i is the eigenvector of X corresponding to the ith largest eigenvalue.

Proof First, note the $\hat{W}$ is a feasible point and $G(\hat{W}) = G(\tilde{W})$, as the two matrices have the same eigenvalues.

The proof relies on the Hoffman-Wielandt inequality [10], which states that for any symmetric matrices A and B,

$$||A - B||_F^2 \geq ||\lambda(A) - \lambda(B)||^2$$

where $\lambda(A)$ denotes the vector of eigenvalues of A in descending order. When applied to the matrices X and $-\tilde{W}$, we have

$$||X-(-\tilde{W})||_F^2 \geq \sum_{i=1}^{n}\left(\lambda_i(X)-\lambda_i(-\tilde{W})\right)^2 = \sum_{i=1}^{n}\left(\lambda_i(X)+\lambda_{n-i+1}(\tilde{W})\right)^2.$$

Expanding these terms gives us the following:

$$||X||_F^2+||\tilde{W}||_F^2+2\langle X,\tilde{W}\rangle \geq ||\lambda(X)||^2+||\lambda(\tilde{W})||^2+2\sum_{i=1}^{n}\lambda_i(X)\lambda_{n-i+1}(\tilde{W})$$

Using the fact that the Frobenius norm of a matrix is the norm of the eigenvalues, and using the simultaneous diagonalizability of $\hat{W}$ and X, we have:

$$\langle X,\tilde{W}\rangle \geq \sum_{i=1}^{n}\lambda_i(X)\lambda_{n-i+1}(\tilde{W}) = \langle X,\hat{W}\rangle$$

So, $\hat{W}$ is a feasible point with an objective value no more than that of $\tilde{W}$, and is also a minimizer. □

Additionally, we present the technical lemma about the gradient of the objective function in (1), which is paramount when deriving algorithms and optimality conditions. First and second derivatives of the eigenvalue function have been studied extensively by Mangus [16] and Andrew et al. [2].

Lemma 4 *Let v_i denote the eigenvector corresponding to the i^{th} eigenvalue of X. If $\lambda_i(X)$ is a simple eigenvalue,*

$$\frac{d}{dX}\lambda_i(X) = v_i v_i^T \tag{13}$$

If $\lambda_i(X)=\lambda_{i+1}(X)=\cdots=\lambda_{i+k}(X)$, then

$$\frac{d}{dX}\sum_{j=0}^{k}\lambda_{i+j}(X) = \sum_{j=0}^{k}v_{i+j}v_{i+j}^T$$

Lemma 4 allows us to easily compute the subgradient of the objective function.

$$\partial\rho(X) = \left\{ V\mathrm{diag}(y_1,y_2,\ldots y_n)V^T \;\middle|\; y_i \in \partial\rho(\lambda_i(X))\right\} \tag{14}$$

where V denotes the matrix of eigenvectors of X. We can now prove Proposition 1.

Proof We start by considering KKT points of (4). The feasible pair (X, W) is a KKT point if there exists a subgradient Z of $G(W)$ such that

$$0 \preceq W \perp X+Z+Y \succeq 0 \tag{15a}$$

$$0 \leq X \perp \sum_i \mu_i A_i + W + \nabla\phi(X) \geq 0 \tag{15b}$$

$$0 \leq Y \perp \kappa I - W \geq 0 \tag{15c}$$

By Lemma 4, if W has eigenvectors V^W and eigenvalues $w_1, w_2, \ldots, w_n$, then

$$\partial G(W) = \left\{ V^W \text{diag}(z_1, z_2, \ldots, z_n) V^{W^T} \;\middle|\; z_i \in \partial g(w) \right\}.$$

We start by claiming that X and W (and hence Z and Y) are simultaneously diagonalizable by citing Lemma 3. Equation (15c) shows that Y and W are simultaneously diagonalizable. Hence X, W, Y, and Z are all simultaneously diagonalizable, and the KKT conditions (15a) and (c) simplify to the following.

$$0 \leq \lambda_i(W) \perp \lambda_i(X) + \lambda_i(Z) + \lambda_i(Y) \geq 0 \quad \forall i = 1, ..., n \tag{16a}$$

$$0 \leq \lambda_i(Y) \perp \kappa - \lambda_i(W) \geq 0 \quad \forall i = 1, ..., n \tag{16b}$$

If $0 < \lambda_i(W) < \kappa$, we have that $\lambda_i(Y) = 0$, and so Eqs. (16a) and (b) are satisfied if $\lambda_i(X) + \lambda_i(Z) = 0$. By construction of g from Lemma 2, there exists $w_i \in \partial\rho(\lambda_i(X))$ and $z_i \in -\partial g(w_i)$ such that $\lambda_i(Z) = z_i$, $\lambda_i(W) = w_i$ is a solution.

When the upper bound on the eigenvalue of W is an active constraint, i.e. when $\lambda_i(W) = \kappa$, then there exists $z_i \in \partial g(\kappa)$ such that $\lambda_i(X) + z_i \leq 0$. Because $\partial g(\kappa) = \{0\}$, $\lambda_i(X) = 0$, which is to say $\lambda_i(W) \in \partial\rho(\lambda_i(X))$.

Finally, we consider when $\lambda_i(W) = 0$. Equation (16a) becomes

$$\lambda_i(X) \geq -z_i$$

for some $z_i \in \partial g(0)$. By Lemma 2, we have that $0 \in \partial\rho(-z_i)$. Because ρ is concave and nondecreasing, if $0 \in \partial\rho(x_1)$, then $0 \in \partial\rho(x_2)$ for all $x_2 \geq x_1$, and so $\lambda_i(W) = 0 \in \partial\rho(\lambda_i(X))$.

We can now say that, in general, any *KKT* point satisfies $\lambda_i(W) \in \partial\rho(\lambda_i(X))$ for $i = 1, .., n$, and by Lemma 4, $W \in \partial\rho(x)$. The KKT conditions for (1) state that there exists a $U \in \partial\rho(x)$ such that

$$0 \leq X \perp \sum \mu_i A_i + U + \nabla\phi(X) \geq 0$$

With the assignment $U = W$, it is clear that if (X, W) is a KKT point of (4), then X is a KKT point of (1).

Conversely, consider any X that is a KKT of (1) with dual variable μ. Then, the assignment $W \in \partial\rho(x)$ and $Y = 0$ satisfy (15b) and (15c). By Lemma 2, we have that there exists a $Z \in \partial G(W)$ such that $Z = -X$ and (15a) is satisfied. □

Such a function is shown for various choices of nonconvex regularizers in Table 2, and can be easily verified by showing that Eq. (5) holds. We note that the function $G(W)$ is used primarily for theoretical analysis and derivation of algorithms. In practice, one only needs the function $\rho'(x)$.

Table 2 Function $G(W)$ and constants κ that satisfy the conditions in Proposition 1 for various concave relaxations of the rank function

	$\partial g(w)$	$G(W)$	κ
Trace inverse	$\gamma + w^{-\frac{1}{2}}$	$\text{trace}(\gamma W - 2\sqrt{W})$	$\frac{1}{\gamma}$
Capped l_1 norm	$\frac{-1}{\gamma}$	$\frac{-1}{\gamma}\text{trace}(W)$	γ
LogDet	$\gamma - \frac{\gamma}{w}$	$\gamma\text{trace}(W) - \gamma \log\det(W)$	1
Schatten-p norm	$\gamma - \left(\frac{w}{p}\right)^{\frac{1}{p-1}}$	$\text{trace}(\gamma W - \frac{2-p}{p} W^{\frac{p}{p-2}})$	$\frac{p}{2\gamma}\gamma^{\frac{p}{2}-1}$
SCAD	$-\alpha\gamma + (\alpha - 1)w$	$\text{trace}(\frac{\alpha-1}{2}W^2 - \alpha\gamma W)$	γ
Laplace	$-\frac{1}{\gamma}\log(\frac{w}{\gamma})$	$\sum_i \frac{\lambda_i(W)}{\gamma}(\log(\frac{\lambda_i(W)}{\gamma}) - 1)$	γ

2.1 Low-rank factorization

While the MM algorithm is efficient in the non-symmetric case, with each iteration having closed form updates which can be calculated in $O(nm^2)$ time, the algorithm is not scalable in the positive semidefinite case, as it needs to solve a semidefinite program at each iteration. Instead, we take advantage of the low rank factorization for semidefinite programs as presented by Burer and Monteiro [3] and utilized to solve the nuclear norm minimization problem by Tasissa and Lai [25]. Let r be an upper bound on the rank of the matrix we seek to reconstruct. Then, if X is positive semidefinite, we have that there exists a matrix $P \in \mathbb{R}^{n\times r}$ such that $X = PP^T$.

$$\begin{aligned} \min_{P\in\mathbb{R}^{n\times r}, W\in\mathcal{S}^n_+} \quad & \langle PP^T, W\rangle + G(W) + \phi(PP^T) \\ \text{subject to} \quad & \mathcal{A}(PP^T) = b,\ 0 \preceq W \preceq \kappa I \end{aligned} \tag{17}$$

While X is replaced with a variable of drastically reduced size, W is left as a positive semidefinite matrix of size n. To reduce the size of W, we propose minimizing the rank of P^TP instead of the rank of PP^T.

$$\begin{aligned} \min_{P\in\mathbb{R}^{n\times r}, W\in\mathcal{S}^r_+} \quad & \langle P^TP, W\rangle + G(W) + \phi(PP^T) \\ \text{subject to} \quad & \mathcal{A}(PP^T) = b,\ 0 \preceq W \preceq \kappa I \end{aligned} \tag{18}$$

Intuitively, this should be equivalent due to the fact that the non-zero eigenvalues of PP^T are equivalent to the nonzero eigenvalues of P^TP. We prove this intuition in the following proposition.

Proposition 2 *Let $P^* \in \mathbb{R}^{n\times r}$ have the singular value decomposition $P^* = \sum_{i=1}^r v_i u_i^T \sigma_i^P$. If (P^*, W_n) is an optimizer of (17), then*

$$W_n^* = \sum_{i=1}^n \lambda_i^W v_i v_i^T = \sum_{i=1}^r \lambda_i^W v_i v_i^T + \kappa \sum_{i=r+1}^n v_i v_i^T,$$

and if (P^*, W_r) *is an optimizer of* (18)*, then* $W_r^* = \sum_{i=1}^r \lambda_i^W u_i u_i^T$*. Furthermore,* (P^*, W_r^*) *is an optimizer of* (18) *if and only if* (P^*, W_n^*) *is an optimizer of* (17)*.*

Proof We start by proving that W_n^* and W_r^* have the eigenvalue decompositions stated in the proposition. By the same reasoning as in Proposition 1, any matrix $W \in \partial\rho(P^*P^{*T})$ is an optimizer to the convex semidefinite program:

$$\begin{aligned} \min_{W \in \mathcal{S}_+^n} \quad & \langle P^*P^{*T}, W\rangle + G(W) \\ \text{subject to} \quad & 0 \preceq W \preceq \kappa I \end{aligned}$$

So, if $P^*P^{*T} = \sum_{i=1}^r (\sigma_i^P)^2 v_i v_i^T$, then there is a minimizer (P^*, W_n^*) such that W_n^* has the eigendecompositon $\sum_i^r \lambda_i^W v_i v_i^T + \sum_{i=r+1}^n \kappa v_i v_i^T$, where $\lambda_i^W \in \partial\rho\big((\sigma_i^P)^2\big)$ and $\kappa = \sup \partial\rho(0)$. Likewise, (P^*, W_r) is an optimizer of (18), then $W_r^* = \sum_{i=1}^r \lambda_i^W u_i u_i^T$ is an optimizer, where $\lambda_i^W \in \partial\rho\big((\sigma_i^P)^2\big)$.

Next, we will show that if $(\Delta P, \Delta W_r)$ was a feasible descent direction in (18) at (P^*, W_r^*), then we can construct a feasible direction for (17) at (P^*, W_n^*), and vice versa. If $(\Delta P, \Delta W_r)$ was a feasible descent direction, then, there exists a subgradient $Z_r \in \partial G(W_r^*)$ such that

$$2\langle P^*W_r^* + \nabla\phi(P^*P^{*T})P^*, \Delta P\rangle + \langle P^{*T}P^* + Z_r, \Delta W_r\rangle < 0. \tag{19}$$

We claim that $(\Delta P, \Delta W_n)$ is a descent direction in (17) with

$$\Delta W_n = V_r U^T \Delta W_r U V_r^T,$$

where $U \in \mathbb{R}^{r\times r}$ is the matrix whose columns are the eigenvectors of W_r, and $V \in \mathbb{R}^{n\times r}$ is the matrix whose columns are the first r eigenvectors of W_n. First note that, $P^*W_r = \sum_{i=1}^r v_i u_i \lambda_i^W \sigma_i^P = W_n^* P^*$, and ΔP is a feasible direction in (17).

Next, consider the gradient of the objective of (17) with respect to W,

$$P^*P^{*T} + \nabla G(W_n^*) = \sum_{i=1}^r v_i v_i^T\big(\sigma_i^P + z_i\big) + \sum_{i=r+1}^n v_i v_i^T(z_\kappa)$$

where $z_i \in \partial g(\lambda_i^W)$ and $z_\kappa \in \partial g(\kappa)$. Specifically, we chose $z_i = \lambda_i(Z_r)$, Z_r be the r by r matrix with eigenvectors U and eigenvalues $z_1, \ldots, z_r$ so that $Z_r \in G(W_r)$. By Lemma 2, $0 \in \partial g(\kappa)$, and so the rank r matrix

$$Z_n := V_r U^T\big(P^{*T}P^* + Z_r\big)UV_r^T$$

is a subgradient of G with respect to W_n. Consider the inner product of the gradient of the objective of (17) with respect to W_n and the proposed descent direction for W_n.

$$\begin{aligned} &\langle P^*P^{*T} + Z_n, \Delta W_n\rangle = \langle V_r U^T\big(P^{*T}P^* + Z_r\big)UV_r^T, \Delta W_n\rangle \\ =&\langle \big(P^{*T}P^* + Z_r\big), UV_r^T \Delta W_n V_r U^T\rangle \\ =&\langle \big(P^{*T}P^* + Z_r\big), \Delta W_r\rangle \end{aligned}$$

Combining these facts gives us that $(\Delta P, \Delta W_n)$ is a descent direction:

$$
\begin{aligned}
&2\langle W_n^* P^* + \nabla\phi(P^*P^{*T})P^*, \Delta P\rangle + \langle P^*P^{*T} + Z_n, \Delta W_n\rangle \\
=&2\langle P^* W_r^* + \nabla\phi(P^*P^{*T})P^*, \Delta P\rangle + \langle P^{*T}P^* + Z_r, \Delta W_r\rangle < 0
\end{aligned}
$$

The proof of the other direction is similar. □

2.2 Extension to nonsymmetric matrices

To extend these methods to general nonsymmetric matrices $X \in \mathbb{R}^{m\times n}$, we can minimize the rank of PSD matrix X^TX, as done by Mohan and Fazel [17]. However, this is computationally inefficient as each iteration requires finding the eigendecomposition of X^TX. With this in mind, we put forth a separate extension in which we minimize the rank of the following auxiliary variable

$$
Z = \begin{bmatrix} G & X^T \\ X & B \end{bmatrix}
$$

It was shown by Liu et al. that for any X, there exists G and B such that $\text{rank}(X) = \text{rank}(Z)$ and $Z \succeq 0$ [14]. We can thus solve the following minimization problem

$$
\begin{aligned}
\min_{Z,W} \quad & \langle Z, W\rangle + G(W) + \phi(X) \\
\text{subject to} \quad & \mathcal{A}(X) = b \\
& Z = \begin{bmatrix} G & X^T \\ X & B \end{bmatrix} \succeq 0 \\
& 0 \preceq W \preceq \kappa I
\end{aligned}
\tag{20}
$$

While inefficient on its own due to the matrix W being $(m+n)\times(m+n)$, this formulation allows us to utilize the Burer-Monteiro approach which allowed us to efficiently solve the semidefinite case in Algorithm 1. We utilize the same upper bound r on the rank of X as before and introduce the matrix $P \in \mathbb{R}^{(m+n)\times r}$ such that $Z = PP^T$. We decompose P into P_m and P_n such that $P = \begin{bmatrix} P_n \\ P_m \end{bmatrix}$ so that $X = P_m P_n^T$. As before, we minimize the rank of $P^TP = P_m^TP_m + P_n^TP_n$.

$$
\begin{aligned}
\min_{W,P_m,P_n} \quad & \langle P_m^TP_m + P_n^TP_n, W\rangle + G(W) + \phi(P_mP_n^T) \\
\text{subject to} \quad & \mathcal{A}(P_mP_n^T) = b, \quad 0 \preceq W \preceq \kappa I
\end{aligned}
\tag{21}
$$

We note that for the special case of minimizing the nuclear norm, $W = I$, we have the well known alternating minimization method when using a quadratic loss function [18, 22, 23] as follows:

$$
\min_{P_m,P_n} \quad ||P_n||_F^2 + ||P_m||_F^2 + \frac{\beta}{2}||\mathcal{A}(P_mP_n^T) - b||^2. \tag{22}
$$

3 Algorithms

In most practical applications, we expect noise in our measurements, and thus an equality constraint may not be practical. For the algorithms in this section, we restrict our focus to the problem of rank minimization with a quadratic loss function, $\phi(X) = \frac{\beta}{2}||\mathcal{A}(X) - b||_F^2$, and no linear constraints. Utilizing the low-rank factorization technique, for the case of non symmetric matrices, we seek to minimize

$$\begin{array}{ll} \min\limits_{X,W} & \langle P_m^T P_m + P_n^T P_n, W\rangle + G(W) + \frac{\beta}{2}||\mathcal{A}(P_m P_n^T) - b||^2 \\ \text{subject to} & 0 \preceq W \preceq \kappa I \end{array} \tag{23}$$

3.1 Alternating methods for rectangular matrices

While the formulation for rectangular matrices could be solved by simply using Algorithm 1, we propose an ADMM algorithm wherein we alternate over the variables P_m, P_n, and W. By doing so, the subproblems in P_m and P_n are strongly convex. The subproblems are as follows:

$$\begin{aligned} P_m^k &= \text{argmin}_{P_m} \langle P_m^T P_m, W\rangle + \frac{\beta}{2}||\mathcal{A}(P_m P_n^T) - b||_F^2 \\ P_n^k &= \text{argmin}_{P_n} \langle P_n^T P_n, W\rangle + \frac{\beta}{2}||\mathcal{A}(P_m P_n^T) - b||_F^2 \end{aligned} \tag{24}$$

The gradients of which can be calculated as

$$\begin{aligned} \nabla_{P_m} F(P_m, P_n, W) &= P_m W + \beta \mathcal{A}^*(\mathcal{A}(P_m P_n^T) - b) P_n \\ \nabla_{P_n} F(P_m, P_n, W) &= P_n W + \beta \mathcal{A}^*(\mathcal{A}(P_m P_n^T) - b)^T P_m \end{aligned}$$

where $F(P_m, P_n, W)$ is the objective function of (23).

The update for W is derived from Proposition 1, and is similar to that of other iteratively reweighted methods [7, 12, 17].

$$W^k = \nabla \rho(P^{kT} P^k)$$

Because we are minimizing the rank of the the smaller matrix $P^T P$, this update is calculated in $\mathcal{O}(r^3)$ operations.

Algorithm 1 Alternating Minimization for Rank Minimization with a General Nonconvex Regularizer (GenAltMin)

Input: $\mathcal{A}, b$
Output: Stationary point X of (23)
Initialization :$P^0 = \text{rand}(n, r)$, $W^0 = I$.
1: **for** $k = 1, .., $ **do**
2: Solve
$$P_m^k = \text{argmin}_{P_m} \langle P_m^T P_m, W\rangle + \frac{\beta}{2}||\mathcal{A}(P_m P_n^T) - b||_F^2$$
3: Solve
$$P_n^k = \text{argmin}_{P_n} \langle P_n^T P_n, W\rangle + \frac{\beta}{2}||\mathcal{A}(P_m P_n^T) - b||_F^2$$
4: $[V^k, \Sigma^k] = \text{eig}(P^{k^T} P^k)$
5: $W^k = V^k \rho'(\Sigma^k) V^{K^T}$
6: Check for Convergence
7: **end for**
8: **return** X

3.2 Alternating steepest descent

For alternating minimization without a regularizer, it has been shown computationally effective to, instead of solving subproblems to optimality, take one step in the gradient direction at each iteration [24]. For the P_n and P_m updates, we can calculate the steepest descent step size. Let d_m and d_n denote the gradient in the P_m and P_n subproblems. Then, the steepest descent step sizes t_m and t_n for each subproblem respectively are can be calculated as follows

$$t_m = \frac{\beta\langle \mathcal{A}(d_m P_n^T), \mathcal{A}(P_m P_n^T) - b\rangle + 2\langle P_m^T d_m, W\rangle}{\beta||\mathcal{A}(d_m P_n^T)||^2 + 2\langle d_m^T d_m, W\rangle}$$
$$t_n = \frac{\beta\langle \mathcal{A}(P_m d_n^T), \mathcal{A}(P_m P_n^T) - b\rangle + 2\langle P_n^T d_n, W\rangle}{\beta||\mathcal{A}(P_m d_n^T)||^2 + 2\langle d_n^T d_n, W\rangle}$$

Note that the step sizes can be calculated with $\mathcal{O}((m+n)r^2 + r|\Omega|)$ computations. Because solving W to optimality is computationally inexpensive by comparison, we update W in the same way as in Algorithm 1. The parameters β and γ are also updated in the previously mentioned way.

Algorithm 2 Alternating Steepest Descent with General Nonconvex Regularizer (GenASD)

Input: $\mathcal{A}, b$
Output: Stationary point X of (23)
Initialization : $P_n^0 = \text{rand}(n, r)$, $W^0 = I$.
for $k = 1, .., $ **do**
  $d_m^k = P_m^{k-1} W + \beta \mathcal{A}^*(\mathcal{A}(P_m^{k-1} P_n^{k-1\,T}) - b) P_n^{k-1}$
  $t_m^k = \frac{\beta \langle \mathcal{A}(d_m P_n^T), \mathcal{A}(P_m P_n^T) - b \rangle + 2 \langle P_m^T d_m, W \rangle}{\beta ||\mathcal{A}(d_m P_n^T)||^2 + 2 \langle d_m^T d_m, W \rangle}$
  $P_m^k = P_m^{k-1} - t_m^k d_m^k$
  $d_n^k = P_n^{k-1} W + \beta \mathcal{A}^*(\mathcal{A}(P_m P_n^{k-1\,T}) - b)^T P_m^k$
  $t_n^k = \frac{\beta \langle \mathcal{A}(P_m d_n^T), \mathcal{A}(P_m P_n^T) - b \rangle + 2 \langle P_n^T d_n, W \rangle}{\beta ||\mathcal{A}(P_m d_n^T)||^2 + 2 \langle d_n^T d_n, W \rangle}$
  $P_n^k = P_n^{k-1} - t_n^k d_n^k$
  $[V^k, \Sigma^k] = \text{eig}(P_n^T P_n + P_m^T P_m)$
  $W^k = V^k \rho'(\Sigma^k) V^{K\,T}$
  $\beta^{k+1} = \min\big(1.2\beta^k, \beta_{\max}\big)$, $\gamma^{k+1} = \max\big(0.8\gamma^k, \gamma_{\min}\big)$
  Check for Convergence
end for
return $X = P_m P_n^T$

3.3 Convergence

Each of the algorithms presented in this section is guaranteed to converge by the main result in [28]. Xu and Yin show convergence of coordinated block descent algorithms to solve nonconvex optimization problems of the following form:

$$\min_{x \in \mathcal{X}} F(x_1, \ldots, x_s) \equiv f(x_1, \ldots, x_s) + \sum_{i=1}^{s} s_i(x_i) \tag{25}$$

Denote

$$f_i^k(x_i) = f(x_1^m, \ldots, x_{i-1}^k, x_i, x_{i+1}^{k-1}, \ldots x_s^{k-1})$$

and

$$\mathcal{X}_i^k(x_i) = \mathcal{X}(x_1^m, \ldots, x_{i-1}^k, x_i, x_{i+1}^{k-1}, \ldots x_s^{k-1}).$$

Xu and Yin analyze three types of updates:

$$x_i^k = \underset{x_i \in \mathcal{X}_i^k}{\text{argmin}}\, f_i^k(x_i) + r_i(x_i) \tag{26}$$

$$x_i^k = \underset{x_i \in \mathcal{X}_i^k}{\text{argmin}}\, f_i^k(x_i) + \frac{L_i^{k-1}}{2} ||x_i^{k-1} - x_i^{k-2}||^2 + r_i(x_i) \tag{27}$$

$$x_i^k = \underset{x_i \in \mathcal{X}_i^k}{\operatorname{argmin}} \langle \nabla f_i^k(\hat{x}_i^{k-1}), x_i \rangle + \frac{L_i^{k-1}}{2} ||x_i - \hat{x}_i^{k-1}||^2 + r_i(x_i) \tag{28}$$

where $\hat{x}_i^{k-1} = x_i^{k-1} + w^k(x_i^{k-1} - x_i^{k-2})$, and $w^k \geq 0$ is the extrapolation weight.

The authors assume that F is continuous, bounded, and has a minimizer. Additionally, they make assumptions on f_i^k depending on the type of update used. For the standard update (26), f_i^k must be strongly convex, and for the proximal linear update (28), ∇f_i^k must be L_i^k-Lipshitz differentiable. For the proximal update (27), no additional assumptions are made; f_i^k need not even be convex.

In both of the algorithms presented in this section, the W update is solved to optimality, and thus $G(W)$ is required to be strongly convex. As shown in Lemma 1, this is satisfied for any differentiable regularizer satisfying Assumption 1.

In Algorithm 1, we utilize the standard update, and so our objective function must be strongly convex. Because the quadratic loss function is block convex in both P_m and P_n, it typically samples a small portion of the matrix and will not be strongly convex. However, the terms $\langle P_m^T P_m^T, W^k \rangle$ and $\langle P_n^T P_n^T, W^k \rangle$ are strongly convex so long as W^k is full rank. Assumption 2 is then necessary to ensure convergence, as strong concavity in ρ ensures ρ is strictly increasing and that that $0 \notin \partial \rho(x)$ for any finite x.

Lastly, because $\nabla_{P_m} F$ and $\nabla_{P_n} F$ are linear, Algorithm 2 converges.

While the capped l_1 norm is non-differentiable, meaning none of the algorithms in this section are guaranteed to converge when using it as the regularizer, one can modify the algorithms slightly so that it does converge as in Shen and Mitchell [21]. The authors utilize the proximal linear update for W as follows:

$$W^{k+1} = \underset{0 \leq W \leq I}{\operatorname{proj}} \left(W^k + \frac{1}{L^k}(X^{k+1} + \gamma I) + w^k(W^k - W^{k-1}) \right)$$

When this update is used in any of the algorithms in this section, convergence is guaranteed without assuming differentiablity of the regularizer.

4 Numerical results

Algorithms 1 and 2 were implemented in MATLAB R2018b, and the source code to run the algorithms and reproduce every result in this section is publicly available at https://github.com/april1729/GenAltMin. The numerical experiments were conducted on a Dell Laptop running Windows 10 with 16 GB of ram and an Intel Core i3-4030U CPU @ 1.90 GHz.

4.1 Synthetic data for rectangular matrices

We now test Algorithms 1 and 2 utilizing synthetically generated low rank matrices with additive Gaussian noise. Throughout this section, we generate a matrix of size m by n with rank r and noise parameter d by the following Matlab command:

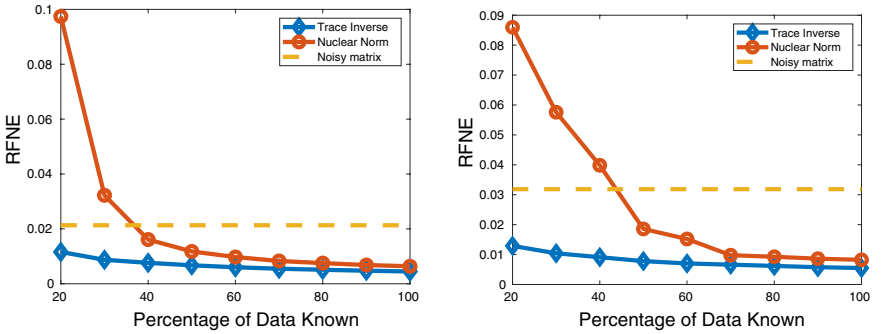
```
M = randn(m,r) * randn(r,n) + d * randn(m,n)
```

Figures 1a and b show the Relative Frobenius Norm Error (RFNE) of the solution recovered by the nuclear norm and by the trace inverse regularizer with varying percentages of known data, along with the relative Frobenius norm of the noise matrix as a baseline. We plot these results for a 300 by 200 matrix and a 1000 by 500 matrix, each averaged over 10 randomly generated instances. In both figures, the trace inverse is able to outperform the baseline when only 20% of the data is available. Note that in each case, the trace inverse regularizer outperforms the nuclear norm.

To show that the superiority of the nonconvex regularizer is not just for certain choices of β, we show how each method performs for values of β between 10^{-3} and 10 for the smaller problem and 10^{-4} and 1 for the larger problem in Fig. 2a and b respectively. When the parameter is differed by an orders of magnitude, the results for the trace inverse regularizer are hardly affected, while the accuracy of the optimal solution to the nuclear norm problem varies a significant amount. In fact, every value of β for the trace inverse regularizer outperformed the optimal value of β for the nuclear norm regularizer.

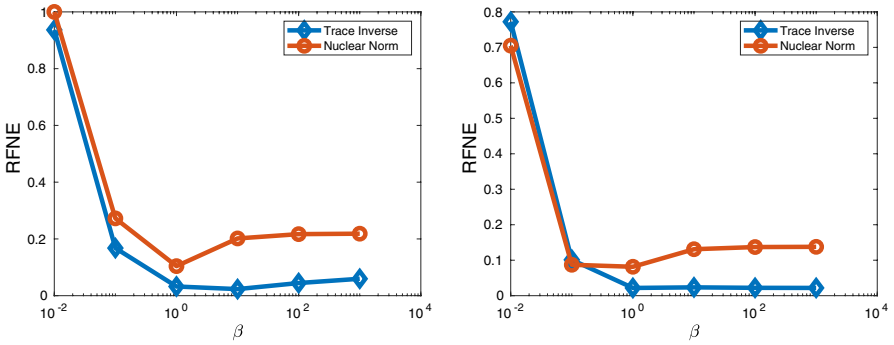
In order to illustrate the estimator bias of the nuclear norm formulation compared to nonconvex approaches, we plot the singular values of the reconstructed matrix utilizing both the trace inverse regularizer and the nuclear norm, along with the singular values of the original matrix. We show this plot for varying values of β of for a 300 by 200 matrix with rank 5 in Fig. 3. We plot the first r singular values and the next r singular values on a different scale, where r is the rank of the matrix being recovered.

For values of β that are smaller than 0.01, the solution is the zero matrix, and for values of β larger than 0.1, the solution is not the correct rank. As expected, there is a very small range in which we obtain a matrix with the correct rank. Additionally,



(a) $m = 300, n = 200, r = 5, d = 0.05$ **(b)** $m = 1000, n = 500, r = 10, d = 0.1$

Fig. 1 RFNE of the matrix recovered from Algorithm 2 using both the nuclear norm and trace inverse regularizer for varying amounts of data known, along with the RFNE of the noise



(a) $m = 300, n = 200, r = 5,$ $d = 0.05, p = 0.2$

(b) $m = 1000, n = 500, r = 10,$ $d = 0.1, p = 0.2$

Fig. 2 RFNE for varying amounts of data known for Algorithm 2 for both the nuclear norm and trace inverse regularizer, along with the RFNE of the noise. The two figures show the results for different trade off parameters

when the nuclear norm algorithm gives a matrix with the correct rank, the singular values reconstructed using the nuclear norm are noticeably smaller. This is due to the fact that the nuclear norm puts equal weight on minimizing each singular value, including the ones that should not be zero. So, by increasing β, the singular values that are supposed to be zero become larger, and by decreasing β, the singular values that are not supposed to be zero become too small.

By contrast, the top r singular values for the matrix reconstructed with Algorithm 2 are approximately equal to the singular values of the original matrix. For values of β less than 0.01 in the first case and 0.001 in the second case, the solution to the trace inverse formulation is the correct rank. As opposed to the convex relaxation, the nonconvex method has a sufficiently large range of β that give a matrix of the correct rank.

While this shows that the nonconvex formulations are significantly more robust to the choice of β, one may wonder if the added parameter controlling the curvature of the regularizer, γ, may contribute to more variability with parameter choices. Figure 4 shows the RFNE for choices of γ distributed between 0.03125 and 256. Surprisingly, the figure shows that for a large range of choices of γ, the results are identical. It is only at $\gamma = 0.125$ that the nonconvex formulation loses the stability it usually has. This behavior is expected due to the fact that the trace inverse regularizer converges to the rank function as γ approaches 0. For values of γ larger than the smallest non-zero singular value of the original low rank matrix (roughly 200), the trace inverse formulation behaves more similarly to the nuclear norm, which one could also expect as the derivative of the nonconvex regularizer is approximately a constant for large values of γ.

Due to the remarkable consistency of the algorithm for varying choices of γ, parameter tuning is not an issue in practice. Ideally, the choice of γ would be approximately half of the largest nonzero singular value of the original low rank matrix so that the gradient of the regularizer is small for the top r singular values.

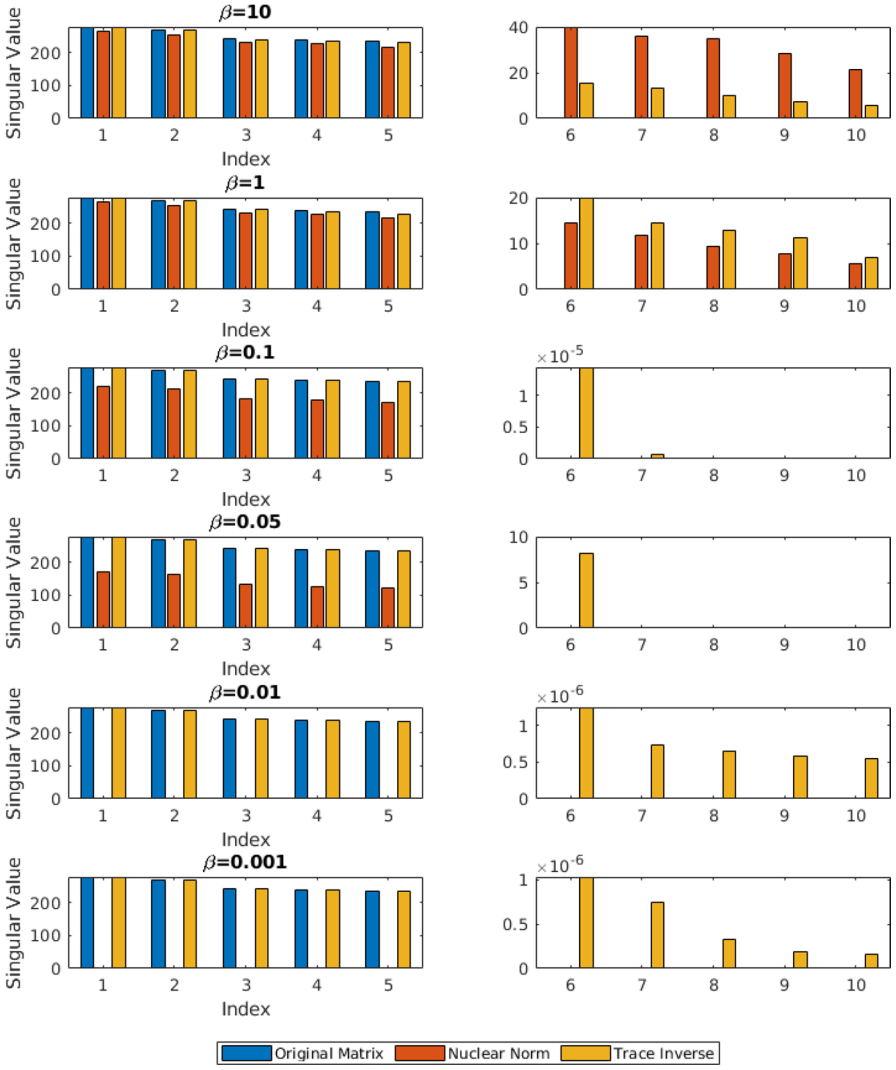
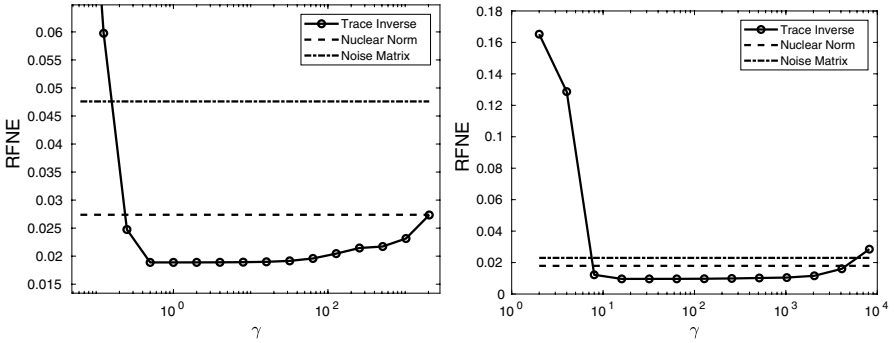


Fig. 3 Singular value distribution for the matrices recovered utilizing Algorithm 2 with the trace inverse regularizer and nuclear norm regularizer with $m = 300, n = 200, r = 5, d = 0.05$, and $p = 0.2$

While this quantity cannot be directly measured with incomplete noisy data, it can be (very roughly) approximated as follows:

$$\gamma = \frac{1}{2\sqrt{rp}} ||P_{\Omega}(\tilde{M})||_F$$

where r is a rough estimate of the rank of the matrix. Note that, unlike rank constrained optimization methods which rely heavily on the rank of the matrix to be



(a) $m = 300, n = 200, r = 5$ $p = 0.3, d = 0.1$

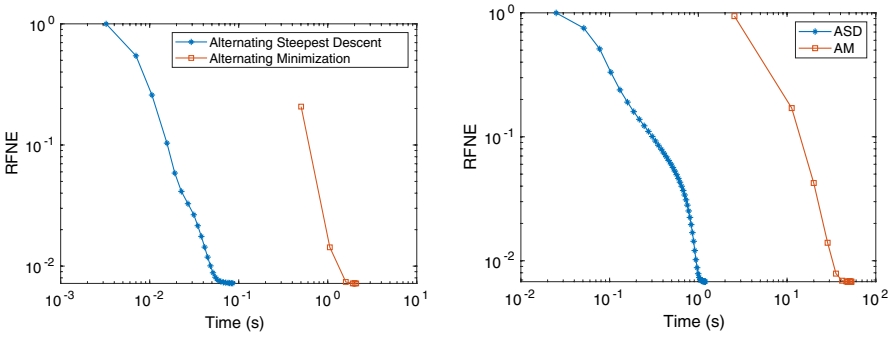
(b) $m = 1000, n = 500, r = 5$ $p = 0.1, d = 0.05$

Fig. 4 RFNE of the matrix recovered by Algorithm 2 utilizing the trace norm regularizer with values of γ between 2^{-4} and 2^{8} in the left plot, and between 2^{1} and 2^{13} in the right plot, along with the RFNE of the noisy matrix and the optimal value to the nuclear norm minimization problem

recovered being known exactly, Fig. 4 indicates that our method will perform well even when the estimate of the rank is off by orders of magnitude.

Before moving on to larger, real data sets, we demonstrate the difference in speed between Algorithm 1 and Algorithm 2. Figures 5a and b plot the convergence of the two algorithms on matrices that are 300 by 200 and 1000 by 500 respectively. First, note that in both figures the two methods converge to the same local optima, suggesting one need not worry about the difference in quality of the output between the two algorithms.

For the smaller case, while clear that taking only one step converges faster than solving the subproblems to optimality, they both converge in under 2 seconds. When solving the subproblems to optimality, however, only 4 iterations are needed to



(a) $m = 300, n = 200, r = 5$ $p = 0.4, d = 0.05$

(b) $m = 1000, n = 500, r = 15$ $p = 0.2, d = 0.01$

Fig. 5 Convergence of Algorithm 1 and Algorithm 2. The RFNE and cumulative runtime is recorded at each iteration

Table 3 Comparison of four different matrix completion algorithms on randomly generated low rank matrices. The algorithm LMaFit reconstructs a matrix of a given rank k. The table shows the results when the algorithm is given the exact rank($k = r$) and an incorrect rank ($k = 2r$)

			GenASD		FaNCL		FPC	LMaFit	
r	Noise	p	Trace inverse	SCAD	Log Sum	Capped L1 norm		k=r	k=2r
m=300, n=200									
5	0.05	0.1	0.0234	0.0232	0.0994	0.0546	0.2374	0.0273	0.2892
5	0.05	0.3	0.0089	0.0089	0.0128	0.0092	0.0171	0.0089	0.1035
5	0.1	0.1	0.0399	0.0402	0.0906	0.0476	0.2573	0.3616	0.3027
5	0.1	0.3	0.018	0.018	0.0203	0.0181	0.0334	0.018	0.1039
10	0.05	0.1	0.7321	0.0476	0.2853	0.1742	0.6683	1.1706	0.8913
10	0.05	0.3	0.0094	0.0093	0.0156	0.0098	0.0202	0.0093	0.1267
10	0.1	0.1	0.7726	0.091	0.3515	0.1942	0.6349	0.8075	0.8429
10	0.1	0.3	0.0193	0.0193	0.0233	0.0195	0.0428	0.0193	0.1679
m=1000, n=500									
5	0.1	0.05	0.031	0.0311	0.0493	0.039	0.1436	0.0314	0.2074
5	0.1	0.1	0.0188	0.0188	0.023	0.0197	0.0484	0.0188	0.1352
5	0.3	0.05	0.1723	0.0947	0.1503	0.1216	0.3023	0.0943	0.2946
5	0.3	0.1	0.0994	0.0582	0.0894	0.057	0.1061	0.0566	0.1333
10	0.1	0.05	0.8952	0.0424	0.4071	0.0795	0.5504	0.0475	0.6989
10	0.1	0.1	0.0207	0.0207	0.0273	0.0222	0.0684	0.0208	0.1612
10	0.3	0.05	0.8675	0.1231	0.5028	0.1544	0.5626	0.1258	0.7173
10	0.3	0.1	0.1139	0.0623	0.1013	0.063	0.1563	0.0622	0.2003

converge. In the larger case, the difference is much more apparent. GenASD still converges in less than half of a second, where as solving the subproblems to optimality takes about 17 seconds.

We compare our algorithm to three other common matrix completion algorithms in Table 3. The algorithm presented by Yao et al. [30], Fast Nonconvex Low-Rank Matrix Learning (FaNCL), is the only other work we know of that solves (1) with iterations having computational complexity $\mathcal{O}(r|\Omega|)$. The authors utilize nonconvex regularizers similar to the ones discussed in this paper, and use singular value thresholding with iteratively reweighted thresholds. The FaNCL algorithm was later improved upon in [32] by incorporating a momentum term for faster convergence. We only compare to the earlier work as that was the code we had available.

We also compare to FPC, which solved the nuclear norm minimization problem [19], and LMaFit, which solves the rank constrained problem [27]. Because LMaFit requires an estimate of the rank, we show results when the algorithm is given the correct rank and a rank twice as large as the original matrix to demonstrate the advantage of a rank minimization approach.

With minor exceptions, the algorithm presented in this paper, FaNCL, and LMaFit when given the correct rank all give approximately the same quality result. GenAltMin solves the problem faster than FaNCL in every case. Although GenAltMin and FaNCL take approximately the same amount of time per iteration, singular value thresholding

methods take significantly more iterations. Our algorithm outperforms FPC for reasons discussed earlier in this section, and also LMaFit when the rank is not well known.

4.2 Collaborative filtering

Perhaps the most widely known application of rank minimization is the Netflix Problem, wherein the goal is to predict how a user would rate a movie based on how she rated other movies, along with how other users with similar taste rated said movie. To formulate this as a matrix completion problem, we have a sparse matrix whose columns correspond to different movies and whose rows correspond to different users, with the entries of the matrix being how a user rated a specific movie. We expect that if every entry of this matrix was observed, the matrix would be low rank because the number of factors contributing to how much someone enjoys a movie is far less than the total number of movies or users in the data set.

We utilize Algorithm 4.2 and LMaFit on the MovieLens100k and MovieLens1m datasets [1], and the Jester dataset [9]. Both MovieLens datasets consist of ratings on various movies, rated from 1 to 5, and the Jester dataset consists of ratings on jokes, rated -10 to 10. The MovieLens100k dataset has 1000 users, 1700 movies, and 100,000 measurements, the MovieLens1m dataset has 6000 users, 4000 movies, and 1 million measurements, and the Jester dataset has 24,983 users, 101 jokes, and 689,000 measurements. Note that while the movie lens datasets are both very sparse (approximately 5%), the Jester dataset has 27% of all possible ratings.

For each dataset, we separate the data into five partitions, and for each partition we use the remaining four partitions to find a low rank matrix, and the fifth partition to test our results. In Table 4, we report the normalized mean absolute error (NMAE), defined as

$$\text{NMAE} = \frac{1}{n_{\text{ratings}}} \sum_i \frac{|y_i - \tilde{y}_i|}{y_{\max} - y_{\min}}$$

Table 4 NMAE utilizing Algorithm 2 with the trace inverse regularizer and with the nuclear norm regularizer, along with LMaFit

	MovieLens100k			MovieLens1m			Jester		
Fold	TI	NN	LmaFit	TI	NN	LmaFit	TI	NN	LmaFit
1	**0.1724**	0.1812	0.1800	**0.1683**	0.1695	0.1820	**0.1570**	0.1607	0.1600
2	**0.1719**	0.1799	0.1775	**0.1676**	0.1699	0.1811	**0.1577**	0.1610	0.1601
3	**0.1702**	0.1785	0.1781	**0.1682**	0.1695	0.1825	**0.1572**	0.1604	0.1596
4	**0.1715**	0.1789	0.1787	**0.1685**	0.1703	0.1824	**0.1572**	0.1603	0.1602
5	**0.1732**	0.1822	0.1788	**0.1678**	0.1691	0.1815	**0.1574**	0.1612	0.1601
Avg	**0.1719**	0.1802	0.1786	**0.1681**	0.1697	0.1819	**0.1573**	0.1607	0.1600

The bold values indicate the best performance across all methods tested and are significant

where n_{ratings} is the total number of ratings used in the testing set, y is the measurements from the dataset, $\tilde{y}$ are the predictions from the low rank matrix, and $y_{\max}$ and $y_{\min}$ are the maximum and minimum ratings for the dataset (5 and 1 for the MovieLens dataset, and -10 and 10 for the Jester dataset). In each case, we use 10 as the upper bound on the rank. We found that the NMAE for LMaFit is minimized when constrained to a rank 1 matrix, which is what is reported.

In every fold in each of the three datasets, Algorithm 4.2 utilizing the trace norm regularizer outperforms the nuclear norm regularizer and LMaFit. To gain insight as to why the trace inverse regularizer outperforms the other methods, we examine the singular value distribution of the resulting low rank matrix. The singular values for the matrices recovered from the MovieLens1M dataset withholding fold 5 is shown for each method in Fig. 6. Comparing the trace inverse to the nuclear norm, the first singular value of the matrix recovered with the trace inverse regularizer is larger, and the rest are smaller, which is expected because the trace inverse puts more weight on minimizing smaller singular value and less weight on minimizing larger singular values. Because the ratings matrix is close to a rank one matrix, penalizing the largest singular value is disadvantageous because we expect it to be large. Additionally, as opposed to the result from LMaFit, the remaining 9 singular values are nonzero. This demonstrates the advantage of rank minimization methods over rank constrained methods: while we may want to put more emphasis on the first singular value, the remaining singular values are still important. In a rank constrained paradigm, there is no way to both keep singular values and also minimize them.

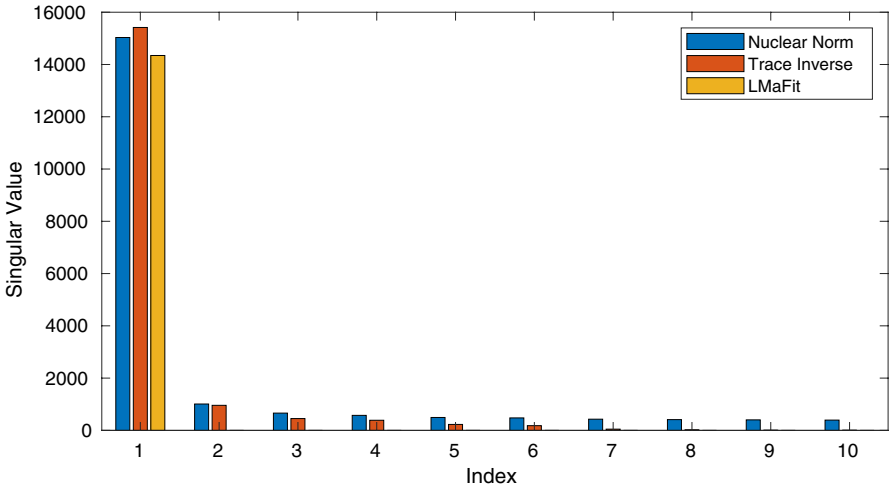


Fig. 6 Singular value decomposition for the matrix recovered from the MovieLens1M dataset withholding fold 5

5 Conclusions

We have shown that the problem of minimizing the rank of a matrix using nonconvex regularizers can be posed as a bi-convex semidefinite optimization problem. By doing so, we were able to derive efficient algorithms using a low rank factorization and show convergence.

The methods are shown to be computationally superior to methods based off of the nuclear norm relaxation, and that the estimator bias is drastically reduced by using nonconvex regularizers. We show that the quality of the result from our algorithm hardly changes when either of the parameters are changed by multiple orders of magnitude. Additionally, we show that our method is faster than other existing methods based off of nonconvex regularizers.

References

1. Movielens. https://grouplens.org/datasets/movielens/. Accessed: 2019-11-21
2. Andrew, A., Chu, K., Lancaster, P.: Derivatives of eigenvalues and eigenvectors of matrix functions. SIAM J. Matrix Anal. Appl. **14**(4), 903–926 (1993). https://doi.org/10.1137/0614061
3. Burer, S., Monteiro, R.: A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Math. Program. **95**(2), 329–357 (2003). https://doi.org/10.1007/s10107-002-0352-8
4. C. Lu J. Tang, S.Y., Lin, Z.: Generalized nonconvex nonsmooth low-rank minimization. Proceedings of the IEEE computer society conference on computer vision and pattern recognition (2014). https://doi.org/10.1109/CVPR.2014.526
5. Candès, E., Tao, T.: The power of convex relaxation: Near-optimal matrix completion. IEEE Trans. Inf. Theor. **56**(5), 2053–2080 (2010). https://doi.org/10.1109/TIT.2010.2044061
6. Fan, J., Li, R.: Variable selection via nonconcave penalized likelihood and its oracle properties. J. Am. Stat. Assoc. **96**(456), 1348–1360 (2001)
7. Fazel, M., Hindi, H., Boyd, S.P.: Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. Proceedings of the 2003 American Control Conference, 2003. **3**, 2156–2162 vol.3 (2003)
8. Geman, D.: Chengda Yang: nonlinear image recovery with half-quadratic regularization. IEEE Trans. Image Process. **4**(7), 932–946 (1995)
9. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. Inf. Retr. **4**(2), 133–151 (2001). https://doi.org/10.1023/A:1011419012209
10. Hoffman, A.J., Wielandt, H.W.: The variation of the spectrum of a normal matrix. Duke Math. J. **20**(1), 37–39 (1953). https://doi.org/10.1215/S0012-7094-53-02004-3
11. Cai, J., Candes, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
12. Lai, M.J., Xu, Y., Yin, W.: Improved iteratively reweighted least squares for unconstrained smoothed l_q minimization. SIAM J. Num. Anal. **51**(2), 927–957 (2013). https://doi.org/10.1137/110840364
13. Li, Q., Qi, Hd.: A sequential semismooth newton method for the nearest low-rank correlation matrix problem. SIAM J. Optim. **21**(4), 1641–1666 (2011)
14. Liu, Z., Vandenberghe, L.: Interior-point method for nuclear norm approximation with application to system identification. SIAM J. Matrix Anal. Appl. **31**(3), 1235–1256 (2010). https://doi.org/10.1137/090755436
15. Lu, C., Zhu, C., Xu, C., Yan, S., Lin, Z.: Generalized singular value thresholding. arXiv **abs/1412.2231** (2014). arXiv: 1412.2231
16. Magnus, J.: On differentiating eigenvalues and eigenvectors. Econo. Theory **1**(2), 179–191 (1985). https://doi.org/10.1017/s0266466600011129

17. Mohan, K., Fazel, M.: Iterative reweighted least squares for matrix rank minimization. 2010 48th Annual Allerton Conference on communication, control and computing (Allerton) (2010).https://doi.org/10.1109/allerton.2010.5706969
18. Rennie, J.D.M., Srebro, N.: Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd International Conference on machine learning, ICML'05, p. 713–719. Association for Computing Machinery, New York, NY, USA (2005). https://doi.org/10.1145/1102351.1102441
19. Ma, S., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. Math. Program. **128**, 321–353 (2009)
20. Sagan, A., Shen, X., Mitchell, J.E.: Two relaxation methods for rank minimization problems. J. Optim. Theory Appl. **186**(3), 806–825 (2020). https://doi.org/10.1007/s10957-020-01731-
21. Shen, X., Mitchell, J.: A penalty method for rank minimization problems in symmetric matrices. Comput. Optim. Appl. **71**(2), 353–380 (2018). https://doi.org/10.1007/s10589-018-0010-6
22. Srebro, N., Rennie, J.D.M., Jaakkola, T.S.: Maximum-margin matrix factorization. In: Proceedings of the 17th International Conference on neural information processing systems, NIPS'04, p. 1329–1336. MIT Press, Cambridge, MA, USA (2004)
23. Hastie, T., Mazumder, R., Lee, J.D., Zadeh, R.: Matrix completion and low-rank svd via fast alternating least squares. J. Mach. Learn. Res. **16**(104), 3367–3402 (2015)
24. Tanner, J., Wei, K.: Low rank matrix completion by alternating steepest descent methods. Appl. Comput. Harmonic Anal. (2015). https://doi.org/10.1016/j.acha.2015.08.003
25. Tasissa, A., Lai, R.: Exact reconstruction of euclidean distance geometry problem using low-rank matrix completion. IEEE Trans. Inform. Theory **65**(5), 3124–3144 (2019). https://doi.org/10.1109/tit.2018.2881749
26. Trzasko, J., Manduca, A.: Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0 -minimization. IEEE Trans. Med. Imag. **28**(1), 106–121 (2009)
27. Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Math. Program. Comput. **4**(4), 333–361 (2012). https://doi.org/10.1007/s12532-012-0044-1
28. Xu, Y., Yin, W.: A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. SIAM J. Imag. Sci. **6**(3), 1758–1789 (2013). https://doi.org/10.1137/120887795
29. Lou, Y., Yin, P., Xin, J.: Point source super-resolution via non-convex l_1 based methods. J. Sci. Comput. **68**(3), 1082–1100 (2016)
30. Yao, Q., Kwok, J., Zhong, W.: Fast low-rank matrix learning with nonconvex regularization. 2015 IEEE International conference on data mining (2015). https://doi.org/10.1109/icdm.2015.9
31. Yao, Q., Kwok, J.T., Gao, F., Chen, W., Liu, T.Y.: Efficient inexact proximal gradient algorithm for nonconvex problems. Proceedings of the Twenty-Sixth International Joint Conference on artificial intelligence (2017). https://doi.org/10.24963/ijcai.2017/462
32. Yao, Q., Kwok, J.T., Wang, T., Liu, T.: Large-scale low-rank matrix learning with nonconvex regularizers. IEEE Trans. Pattern Anal. Mach. Intell. **41**(11), 2628–2643 (2019). https://doi.org/10.1109/TPAMI.2018.2858249
33. Zhang, C.H.: Nearly unbiased variable selection under minimax concave penalty. Annal. Stat. **38**(2), 894–942 (2010)
34. Zhang, D., Hu, Y., Ye, J., Li, X., He, X.: Matrix completion by truncated nuclear norm regularization. 2012 IEEE Conference on computer vision and pattern recognition pp. 2192–2199 (2012). https://doi.org/10.1109/CVPR.2012.6247927