Iterative Reweighted Algorithms for Matrix Rank Minimization

Karthik Mohan KARNA@UW.EDU
Maryam Fazel MFAZEL@UW.EDU
Department of Electrical Engineering
University of Washington
Seattle, WA 98195-4322, USA

Editor:

Abstract

The problem of minimizing the rank of a matrix subject to affine constraints has many applications in machine learning, and is known to be NP-hard. One of the tractable relaxations proposed for this problem is nuclear norm (or trace norm) minimization of the matrix, which is guaranteed to find the minimum rank matrix under suitable assumptions.

In this paper, we propose a family of Iterative Reweighted Least Squares algorithms IRLS-p (with $0 \leq p \leq 1$), as a computationally efficient way to improve over the performance of nuclear norm minimization. The algorithms can be viewed as (locally) minimizing certain smooth approximations to the rank function. When $p = 1$, we give theoretical guarantees similar to those for nuclear norm minimization, i.e., recovery of low-rank matrices under certain assumptions on the operator defining the constraints. For $p < 1$, IRLS-p shows better empirical performance in terms of recovering low-rank matrices than nuclear norm minimization. We provide an efficient implementation for IRLS-p, and also present a related family of algorithms, sIRLS-p. These algorithms exhibit competitive run times and improved recovery when compared to existing algorithms for random instances of the matrix completion problem, as well as on the MovieLens movie recommendation data set.

Keywords: Matrix Rank Minimization, Matrix Completion, Iterative Algorithms, Null-space Property

1. Introduction

The *Affine Rank Minimization Problem* (ARMP), or the problem of finding the minimum rank matrix in an affine set, arises in many engineering applications such as collaborative filtering (e.g., Candes and Recht, 2009; Srebro et al., 2005), low-dimensional Euclidean embedding (e.g., Fazel et al., 2003), low-order model fitting and system identification (e.g., Liu and Vandenberghe, 2008), and quantum tomography (e.g., Gross et al., 2010). The problem is as follows,

$$\begin{array}{ll} \text{minimize} & \text{rank}(X) \\ \text{subject to} & \mathcal{A}(X) = b, \end{array} \tag{1}$$

where $X \in \mathbb{R}^{m\times n}$ is the optimization variable, $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^q$ is a linear map, and $b \in \mathbb{R}^q$ denotes the measurements. When X is restricted to be a diagonal matrix, ARMP reduces to the cardinality minimization or sparse vector recovery problem,

$$\begin{array}{ll} \text{minimize} & \text{card}(x) \\ \text{subject to} & Ax = b, \end{array}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ and $\text{card}(x)$ denotes the number of non-zeros entries of x. A commonly used convex heuristic for this problem is ℓ_1 minimization. The field of compressed sensing has shown that under certain conditions on the matrix A, this heuristic solves the original cardinality minimization problem (Candes and Tao, 2004). There are also geometric arguments in favor of ℓ_1 norm as a good convex relaxation, see (Chandrasekaran et al., 2010b) for a unifying analysis for general linear inverse problems. However, it has been observed empirically (e.g. Candes et al., 2008; Lobo et al., 2006) that by appropriately weighting the ℓ_1 norm and iteratively updating the weights, the recovery performance of the ℓ_1 algorithm is enhanced. It is also shown theoretically that for sparse recovery from noisy measurements, this algorithm has a better recovery error than ℓ_1 minimization under suitable assumptions on the matrix A (Needell, 2009; Zhang, 2010). The Reweighted ℓ_1 algorithm has also been generalized to the recovery of low-rank matrices (Fazel et al., 2003; Mohan and Fazel, 2010a).

Another simple and computationally efficient reweighted algorithm for sparse recovery, proposed in Daubechies et al. (Daubechies et al., 2010), is the Iterative Reweighted Least Squares algorithm (IRLS-p, for any $0 < p \leq 1$). Its kth iteration is given by

$$x^{k+1} \quad = \quad \underset{x}{\text{argmin}} \left\{ \sum_i w_i^k x_i^2 : Ax = b, \right\},$$

where $w^k \in \mathbb{R}^n$ is a weight vector with $w_i^k = (|x_i^k|^2 + \gamma)^{p/2-1}$ ($\gamma > 0$ being a regularization parameter added to ensure that w^k is well defined). See (Daubechies et al., 2010, section 1) for a history of earlier algorithms with a similar idea. For $p = 1$, (Daubechies et al., 2010) gave a theoretical guarantee for sparse recovery similar to ℓ_1 minimization. The starting point x^0 is set to zero, so the first iteration gives the least norm solution to $Ax = b$. It was empirically observed in (Chartrand and Staneva, 2008; Chartrand and Yin, 2008) that IRLS-p shows a better recovery performance than ℓ_1 minimization for $p < 1$ and has a similar performance as the reweighted ℓ_1 algorithm when p is set to zero. The computational benefits of IRLS-1, as well as the above mentioned theoretical and empirical improvements of IRLS-p, $p < 1$, over ℓ_1 minimization, motivate us to ask: *Would an iterative reweighted algorithm bring similar benefits for recovery of low rank matrices, and would an improved performance be at the price of additional computational cost relative to the standard nuclear norm minimization*?

1.1 Iterative Reweighted Least Squares for ARMP

Towards answering this question, we propose the iterative reweighted least squares algorithm for rank minimization, outlined below.

Data: $\mathcal{A}$, b
Set $k = 1$. Initialize $W_p^0 = I, \gamma^1 > 0$;
while *not converged* **do**
  $X^k = \underset{X}{\operatorname{argmin}} \{\operatorname{Tr}(W_p^{k-1} X^T X) : \mathcal{A}(X) = b\}$;
  $W_p^k = (X^{k^T} X^k + \gamma^k I)^{\frac{p}{2}-1}$;
  Choose $0 < \gamma^{k+1} \leq \gamma^k$;
  $k = k + 1$;
end

Algorithm 1: IRLS-p Algorithm for Matrix Rank Minimization with $0 \leq p \leq 1$

Each iteration of Algorithm 1 minimizes a weighted Frobenius norm of the matrix X, since $\operatorname{Tr}(W_p^{k-1} X^T X) = \|(W_p^{k-1})^{1/2} X\|_F^2$. While minimizing the Frobenius norm subject to affine constraints doesn't lead to low-rank solutions in general, through a careful reweighting of this norm we show that Algorithm 1 does indeed produce low-rank solutions under suitable assumptions. Usually a reweighted algorithm trades off computational time for improved recovery performance when compared to the unweighted convex heuristic. As an example, the reweighted ℓ_1 algorithm for sparse recovery (Candes et al., 2008) and the reweighted nuclear norm algorithm (Mohan and Fazel, 2010a) for matrix rank minimization solve the corresponding standard convex relaxations (ℓ_1 and nuclear norm minimization respectively) in their first iteration. Thus these algorithms take at least as much time as the corresponding convex algorithms. However, the iterates of the IRLS-p family of algorithms simply minimize a weighted Frobenius norm, and have run-times comparable with the nuclear norm heuristic, while (for $p < 1$) also enjoying improved recovery performance. In the $p = 1$ case, we show that the algorithm minimizes a certain smooth approximation to the nuclear norm, allowing for efficient implementations while having theoretical guarantees similar to nuclear norm minimization.

1.2 Contributions

Contributions of this paper are as follows.

- We give a convergence analysis for IRLS-p for all $0 \leq p \leq 1$. We also propose a related algorithm, sIRLS-p (or short IRLS), which can be seen as a first-order method for locally minimizing a smooth approximation to the rank function, and give convergence results for this algorithm as well. The results exploit the fact that these algorithms can be derived from the KKT conditions for minimization problems whose objectives are suitable smooth approximations to the rank function.

- We prove that the IRLS-1 algorithm is guaranteed to recover a low-rank matrix if the linear map $\mathcal{A}$ satisfies the following *Null space property* (NSP). We show this property is both necessary and sufficient for low-rank recovery. The specific NSP that we use first appeared in (Oymak and Hassibi, 2010; Oymak et al., 2011), and is expressed only in terms of the singular values of the matrices in the null space of $\mathcal{A}$.

  Definition 1.1 *Given $\tau > 0$, the linear map $\mathcal{A} : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^p$ satisfies the* τ-Null space Property *(τ-NSP) of order r if for every $Z \in \mathcal{N}(\mathcal{A})\backslash\{0\}$, we have*

$$\sum_{i=1}^{r} \sigma_i(Z) \quad < \quad \tau \sum_{i=r+1}^{n} \sigma_i(Z) \tag{2}$$

  where $\mathcal{N}(\mathcal{A})$ denotes the null space of $\mathcal{A}$ and $\sigma_i(Z)$ denotes the i^{th} largest singular value of Z. It is shown in (Oymak and Hassibi, 2010) that certain random Gaussian maps $\mathcal{A}$ satisfy this property with high probability.

- We give a gradient projection algorithm to implement IRLS-p. We extensively compare these algorithms with other state of the art algorithms on both 'easy' and 'hard' randomly picked instances of the matrix completion problem (these notions are made precise in the numerical section). We also present comparisons on the MovieLens movie recommendation dataset. Numerical experiments demonstrate that IRLS-0 and sIRLS-0 applied to the matrix completion problem have a better recovery performance than the Singular Value Thresholding algorithm, an implementation of Nuclear Norm Minimization (Cai et al., 2008), on both easy and hard instances. Importantly, in the case where there is no apriori information on the rank of the low rank solution (which is common in practice), our algorithm has a significantly better recovery performance on hard problem instances as compared to other state of the art algorithms for matrix completion including IHT, FPCA (Goldfarb and Ma, 2011), and Optspace (Keshavan and Oh, 2009).

1.3 Related Work

We review related algorithms for recovering sparse vectors and low rank matrices. Many approaches have been proposed for recovery of sparse vectors from linear measurements including ℓ_1 minimization, greedy algorithms (e.g. CoSaMP (Needell and Tropp, 2008), IHT (Goldfarb and Ma, 2011), and GraDes (Garg and Khandekar, 2009)). As mentioned earlier, reweighted algorithms including Iterative reweighted ℓ_1 (Candes et al., 2008) and Iterative Reweighted Least Squares (Rao and Kreutz-Delgado, 1999; Wipf and Nagarajan, 2010; Daubechies et al., 2010) with $0 < p \leq 1$ have been proposed to improve on the recovery performance of ℓ_1 minimization.

For the ARMP, analogous algorithms have been proposed including nuclear norm minimization (Fazel et al., 2001), reweighted nuclear norm minimization (Mohan and Fazel, 2010a; Fazel et al., 2003), as well as greedy algorithms such as AdMiRA (Lee and Bresler,

2010) which generalizes CoSaMP, SVP (Meka et al., 2010), a hard-thresholding algorithm that we also refer to as IHT, and Optspace (Keshavan and Oh, 2009).

Developing efficient implementations for nuclear norm minimization is an important research area since standard semidefinite programming solvers cannot handle large problem sizes. Towards this end, algorithms including SVT (Cai et al., 2008), NNLS (Toh and Yun, 2010), FPCA (Goldfarb and Ma, 2011) have been proposed. A spectral regularization algorithm (Mazumder et al., 2010) has also been proposed for the specific problem of matrix completion.

In a preliminary conference version of this paper (Mohan and Fazel, 2010b), we proposed the IRLS-p family of algorithms for ARMP, analogous to IRLS for sparse vector recovery (Daubechies et al., 2010). The present paper gives a new and improved theoretical analysis for IRLS-1 via a simpler NSP condition, obtains complete convergence results for all $0 \leq p \leq 1$, and gives more extensive numerical experiments.

Independent of us and at around the same time as the publication of our conference paper, Fornasier et al. (Fornasier et al., 2010) proposed the IRLSM algorithm, which is similar to our IRLS-1 algorithm but with a different weight update (a thresholding operation is used to ensure the weight matrix is invertible). The authors employ the Woodbury matrix inversion lemma to speed up the implementation of IRLSM and compare it to two other algorithms, Optspace and FPCA, showing that IRLSM has a lower relative error and comparable computational times. However, their analysis of low-rank recovery uses a Strong Null Space Property (SRNSP) which is equivalent to the condition in (Mohan and Fazel, 2010b) and is less general than the condition we consider in this paper, as discussed in Section 3. Also, the authors have only considered recovery of matrices of size 500×500 in their experiments. We present extensive comparisons of our algorithms with algorithms including Optspace, FPCA, and IHT for recovery of matrices of different sizes. We observe that our IRLS and sIRLS implementations run faster than IRLSM in practice. IRLSM solves a quadratic program in each iteration using a sequence of inversions, which can be expensive for large problems, even after exploiting the matrix inversion lemma. Finally, (Fornasier et al., 2010) only considers the $p = 1$ case.

The rest of the paper is organized as follows. We introduce the IRLS-p algorithm for ARMP in Section 2 and give convergence and performance guarantees in Section 3. In Section 4, we discuss an implementation for IRLS-p, tailored for the *matrix completion problem*, and in Section 5 we present the related algorithm sIRLS-p. Numerical experiments for IRLS-0 and sIRLS-0 for the matrix completion problem, as well as comparisons with SVT, IHT, IRLSM, Optspace and FPCA are given in Section 6. The last section summarizes the paper along with future research directions.

Notation. Let $\mathcal{N}(\mathcal{A})$ denote the null space of the operator $\mathcal{A}$ and $\text{Ran}(\mathcal{A}^*)$ denote the range space of the adjoint of $\mathcal{A}$. Let $\sigma(X)$ denote the vector of decreasingly ordered singular values of X so that $\sigma_i(X)$ denotes the i^{th} largest singular value of X. Also let $\|X\|$ denote

the spectral norm of X while its nuclear norm is defined as $\|X\|_* = \sum_i \sigma_i(X)$. I_k denotes the identity matrix of size $k \times k$.

2. Iterative Reweighted Least Squares (IRLS-p)

In this section, we describe the IRLS-p family of algorithms (Algorithm 1). Recall that replacing the rank function in (1) by $\|X\|_*$ yields the nuclear norm heuristic,

$$\begin{array}{ll} \text{minimize} & \|X\|_* \\ \text{subject to} & \mathcal{A}(X) = b. \end{array}$$

We consider other (convex and non-convex) smooth approximations to the rank function. Define the *smooth Schatten-p function* as

$$\begin{aligned} f_p(X) &= \mathrm{Tr}(X^T X + \gamma I)^{p/2} \\ &= \textstyle\sum_{i=1}^n (\sigma_i^2(X) + \gamma)^{\frac{p}{2}}. \end{aligned}$$

Note that $f_p(X)$ is differentiable for $p > 0$ and convex for $p \geq 1$. With $\gamma = 0$, $f_1(X) = \|X\|_*$, which is also known as the Schatten-1 norm. With $\gamma = 0$ and $p \to 0$, $f_p(X) \to \mathrm{rank}(X)$. Thus, it is of interest to consider the problem

$$\begin{array}{ll} \text{minimize} & f_p(X) \\ \text{subject to} & \mathcal{A}(X) = b, \end{array} \tag{3}$$

the optimality conditions of which motivate the IRLS-p algorithms for $0 \leq p \leq 1$. We show that IRLS-1 solves the smooth Schatten-1 norm or nuclear norm minimization problem, i.e., finds a globally optimal solution to (3) with $p = 1$. For $p < 1$, we show that IRLS-p finds a stationary point of (3).

We now give an intuitive way to derive the IRLS-p algorithm from the KKT conditions of (3). The Lagrangian corresponding to (3) is

$$L(X, \lambda) = f_p(X) + \langle \widehat{\lambda},\ \mathcal{A}(X) - b \rangle,$$

and the KKT conditions are $\nabla_X L(X, \widehat{\lambda}) = 0$, $\mathcal{A}(X) = b$. Note that $\nabla f_p(X) = pX(X^T X + \gamma I)^{p/2-1}$ (see e.g. Lewis, 1996). Letting $\lambda = \frac{\widehat{\lambda}}{p}$, we have that the KKT conditions for (3) are given by

$$\begin{aligned} 2X(X^T X + \gamma I)^{p/2-1} + \mathcal{A}^*(\lambda) &= 0 \\ \mathcal{A}(X) &= b. \end{aligned} \tag{4}$$

Let $W_p^k = ({X^k}^T X^k + \gamma I)^{p/2-1}$. The first condition in (4) can be written as $X = -\frac{1}{2}\mathcal{A}^*(\lambda)(X^T X + \gamma I)^{1-p/2}$. This is a fixed point equation, and a solution can be obtained by iteratively solving for X as $X^{k+1} = \frac{1}{2}\mathcal{A}^*(\lambda)(W_p^k)^{-1}$, along with the condition $\mathcal{A}(X^{k+1}) = b$. Note that X^{k+1} and the dual variable λ satisfy the KKT conditions for the convex optimization problem,

$$\begin{array}{ll} \text{minimize} & \mathrm{Tr}\, W_p^k X^T X \\ \text{subject to} & \mathcal{A}(X) = b. \end{array} \tag{5}$$

This idea leads to the IRLS-p algorithm described in Algorithm 1. Note that we also let $p = 0$ in Algorithm 1; to derive IRLS-0, we define another non-convex surrogate function by taking limits over $f_p(X)$. For any positive scalar x, it holds that $\lim_{p\to 0} \frac{1}{p}(x^p - 1) = \log x$. Therefore,

$$\begin{aligned} \lim_{p\to 0} \frac{f_p(X) - n}{p} &= \tfrac{1}{2} \log\det(X^T X + \gamma I) \\ &= \textstyle\sum_i \tfrac{1}{2} \log(\sigma_i^2(X) + \gamma). \end{aligned}$$

Thus IRLS-0 can be seen as iteratively solving (as outlined previously) the KKT conditions for the non-convex problem,

$$\begin{array}{ll} \text{minimize} & \log\det(X^T X + \gamma I) \\ \text{subject to} & \mathcal{A}(X) = b. \end{array} \tag{6}$$

Another way to derive the IRLS-p algorithm uses an alternative characterization of the Smooth Schatten-p function; see appendix A.

3. IRLS-p: Theoretical results

In this section, convergence properties for the IRLS-p family of algorithms are studied. We also give a matrix recovery guarantee for IRLS-1, under suitable assumptions on the null space of the linear operator $\mathcal{A}$.

3.1 Convergence of IRLS-p

We show that the difference between successive iterates of the IRLS-p ($0 \le p \le 1$) algorithm converges to zero and that every cluster point of the iterates is a stationary point of (3). These results generalize the convergence results given for IRLS-1 in (Mohan and Fazel, 2010b; Fornasier et al., 2010) to IRLS-p with $0 < p \le 1$. In this section, we drop the subscript on W_p^k for ease of notation. Our convergence analysis relies on useful auxiliary functions defined as

$$\mathcal{J}^p(X, W, \gamma) := \begin{cases} \frac{p}{2}(\mathrm{Tr}(W(X^T X + \gamma I)) + \frac{2-p}{p}\mathrm{Tr}((W)^{\frac{p}{p-2}})) & \text{if } 0 < p \le 1 \\ \mathrm{Tr}(W(X^T X + \gamma I)) - \log\det W - n & \text{if } p = 0. \end{cases}$$

These functions can be obtained from the alternative characterization of Smooth Schatten-p function with details in Appendix A. We can express the iterates of IRLP-p as

$$\begin{aligned} X^{k+1} &= \operatorname*{argmin}_{X:\mathcal{A}(X)=b} \mathcal{J}^p(X, W^k, \gamma^k) \\ W^{k+1} &= \operatorname*{argmin}_{W\succ 0} \mathcal{J}^p(X^{k+1}, W, \gamma^{k+1}), \end{aligned}$$

and it follows that

$$\begin{aligned} \mathcal{J}^p(X^{k+1}, W^{k+1}, \gamma^{k+1}) &\le \mathcal{J}^p(X^{k+1}, W^k, \gamma^{k+1}) \\ &\le \mathcal{J}^p(X^{k+1}, W^k, \gamma^k) \\ &\le \mathcal{J}^p(X^k, W^k, \gamma^k). \end{aligned} \tag{7}$$

The following theorem shows that the difference between successive iterates converges to zero. The proof is given in Appendix B.

Theorem 3.1 *Given any $b \in \mathbb{R}^q$, the iterates $\{X^k\}$ of IRLS-p ($0 < p \leq 1$) satisfy*

$$\sum_{k=1}^{\infty} \|X^{k+1} - X^k\|_F^2 \leq 2D^{\frac{2}{p}},$$

where $D := \mathcal{J}^p(X^1, W^0, \gamma^0)$. In particular, we have that $\lim_{k\to\infty} \left(X^k - X^{k+1}\right) = 0$.

Theorem 3.2 *Let $\gamma_{\min} := \lim_{k\to\infty} \gamma^k > 0$. Then the sequence of iterates $\{X^k\}$ of IRLS-p ($0 \leq p \leq 1$) is bounded, and every cluster point of the sequence is a stationary point of (3) (when $0 < p \leq 1$), or a stationary point of (6) (when $p = 0$).*

Proof Let

$$g(X, \gamma) = \begin{cases} \operatorname{Tr}(X^T X + \gamma I)^{\frac{p}{2}} & \text{if } 0 < p \leq 1 \\ \log\det(X^T X + \gamma I) & \text{if } p = 0 \end{cases}$$

Then $g(X^k, \gamma^k) = \mathcal{J}^p(X^k, W^k, \gamma^k)$ and it follows from (7) that $g(X^{k+1}, \gamma^{k+1}) \leq g(X^k, \gamma^k)$ for all $k \geq 1$. Hence the sequence $\{g(X^k, \gamma^k)\}$ converges. This fact together with $\gamma_{\min} > 0$ implies that the sequence $\{X^k\}$ is bounded.

We now show that every cluster point of $\{X^k\}$ is a stationary point of (3). Suppose to the contrary and let $\widetilde{X}$ be a cluster point of $\{X^k\}$ that is not a stationary point. By the definition of cluster point, there exists a subsequence $\{X^{n_i}\}$ of $\{X^k\}$ converging to $\widetilde{X}$. By passing to a further subsequence if necessary, we can assume that $\{X^{n_i+1}\}$ is also convergent and we denote its limit by $\widehat{X}$. By definition, X^{n_i+1} is the minimizer of

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} W^{n_i} X^T X \\ \text{subject to} & \mathcal{A}(X) = b. \end{array} \tag{8}$$

Thus, X^{n_i+1} satisfies the KKT conditions of (8), i.e.,

$$X^{n_i+1} W^{n_i} \in \operatorname{Ran}(\mathcal{A}^*) \text{ and } \mathcal{A}(X^{n_i+1}) = b,$$

where $\operatorname{Ran}(\mathcal{A}^*)$ denotes the range space of $\mathcal{A}^*$. Passing to limits, we see that

$$\widehat{X}\widetilde{W} \in \operatorname{Ran}(\mathcal{A}^*) \text{ and } \mathcal{A}(\widehat{X}) = b, \tag{9}$$

where $\widetilde{W} = (\widetilde{X}^T\widetilde{X} + \gamma_{\min} I)^{-1}$. From (9), we conclude that $\widehat{X}$ is a minimizer of the following convex optimization problem,

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr} \widetilde{W} X^T X \\ \text{subject to} & \mathcal{A}(X) = b. \end{array} \tag{10}$$

Next, by assumption, $\widetilde{X}$ is not a stationary point of (3) (for $0 < p \leq 1$) nor (6) (for $p = 0$). This implies that $\widetilde{X}$ is not a minimizer of (10) and thus $\operatorname{Tr} \widetilde{W}\widehat{X}^T\widehat{X} < \operatorname{Tr} \widetilde{W}\widetilde{X}^T\widetilde{X}$. This is equivalent to $\mathcal{J}^p(\widehat{X}, \widetilde{W}, \gamma_{\min}) < \mathcal{J}^p(\widetilde{X}, \widetilde{W}, \gamma_{\min})$. From this last relation and (7) it follows that,

$$\begin{array}{rcl} \mathcal{J}^p(\widehat{X}, \widehat{W}, \gamma_{\min}) & < & \mathcal{J}^p(\widetilde{X}, \widetilde{W}, \gamma_{\min}), \\ g(\widehat{X}, \gamma_{\min}) & < & g(\widetilde{X}, \gamma_{\min}). \end{array} \tag{11}$$

On the other hand, since the sequence $\{g(X^k, \gamma^k)\}$ converges, we have that

$$\lim g(X^i, \gamma^i) = \lim g(X^{n_i}, \gamma^{n_i}) = g(\widetilde{X}, \gamma_{\min}) = \lim g(X^{n_{i+1}}, \gamma^{n_{i+1}}) = g(\widehat{X}, \gamma_{\min})$$

which contradicts (11). Hence, every cluster point of $\{X^k\}$ is a stationary point of (3) (when $0 < p \leq 1$) and a stationary point of (6) (when $p = 0$). ■

3.2 Performance guarantee for IRLS-1

In this section, we discuss necessary and sufficient conditions for low-rank recovery using IRLS-1. We show that any low-rank matrix satisfying $\mathcal{A}(X) = b$ can be recovered via IRLS-1, if the null space of $\mathcal{A}$ satisfies a certain property. If the desired matrix is not low-rank, we show IRLS-1 recovers it to within an error that is a constant times the best rank-r approximation error, for any r. We first give a few definitions and lemmas.

Definition 3.3 *Given $x \in \mathbb{R}^n$, let $x_{[i]}$ denote the i^{th} largest element of x so that $x_{[1]} \geq x_{[2]} \geq \ldots x_{[n-1]} \geq x_{[n]}$. A vector $x \in \mathbb{R}^n$ is said to be* majorized *by $y \in \mathbb{R}^n$ (denoted as $x \prec y$) if $\sum_{i=1}^k x_{[i]} \leq \sum_{i=1}^k y_{[i]}$ for $k = 1, 2, \ldots, n-1$ and $\sum_{i=1}^n x_{[i]} = \sum_{i=1}^n y_{[i]}$. A vector $x \in \mathbb{R}^n$ is said to be* weakly sub-majorized *by $y \in \mathbb{R}^n$ (denoted as $x \prec_w y$) if $\sum_{i=1}^k x_{[i]} \leq \sum_{i=1}^k y_{[i]}$ for $k = 1, 2, \ldots, n$.*

Lemma 3.4 ((Horn and Johnson, 1990)) *For any two matrices, $A, B \in \mathbb{R}^{m \times n}$ it holds that $|\sigma(A) - \sigma(B)| \prec_w \sigma(A - B)$.*

Lemma 3.5 ((Horn and Johnson, 1991; Marshall and Olkin, 1979)) *Let $g : D \to \mathbb{R}$ be a convex and increasing function where $D \subset \mathbb{R}$. Let $x, y \in D^n$. Then if $x \prec_w y$, we have $(g(x_1), g(x_2), \ldots, g(x_n)) \prec_w (g(y_1), g(y_2), \ldots, g(y_n))$.*

Since $g(x) = (x^2 + \gamma)^{\frac{1}{2}}$ is a convex and increasing function on $\mathbb{R}_+$, applying Lemma 3.5 to the majorizaiton inequality in Lemma 3.4 we have

$$\sum_{i=1}^n (|\sigma_i(A) - \sigma_i(B)|^2 + \gamma)^{\frac{1}{2}} \leq \sum_{i=1}^n (\sigma_i^2(A - B) + \gamma)^{\frac{1}{2}}. \tag{12}$$

Let X_0 be a matrix (that is not necessarily low-rank) and let the measurements be given by $b = \mathcal{A}(X_0)$. In this section, we give necessary and sufficient conditions for recovering X_0

using IRLS-1. Let $X_0^\gamma := (X_0^T X_0 + \gamma I)^{\frac{1}{2}}$ be the γ-approximation of X_0, and let $X_{0,r}$, $X_{0,r}^\gamma$ be the best rank-r approximations of X_0, X_0^γ respectively.

Recall the Null space Property τ-NSP defined earlier in (2). As a rough intuition, in the context of recovering matrices of rank r this condition requires that every nonzero matrix in the null space of $\mathcal{A}$ has a rank larger than $2r$.

Theorem 3.6 *Assume that $\mathcal{A}$ satisfies τ-NSP of order r for some $0 < \tau < 1$. For every X_0 satisfying $\mathcal{A}(X_0) = b$ it holds that*

$$f_1(X_0 + Z) > f_1(X_0), \quad \textit{for all} \quad Z \in \mathcal{N}(\mathcal{A}) \backslash \{0\} \quad \textit{satisfying} \quad \|Z\|_* \geq C\|X_0^\gamma - X_{0,r}^\gamma\|_*. \quad (13)$$

Furthermore, we have the following bounds,

$$\begin{aligned} \|\bar{X} - X_0\|_* &\leq C\|X_0^\gamma - X_{0,r}^\gamma\|_* \\ \|\bar{X}_r - X_0\|_* &\leq (2C+1)\|X_0^\gamma - X_{0,r}^\gamma\|_*, \end{aligned}$$

where $C = 2\frac{(1+\tau)}{1-\tau}$, and $\bar{X}$ is the output of IRLS-1. Conversely, if (13) holds, then $\mathcal{A}$ satisfies δ-NSP of order r, where $\delta > \tau$.

Proof Let $Z \in \mathcal{N}(A)\backslash\{0\}$ and $\|Z\|_* \geq C\|X_0^\gamma - X_{0,r}^\gamma\|_*$. We see that

$$\begin{aligned} f_1(X_0+Z) &= \mathrm{Tr}((X_0+Z)^T(X_0+Z)+\gamma I)^{\frac{1}{2}} = \sum_{i=1}^{n}(\sigma_i^2(X_0+Z)+\gamma)^{\frac{1}{2}} \\ &\geq \sum_{i=1}^{r}((\sigma_i(X_0)-\sigma_i(Z))^2+\gamma)^{\frac{1}{2}} + \sum_{i=r+1}^{n}((\sigma_i(Z)-\sigma_i(X_0))^2+\gamma)^{\frac{1}{2}} \\ &= \sum_{i=1}^{r}((\sigma_i^2(X_0)+\gamma)+\sigma_i^2(Z)-2\sigma_i(X_0)\sigma_i(Z))^{\frac{1}{2}} \\ &\quad + \sum_{i=r+1}^{n}((\sigma_i^2(Z)+\gamma)+\sigma_i^2(X_0)-2\sigma_i(Z)\sigma_i(X_0))^{\frac{1}{2}} \\ &\geq \sum_{i=1}^{r}|(\sigma_i^2(X_0)+\gamma)^{\frac{1}{2}}-\sigma_i(Z)| + \sum_{i=r+1}^{n}|(\sigma_i^2(Z)+\gamma)^{\frac{1}{2}}-\sigma_i(X_0)| \\ &\geq \sum_{i=1}^{r}(\sigma_i^2(X_0)+\gamma)^{\frac{1}{2}} - \sum_{i=1}^{r}\sigma_i(Z) + \sum_{i=r+1}^{n}(\sigma_i^2(Z)+\gamma)^{\frac{1}{2}} - \sum_{i=r+1}^{n}\sigma_i(X_0) \\ &\geq f^1(X_0) - \sum_{i=r+1}^{n}(\sigma_i^2(X_0)+\gamma)^{\frac{1}{2}} - \sum_{i=1}^{r}\sigma_i(Z) + \sum_{i=r+1}^{n}\sigma_i(Z) - \sum_{i=r+1}^{n}\sigma_i(X_0), \end{aligned}$$

where the first inequality follows from (12). Since τ-NSP holds, we further have that

$$\begin{aligned}\sum_{i=1}^{r}\sigma_i(Z) < \tau\sum_{i=r+1}^{n}\sigma_i(Z) &= \sum_{i=r+1}^{n}\sigma_i(Z) - (1-\tau)\sum_{i=r+1}^{n}\sigma_i(Z)\\ &\le \sum_{i=r+1}^{n}\sigma_i(Z) - C\frac{1-\tau}{1+\tau}\|X_0^\gamma - X_{0,r}^\gamma\|_*\\ &= \sum_{i=r+1}^{n}\sigma_i(Z) - 2\|X_0^\gamma - X_{0,r}^\gamma\|_* \qquad (14)\end{aligned}$$

where the second inequality uses $\|Z\|_* \ge C\|X_0^\gamma - X_{0,r}^\gamma\|_*$ and τ-NSP. Combining (14) and (14), we obtain

$$f_1(X_0+Z) \ge f_1(X_0) + \sum_{i=r+1}^{n}\sigma_i(Z) - \sum_{i=1}^{r}\sigma_i(Z) - 2\|X_0^\gamma - X_{0,r}^\gamma\|_* > f_1(X_0). \qquad (15)$$

This proves (13).

Next, if $\widehat{X}$ is an optimal solution of (3) with $p=1$, then $Z = \widehat{X} - X_0 \in \mathcal{N}(\mathcal{A})$ and it follows immediately from (15) that

$$\|\widehat{X} - X_0\|_* = \|Z\|_* \le C\|X_0^\gamma - X_{0,r}^\gamma\|_*.$$

Note that problem (3) is convex when $p=1$, and every stationary point of (3) is a global minimum. Hence, by Theorem 3.2, IRLS-1 converges to a global minimum of the problem (3). It follows that $\bar{X} = \widehat{X}$ and

$$\|\bar{X} - X_0\|_* \le C\|X_0^\gamma - X_{0,r}^\gamma\|_*. \qquad (16)$$

Finally, since $|\sigma(\bar{X}) - \sigma(X_0)| \prec_w \sigma(\bar{X} - X_0)$, we have that

$$\sum_{i=r+1}^{n}\sigma_i(\bar{X}) - \sum_{i=r+1}^{n}\sigma_i(X_0) \le \sum_{i=1}^{n}|\sigma_i(\bar{X}) - \sigma_i(X_0)| \le \|\bar{X} - X_0\|_*. \qquad (17)$$

Thus,

$$\begin{aligned}\|\bar{X}_r - X_0\|_* &\le \|\bar{X} - \bar{X}_r\|_* + \|\bar{X} - X_0\|_*\\ &\le \|X_0 - X_{0r}\|_* + \|\bar{X} - X_0\|_* + \|\bar{X} - X_0\|_*\\ &\le (2C+1)\|X_0^\gamma - X_{0,r}^\gamma\|_*\end{aligned}$$

where the second inequality follows from (17) and the third inequality from (16).

Conversely, suppose that (13) holds, i.e., $f_1(X_0+Z) > f_1(X_0)$ for all $\|Z\|_* \ge C\|X_0^\gamma - X_{0,r}^\gamma\|_*$, $Z \in \mathcal{N}(A)\backslash\{0\}$. We would like to show that $\mathcal{A}$ satisfies δ-NSP of order r. Assume to the contrary that there exists $Z \in \mathcal{N}(A)$ such that

$$\sum_{i=1}^{r}\sigma_i(Z) \ge \delta\sum_{i=r+1}^{n}\sigma_i(Z). \qquad (18)$$

Let $\alpha = \frac{1-\tau\delta}{2(1+\tau)}$ and set $X_0 = -Z_r - \alpha(Z - Z_r)$. Note that $\alpha < (1+\delta)/C$. Assume that Z satisfies

$$\left(\frac{1+\delta}{C} - \alpha\right) \sum_{i=r+1}^{n} \sigma_i(Z) \geq (n-r)\sqrt{\gamma}. \tag{19}$$

If not, Z can be multiplied by a large enough positive constant so that it satisfies both (19) and (18).

$$\|Z\|_* \geq (1+\delta) \sum_{i=r+1}^{n} \sigma_i(Z) \tag{20}$$

Combining (19) and (20), we obtain that

$$\begin{aligned} \|X_0^\gamma - X_{0,r}^\gamma\|_* = \sum_{i=r+1}^{n} (\alpha^2\sigma_i^2(Z) + \gamma)^{\frac{1}{2}} &\leq \sum_{i=r+1}^{n} (n-r)\sqrt{\gamma} + \alpha \sum_{i=r+1}^{n} \sigma_i(Z) \\ &\leq \frac{\|Z\|_*}{C} \end{aligned} \tag{21}$$

Moreover, it follows from (18) and the definition of X_0 that

$$\begin{array}{rcl} f_1(X_0) & = & \sum_{i=1}^{n} (\sigma_i^2 + \gamma)^{\frac{1}{2}} \\ & \geq & \sum_{i=1}^{r} \sigma_i(X_0) + \sum_{i=r+1}^{n} \sigma_i(X_0) \\ & = & \sum_{i=1}^{r} \sigma_i(Z) + \alpha \sum_{i=r+1}^{n} \sigma_i(Z) \\ & \geq & (\alpha+\delta) \sum_{i=r+1}^{n} \sigma_i(Z). \end{array} \tag{22}$$

On the other hand, notice also by the definition of α that $\alpha > \frac{1-\delta}{2}$. Also assume Z satisfies

$$(2\alpha + \delta - 1) \sum_{i=r+1}^{n} \sigma_i(Z) \geq n\sqrt{r}. \tag{23}$$

If not, Z can be multiplied by a large enough positive constant to satisfy (23) and also (18). Combining (23) and (22), we obtain further that

$$\begin{aligned} f_1(X_0 + Z) = \sum_{i=1}^{n} (\sigma_i^2(X_0 + Z) + \gamma)^{\frac{1}{2}} &= r\sqrt{\gamma} + \sum_{i=r+1}^{n} ((1-\alpha)^2\sigma_i^2(Z) + \gamma)^{\frac{1}{2}} \\ &\leq n\sqrt{\gamma} + (1-\alpha) \sum_{i=r+1}^{n} \sigma_i(Z) \leq f^1(X_0). \end{aligned} \tag{24}$$

Now it is easy to see that (21) and (24) together contradicts (13), which completes the proof. ■

Thus when the sufficient condition (τ-NSP of order r) holds, we have shown that the best rank-r approximation of the IRLS-1 solution is not far away from $\widetilde{X}$, the solution we wish to recover, and the distance between the two is bounded by a (γ-approximate) rank-r

approximation error of X_0. It has been shown in (Oymak and Hassibi, 2010) that 1-NSP of order r holds for a random Gaussian map $\mathcal{A}$ with high probability when the number of measurements is large enough. The necessity statement can be rephrased as δ-NSP with $\delta > \tau$ being a necessary condition for the following to hold: whenever there is an X such that $f_1(X) \leq f_1(X_0)$, we have that $\|X - X_0\|_* \leq C\|X_0^\gamma - X_{0,r}^\gamma\|_*$. Note that the necessary and sufficient conditions for recovery of approximately low-rank matrices using IRLS-1 mirror the conditions for recovery of approximately low-rank matrices using nuclear norm minimization which is that $\mathcal{A}$ satisfy 1-NSP.

Note that the null space property SRNSP of order r, considered in (Fornasier et al., 2010) and shown to be sufficient for low-rank recovery using IRLSM, is equivalent to 1-NSP of order $2r$. In this paper, we show that τ-NSP of order r (as contrasted with the order $2r$ NSP) is both necessary and sufficient for recovery of approximately low-rank matrices using IRLS-1. Thus, our condition for the recovery of approximately low-rank matrices using IRLS-1 generalizes those stated in (Mohan and Fazel, 2010b; Fornasier et al., 2010), since it places a weaker requirement on the linear map $\mathcal{A}$.

4. A Gradient projection based implementation of IRLS

In this section, we describe IRLS-GP, a gradient projection based implementation of IRLS-p for the Matrix Completion Problem.

4.1 The Matrix Completion problem

The matrix completion problem is a special case of the affine rank minimization problem with constraints that restrict some of the entries of the matrix variable X to equal given values. The problem can be written as

$$\begin{array}{ll} \text{minimize} & \text{rank}(X) \\ \text{subject to} & X_{ij} = (X_0)_{ij}, \quad (i,j) \in \Omega \end{array}$$

where X_0 is the matrix we would like to recover and Ω denotes the set of entries which are revealed. We define the operator $\mathcal{P}_\Omega : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ as

$$(\mathcal{P}_\Omega(X))_{ij} = \begin{cases} X_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

Also, Ω^c denotes the complement of Ω, i.e., all index pairs (i,j) except those in Ω.

4.2 IRLS-GP

To apply Algorithm 1 to the matrix completion problem (25), we replace the constraint $\mathcal{A}(X) = b$ by $\mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0)$. Each iteration of IRLS solves a quadratic program (QP). We note that calculating $\mathcal{P}_\Omega(X)$ is computationally cheap, so the gradient projection algorithm could be used to solve the quadratic program (QP) in each iteration of IRLS. We call this implementation IRLS-GP (Algorithm 2).

Data: Ω, $\mathcal{P}_\Omega(X_0)$
Result: $\widehat{X} : \mathcal{P}_\Omega(\widehat{X}) = b$
Set $k = 0$. Initialize $X^0 = 0, \gamma^0 > 0$, $s^0 = (\gamma^0)^{(1-\frac{p}{2})}$;
while *IRLS iterates not converged* **do**
  $W^k = ({X^k}^T X^k + \gamma^k I)^{\frac{p}{2}-1}$. Set $X_{\text{temp}} = X^k$;
  while *Gradient projection iterates not converged* **do**
    $X_{\text{temp}} = \mathcal{P}_{\Omega^c}(X_{\text{temp}} - s^k X_{\text{temp}} W^k) + \mathcal{P}_\Omega(X_0)$;
  end
  Set $X^{k+1} = X_{\text{temp}}$;
  Choose $0 < \gamma^{k+1} \leq \gamma^k$, $s^{k+1} = (\gamma^{k+1})^{(1-\frac{p}{2})}$;
  $k = k + 1$;
end

Algorithm 2: IRLS-GP for Matrix Completion

The step size used in the gradient descent step is $s^k = 1/L^k$, where $L^k = 2\|W^k\|_2$ is the Lipschitz constant of the gradient of the quadratic objective, $\text{Tr}(W^k X^T X)$ at the k^{th} iteration of IRLS. We also warm-start the gradient projection algorithm to solve for the $(k+1)^{th}$ iterate of IRLS with the solution of the k^{th} iterate of IRLS and find that this speeds up the convergence of the gradient projection algorithm in subsequent iterations. At each iteration of IRLS, computing the weighting matrix involves an inversion operation which can be expensive for large n. To work around this, we observe that the singular values of subsequent iterates of IRLS cluster into two distinct groups, so that a low rank approximation of the iterates (obtained by setting the smaller set of singular values to zero) can be used to compute the weighting matrix efficiently. Computing the singular value decomposition (SVD) can be expensive. Randomized algorithms (e.g., Halko et al., 2011) can be used to compute the top r singular vectors and singular values of a matrix X efficiently, with small approximation errors, if $\sigma_{r+1}(X)$ is small. We describe our computations of the weighting matrix below.

Computing the Weighting matrix efficiently
Let $U\Sigma V^T$ be the truncated SVD of X^k (keeping top r terms in the SVD with r being determined at each iteration), so that $U \in \mathbb{R}^{m\times r}, \Sigma \in \mathbb{R}^{r\times r}, V \in \mathbb{R}^{n\times r}$. Then for $p = 0$, ${W^k}^{-1} = \frac{1}{\gamma^k}(U\Sigma V^T)^T(U\Sigma V^T) + \gamma^k I_n$. It is easy to check that $W^k = V(\gamma^k(\Sigma^2 + \gamma^k I)^{-1} - I_r)V^T + I_n$. Thus the cost of computing the weighting matrix given the truncated SVD is $O(nr^2)$, saving significant computational costs. At each iteration, we choose r to be $\min\{r_{\max}, \widehat{r}\}$ where $\widehat{r}$ is the largest integer such that $\sigma_{\widehat{r}}(X^k) > 10^{-2} \times \sigma_1(X^k)$ (this is justified in our experiments, since X_0 is generated randomly and has a reasonable condition number). Also, since the singular values of X^k tend to separate into two clusters, we observe that this choice eliminates the cluster with smaller singular values and gives a good estimate of the rank r to which X^k can be well approximated. We find that combining warm-

starts for the gradient projection algorithm along with the use of randomized algorithms for SVD computations speeds up the overall computational time of the gradient projection implementation considerably.

5. sIRLS-p: A first-order algorithm for minimizing the smooth Schatten-p function

In this section, we present the sIRLS-p family of algorithms that are related to the IRLS-GP implementation discussed in the previous section. We first describe the algorithm before discussing its connection to the IRLS-p family.

Data: $\mathcal{P}_\Omega$, b
Result: $\widehat{X} : \mathcal{P}_\Omega(\widehat{X}) = b$
Set $k = 0$. Initialize $X^0 = \mathcal{P}_\Omega(X_0), \gamma^0 > 0$, $s^0 = (\gamma^0)^{(1-\frac{p}{2})}$;
while *not converged* **do**
  $W_p^k = ({X^k}^T X^k + \gamma^k I)^{\frac{p}{2}-1}$;
  $X^{k+1} = \mathcal{P}_{\Omega^c}(X^k - s^k X^k W_p^k) + \mathcal{P}_\Omega(X_0)$;
  Choose $0 < \gamma^{k+1} \leq \gamma^k$, $s^{k+1} = (\gamma^{k+1})^{(1-\frac{p}{2})}$;
  $k = k + 1$
end

Algorithm 3: sIRLS-p for Matrix Completion Problem

We note that sIRLS-p (Algorithm 3) is a gradient projection algorithm applied to

$$\begin{array}{ll} \text{minimize} & f_p(X) = \mathrm{Tr}(X^T X + \gamma I)^{\frac{p}{2}} \\ \text{subject to} & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0), \end{array} \tag{26}$$

while sIRLS-0 is a gradient projection algorithm applied to

$$\begin{array}{ll} \text{minimize} & \log\det(X^T X + \gamma I) \\ \text{subject to} & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0). \end{array} \tag{27}$$

Indeed $\nabla f_p(X^k) = X^k W_p^k$ and the gradient projection iterates,

$$\begin{aligned} X^{k+1} &= \mathcal{P}_{\{X:\mathcal{P}_\Omega(X)=\mathcal{P}_\Omega(X_0)\}}(X^k - s^k \nabla f_p(X^k)) \\ &= \mathcal{P}_{\Omega^c}(X^k - s^k \nabla f_p(X^k)) + \mathcal{P}_\Omega(X_0) \end{aligned}$$

are exactly the same as the sIRLS-p (Algorithm 3) iterates when γ^k is a constant. In other words, for large enough k (i.e., k such that $\gamma^k = \gamma_{\min}$), the iterates of sIRLS-p and sIRLS-0 are nothing but gradient projection applied to (26) and (27), with $\gamma = \gamma_{\min}$. We have the following convergence results for sIRLS.

Theorem 5.1 *Every cluster point of sIRLS-p* ($0 < p \leq 1$) *is a stationary point of (26) with* $\gamma = \gamma_{\min}$.

Theorem 5.2 *Every cluster point of sIRLS-0 is a stationary point of (27) with* $\gamma = \gamma_{\min}$.

The proof of Theorem 5.1 can be found in appendix C, and Theorem 5.2 can be proved in a similar way. With $p = 1$, Theorem 5.1 implies that sIRLS-1 has the same performance guarantee as IRLS-1 given in Theorem 3.6. Note that despite sIRLS-p with $p < 1$ being a gradient projection algorithm applied to non-convex problems (26) and (27), a simple step-size suffices for convergence, and we do not consider a potentially expensive line search at each iteration of the algorithm.

We now relate the sIRLS-p family to the IRLS-p family of algorithms. Each iteration of the IRLS-p algorithm is a quadratic program, and IRLS-GP uses iterative gradient projection to solve this quadratic program. We note that sIRLS-p is nothing but IRLS-GP, with each quadratic program solved approximately by terminating at the first iteration in the gradient projection inner loop. Barring this connection, the IRLS-p and sIRLS-p algorithms can be seen as two different approaches to solving the smooth Schatten-p minimization problem (26). We examine the trade off between these two algorithms along with comparisons to other state of the art algorithms for matrix completion in the next section.

6. Numerical results

In this section, we give numerical comparisons of sIRLS-$0, 1$ and IRLS-$0, 1$ with other algorithms. We begin by examining the behavior of IRLS-0 (through the IRLS-GP implementation) and its sensitivity to γ^k (regularization parameter in the weighting matrix, W^k).

6.1 Choice of regularization γ

We find that IRLS-0 converges faster when the regularization parameter in the weighting matrix, γ^k, is chosen appropriately. We consider an exponentially decreasing model $\gamma^k = \gamma^0/(\eta)^k$, where γ^0 is the initial value and η is a scaling parameter. We run sensitivity experiments to determine good choices of γ^0 and η for the matrix completion problem. For this and subsequent experiments, the indices corresponding to the known entries Ω are generated using i.i.d Bernoulli $\{0, 1\}$ random variables with a mean support size $|\Omega| = q$, where q/n^2 is the probability for an index (i, j) to belong to the support set. The completed and unknown matrix X_0 of rank r is generated as YY^T, where $Y \in \mathbb{R}^{n\times r}$ is generated using i.i.d gaussian entries. All experiments are conducted in Matlab on a Intel 3 Ghz core 2 duo processor with 3.25 GB RAM.

As will be seen from the results, the regularization parameter γ^k plays an important role in the recovery. We let $\gamma^0 = \gamma_c \|X_0\|_2^2$ where γ_c is a proportional parameter that needs to be estimated. For the sensitivity analysis of IRLS-0 (with respect to γ^0 and η), we consider matrices of size 500×500.

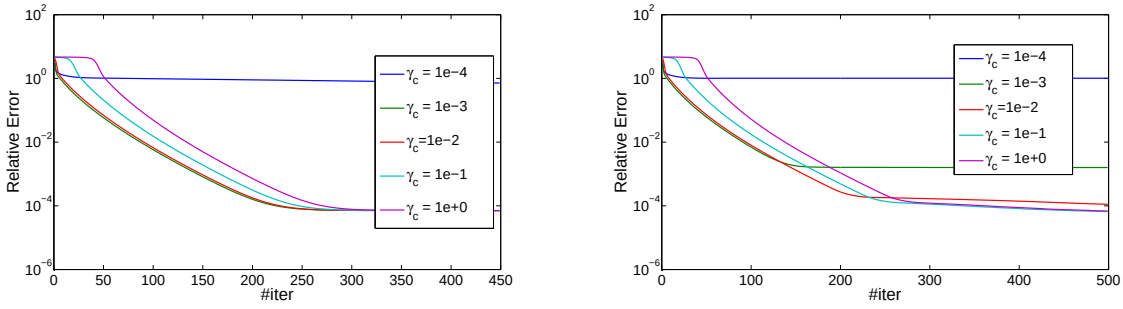

Figure 1: $n = 500$, $r = 5$,$\eta = 1.15$. $\gamma^0 = \gamma_c * \|X_0\|_2^2$. *From left to right*: Recovery error using IRLS-0 for $\|X_0\|_2 = 1, \|X_0\|_2 = 1000$.

6.1.1 Choice of γ_c

As can be seen from Figures 1 a) and b), choosing γ_c appropriately leads to a better convergence rate of IRLS-0. Small values of γ_c ($< 10^{-3}$) don't give good recovery results (premature convergence to a larger relative error). However larger values of γ_c (> 1) might lead to a delayed convergence. As a heuristic, we observe that $\gamma_c = 10^{-2}$ works well. We also note that this choice of γ_c works well even if the spectral norm of X_0 varies from 1 to 1000. Thus for future experiments, we normalize X_0 to have a spectral norm of 1.

6.1.2 Choice of η

Figures 2 a),b),c), and d) look at the sensitivity of the IRLS-0 algorithm to the scaling parameter, η. We observe that for a good choice of γ^0 (described earlier), η depends on the rank of X_0 to be recovered. More specifically, η seems to have an inverse relationship with the rank of X_0. From Figures 2 a) and d), it is clear that $\eta = 1.3$ works well if rank of X_0 equals 2 and $\eta = 1.05$ works well when rank of X_0 equals 15. More generally, the choice of η seems to depend on the hardness of the problem instance being considered. We formalize this notion in the next section.

6.2 Numerical experiments

We classify our numerical experiments into two categories based on the *degrees of freedom ratio*, given by $\text{FR} = r(2n - r)/q$. Note that for a $n \times n$ matrix of rank r, $r(2n - r)$ is the number of degrees of freedom in the matrix. Thus if FR is large (close to 1), recovering X_0 becomes harder (as the number of measurements is close to the degrees of freedom) and conversely if FR is close to zero, recovering X_0 becomes easier.

We conduct subsequent numerical experiments over what we refer to in this paper as *Easy problems* ($FR < 0.4$) and *Hard problems*($FR > 0.4$). We define the recovery to be successful when the relative error, $\|X - X_0\|_F / \|X_0\|_F \leq 10^{-3}$ (with X being the output of the algorithm considered) and unsuccessful recovery otherwise. For each problem (easy or

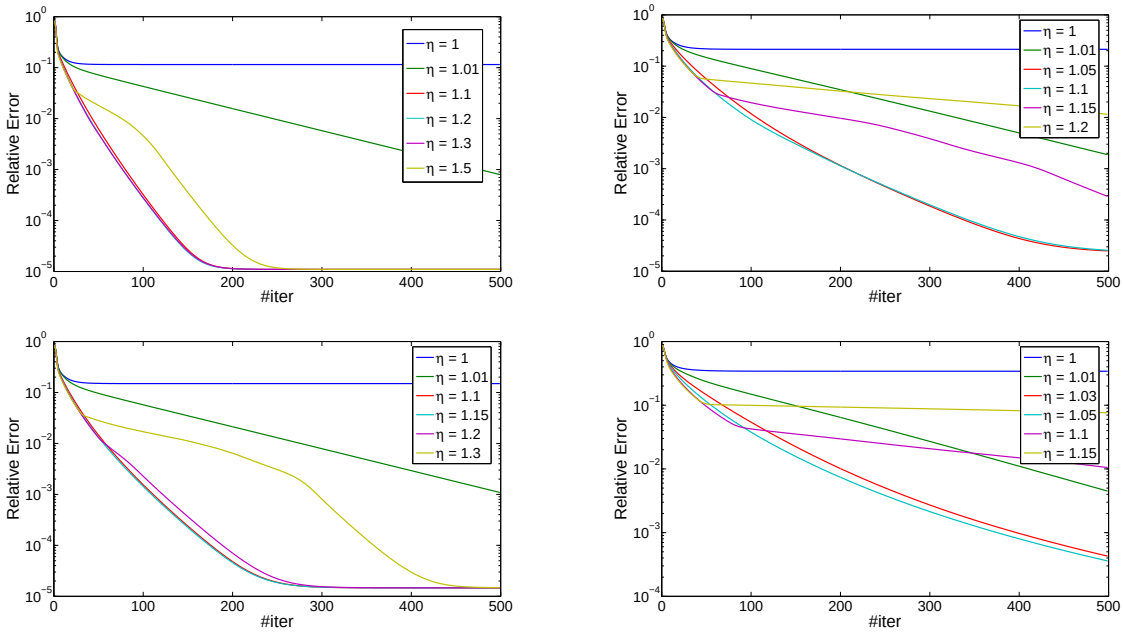

Figure 2: $n = 500$ $\gamma_c = 1e-2$. *Clockwise from top left*: Recovery error using IRLS-0 for ranks $2, 5, 10, 15$ respectively.

hard), the results are reported over 10 random generations of the support set, Ω and X_0. We use NS to denote the number of successful recoveries for a given problem. Also, computation times are reported in seconds. For sIRLS and IRLS-GP (implementation of IRLS), we fix $\eta = 1.1$ if $FR < 0.4$ and $\eta = 1.03$ if $FR > 0.4$ based on our earlier observations. In the next few sections, we compare the IRLS implementations with other state of the art algorithms on both exact and noisy matrix completion problems.

6.3 Comparison of (s)IRLS and Nuclear Norm Minimization

In this section, we compare the gradient projection implementation IRLS-GP, of IRLS-0,1 and the algorithm sIRLS-0,1 with the *Singular Value Thresholding* (SVT) algorithm (an implementation for nuclear norm minimization (Cai et al., 2008)) on both easy and hard problem sets. Note that SVT is not the only implementation of nuclear norm minimization. Other implementations include NNLS (Toh and Yun, 2010) and Spectral Regularization (Mazumder et al., 2010).

When we refer to IRLS-0,1 in the tables and in subsequent paragraphs, we mean their gradient projection implementation, IRLS-GP. We compare (s)IRLS-0,1 and SVT in Tables 1 and 3. A few aspects of these comparisons are highlighted below.

Problem				IRLS-1		sIRLS-1		IRLS-0	
n	r	$\frac{q}{n^2}$	FR	# iter	Time	# iter	Time	# iter	Time
100	10	0.57	0.34	133	4.49	132	1.63	54	0.79
200	10	0.39	0.25	140	4.49	140	2.41	60	1.34
500	10	0.2	0.2	160	24.46	163	8	77	9.63
500	10	0.12	0.33	271	37.47	336	13.86	220	22.74
1000	10	0.12	0.17	180	113.72	195	32.21	109	55.42
1000	50	0.39	0.25	140	134.30	140	102.64	51	59.74
1000	20	0.12	0.33	241	156.09	284	57.85	188	96.20
2000	20	0.12	0.17	180	485.24	190	166.28	100	235.94
2000	40	0.12	0.33	236	810.13	270	322.96	170	432.34

Table 1: Comparison of IRLS-0,1 and sIRLS-1. Performance on Easy Problems $FR < 0.4$.

Problem				sIRLS-1		sIRLS-0		SVT	
n	r	$\frac{q}{n^2}$	FR	# iter	Time	# iter	Time	# iter	Time
100	10	0.57	0.34	132	1.63	59	0.84	170	5.69
200	10	0.39	0.25	140	2.41	63	1.31	109	3.74
500	10	0.2	0.2	163	8	98	4.97	95	5.9
500	10	0.12	0.33	336	13.86	280	11.03	-	-
1000	10	0.12	0.17	195	32.21	140	20.80	85	10.71
1000	50	0.39	0.25	140	102.64	60	61.32	81	49.17
1000	20	0.12	0.33	284	57.85	241	43.11	-	-
2000	20	0.12	0.17	190	166.28	130	98.55	73	42.31
2000	40	0.12	0.33	270	322.96	220	227.07	-	-

Table 2: Comparison of sIRLS-0,1 with SVT. Performance on Easy Problems $FR < 0.4$.

Problem				sIRLS-1			IRLS-0			sIRLS-0		
n	r	$\frac{q}{n^2}$	FR	# iter	NS	Time	# iter	NS	Time	# iter	NS	Time
40	9	0.5	0.8	4705	4	163.2	1385	10	17.36	2364	9	30.22
100	14	0.3	0.87	10000	0	545.91	4811	10	89.51	5039	7	114.54
500	20	0.1	0.78	10000	0	723.58	4646	8	389.66	5140	10	315.57
1000	20	0.1	0.4	645	10	142.84	340	10	182.78	406	10	97.15
1000	20	0.06	0.66	10000	0	1830.98	2679	10	921.15	2925	10	484.84
1000	30	0.1	0.59	1152	10	295.56	781	10	401.98	915	10	244.23
1000	50	0.2	0.49	550	10	342	191	10	239.77	270	10	234.25

Table 3: Comparison of sIRLS-1,IRLS-0 and sIRLS-0. Performance on Hard Problems $FR \geq 0.4$

6.3.1 IRLS-0 vs IRLS-1

Between IRLS-0 and IRLS-1, IRLS-0 takes fewer iterations to converge successfully and has a lower computational time (Table 1). The same holds true between sIRLS-0 and sIRLS-1. sIRLS-0 is also successful on more hard problem instances than sIRLS-1 (Table 3). This indicates that (s)IRLS-p with $p = 0$ has a better recovery performance and computational time as compared to $p = 1$.

6.3.2 IRLS vs sIRLS

Between sIRLS and IRLS, sIRLS-1 takes more iterations to converge as compared to IRLS-1. However because it has a lower per iteration cost, sIRLS-1 takes significantly lower computational time than IRLS-1 (Table 1). The same holds true for sIRLS-0. Thus sIRLS-0,1 are not only simpler algorithms, they also have a lower overall run time as compared to IRLS-0,1.

6.3.3 Comparison on easy problems

Table 2 shows that sIRLS-0 and sIRLS-1 have competitive computational times as compared to SVT (implementation available at Candes and Becker, 2010). There are also certain instances where SVT fails to have successful recovery while sIRLS-1 succeeds. Thus sIRLS-1 is competitive and in some instances better than SVT.

6.3.4 Comparison on hard problems

For hard problems, Table 3 shows that sIRLS-0 and IRLS-0 are successful in almost all problems considered, while sIRLS-1 is not successful in 4 problems. We also found that SVT was not successful in recovery for any of the hard problems. (s)IRLS-0 also compares favorably with FPCA (Goldfarb and Ma, 2011) and Optspace (Keshavan and Oh, 2009) in terms of recovery and computational time on the easy and hard problem sets. These results are given subsequently.

In summary, (s)IRLS-1,(s)IRLS-0 have a better recovery performance than a nuclear norm minimization implementation (SVT) as evidenced by successful recovery over both easy and hard problem sets. We note that (s)IRLS-1 converges to the Nuclear Norm Minimizer (when the regularization, $\gamma \to 0$) and empirically has a better recovery performance than SVT. We also note that among the family of (s)IRLS-p algorithms tested, sIRLS-0 and IRLS-0 are better in both recovery performance and computational times.

6.4 Comparison of algorithms for Exact Matrix Completion

As observed in the previous section, sIRLS has a lower total run time compared to IRLS-GP. Thus in subsequent experiments we compare other algorithms only with sIRLS.

Problem				sIRLS		IRLSM		IHT		Optspace	
n	r	$\frac{q}{n^2}$	FR	# iter	Time	#iter	Time	#iter	Time	#iter	Time
100	10	0.57	0.34	56	0.8	25	1.16	44	0.51	27	0.6
200	10	0.39	0.25	61	0.96	33	4.62	53	0.95	19	1.28
500	10	0.2	0.2	99	4.5	61	66.57	105	3.63	18	8.08
500	10	0.12	0.33	285	13.24	124	162.15	344	12.50	29	12.45
1000	10	0.12	0.17	143	21.17	93	496.18	192	19.30	16	28.93
1000	10	0.39	0.25	60	27.39	27	777.04	46	19.58	17	1755.44
1000	20	0.12	0.33	244	45.33	160	789.33	289	40.39	38	241.94
2000	20	0.12	0.17	130	82.47	84	4320.20	179	80.76	14	428.37
2000	40	0.12	0.33	230	229.53	140	9859	270	225.46	28	4513

Table 4: Comparison of sIRLS-0,IRLSM, IHT and Optspace on Easy Problems with rank of the matrix to be recovered known apriori.

6.4.1 Design of Experiments

In this section, we report results from two sets of experiments. In the first set, we compare sIRLS-0 (henceforth referred to as sIRLS), *Iterative Hard Thresholding* algorithm (IHT) (Goldfarb and Ma, 2011; Meka et al., 2010), Optspace and IRLSM (Fornasier et al., 2010) over easy and hard problems with the assumption that the rank of the matrix to be recovered (X_0) is known. We implement IRLSM as described in (Fornasier et al., 2010) while we use the implementation of Optspace available on the authors webpage (Keshavan et al., 2009a). When the rank of X_0 (denoted as r) is known, the weighting matrix W^k for sIRLS is computed using a rank r approximation of X^k (also see section 3.1). The second set of experiments correspond to the case where the rank of X_0 is unknown, which is a more practical assumption.

6.4.2 Rank of X_0 known apriori

All the algorithms are fully successful (NS = 10) on the easy problem sets. As seen in Table 4, IRLSM and Optspace take fewer iterations to converge as compared to sIRLS and IHT. On the other hand, sIRLS and IHT are significantly faster than Optspace and much faster than IRLSM (Fornasier et al., 2010). Since IRLSM takes a significantly larger amount of time to converge, we do not test it on the hard problems. On hard problems, sIRLS, Optspace and IHT are fully successful on most of the problems (see Table 5). However, Optspace takes considerably higher time as compared to IHT and sIRLS. Thus, when the rank of X_0 is known, sIRLS is competitive with IHT in performance and computational time and much faster than Optspace and IRLSM.

Problem				sIRLS-0			IHT			Optspace		
n	r	$\frac{q}{n^2}$	FR	# iter	NS	Time	# iter	NS	Time	# iter	NS	Time
40	9	0.5	0.8	1718	10	12.67	1635	10	12.16	1543	7	6.82
100	14	0.3	0.87	4298	8	60.18	4868	10	68.56	4011	5	131.62
1000	20	0.1	0.40	417	10	78.26	466	10	65.84	69	10	409.66
1000	20	0.08	0.50	814	10	151.86	947	10	134.30	103	10	580.09
1000	20	0.07	0.57	1368	10	251.46	1564	10	225.13	147	10	806.32
1000	30	0.1	0.59	949	10	226.12	1006	10	189.33	134	10	1904.47
1000	50	0.2	0.49	270	10	123.84	254	10	105.88	46	10	2968

Table 5: Comparison of sIRLS-0, IHT and Optspace on Hard Problems with rank of the matrix to be recovered known apriori.

6.4.3 Rank of X_0 unknown

A possible disadvantage of IHT and Optspace could be their sensitivity to the knowledge of the rank of X_0. Thus, our second set of experiments compare sIRLS, IHT, Optspace and FPCA (Goldfarb and Ma, 2011) over easy and hard problems when the rank of X_0 is unknown. We use a heuristic for determining the approximate rank of X^k at each iteration for sIRLS, IHT and FPCA. Computing the approximate rank is important for speeding up the SVD computations in all of these algorithms.

Choice of rank

We choose r (the rank at which the SVD of X^k is truncated) to be $\min\{r_{\max}, \widehat{r}\}$ where $\widehat{r}$ is the largest integer such that $\sigma_{\widehat{r}}(X^k) > \alpha \times \sigma_1(X^k)$. For IHT we find that $\alpha = 5e-2$ works well while for sIRLS and FPCA, $\alpha = 1e-2$ works well. The justification for this heuristic was mentioned in Section 4.2. The SVD computations in IHT, sIRLS, Optspace and SVT are based on a randomized algorithm (Halko et al., 2011). We note that Linear-Time SVD (Drineas et al., 2006) is used to compute the SVD in the FPCA implementation (Goldfarb and Ma, 2009), and although faster than randomized SVD algorithm we use, it can be significantly less accurate.

Comparison of algorithms

All the algorithms compared are successful on the easy problems. However, Optspace takes much more time to converge on recovering matrices with high rank as can be seen from from Table 6. sIRLS, FPCA and IHT have competitive times on all the problems. For hard problems, however, sIRLS has a clear advantage over IHT, Optspace and FPCA in successful recovery (Table 7). sIRLS is fully successful on all problems except the second and third on which it has a success rate of 5 and 9 respectively. On the other hand IHT, Optspace and FPCA have partial or unsuccessful recovery in many problems. sIRLS is competitive with IHT and FPCA on computational times while Optspace is much slower

Problem				sIRLS		IHT		FPCA	Optspace	
n	r	$\frac{q}{n^2}$	FR	# iter	Time	# iter	Time	Time	# iter	Time
100	10	0.57	0.34	59	1.61	38	1.15	0.13	25	0.62
200	10	0.39	0.25	62	2.55	44	1.96	0.37	17	1.4
500	10	0.2	0.2	98	9.39	71	8.65	2.52	17	8.46
500	10	0.12	0.33	283	16.07	225	23.23	71.26	30	11.63
1000	10	0.12	0.17	140	38.44	104	31.47	11.24	14	26.31
1000	50	0.39	0.25	60	217.79	35	132.36	15.10	17	1774.08
1000	20	0.12	0.33	241	77.52	177	70.13	18.51	30	199.98
2000	20	0.12	0.17	130	236.19	98	152.36	42.06	12	374.15
2000	40	0.12	0.33	220	234.44	167	323.67	76.26	26	3466

Table 6: Comparison of sIRLS-0, IHT, FPCA and Optspace on Easy Problems when no prior information is available on the rank of the matrix to be recovered.

Problem		sIRLS			IHT			FPCA		Optspace		
n	FR	# iter	NS	Time	# iter	NS	Time	NS	Time	# iter	NS	Time
40	0.8	1498	10	12.91	-	0	-	5	1.69	-	0	-
100	0.87	4934	5	72.36	-	0	-	0	-	-	0	-
500	0.78	4859	9	326.06	-	0	-	0	-	-	0	-
1000	0.40	406	10	115.73	280	10	72.67	10	26.54	40	10	256.92
1000	0.57	1368	10	237.22	1059	10	244.49	0	-	133	5	769.29
1000	0.66	2961	10	554.25	-	0	-	0	-	-	0	-
1000	0.59	897	10	276.08	660	10	213.95	10	62.43	89	5	1420.81
1000	0.49	270	10	263.45	203	10	186.15	10	25.21	45	10	2924.68

Table 7: Comparison of sIRLS-0, IHT, FPCA and Optspace on Hard Problems when no prior information is available on the rank of the matrix to be recovered.

Problem				sIRLS			IHT			Optspace		
n	r	$\frac{q}{n^2}$	FR	# iter	NS	Time	#iter	NS	Time	#iter	NS	Time
100	10	0.57	0.34	51	10	0.55	41	10	0.44	28	10	0.29
200	10	0.39	0.25	56	10	0.78	48	10	0.60	19	10	0.56
500	10	0.2	0.2	96	10	4.11	88	10	2.76	18	10	6.82
500	10	0.12	0.33	298	10	12.60	298	10	9.56	29	10	12.45
1000	10	0.12	0.17	141	10	19.66	132	10	12.28	15	10	37.17
1000	10	0.39	0.25	50	10	20.94	40	10	15.53	18	10	1197.48
1000	20	0.12	0.33	254	10	44.35	247	10	32.03	25	10	220.60
2000	20	0.12	0.17	130	10	77.30	121	10	84.23	12	10	469.67
2000	40	0.12	0.33	236	10	221.31	227	10	170.88	28	10	4515.06

Table 8: Comparison of sIRLS, IHT and Optspace on the Noisy Matrix Completion problem.

than all the other algorithms. Thus, when the rank of X_0 is not known apriori, sIRLS has a distinct advantage over IHT, Optspace and FPCA in successfully recovering X_0 for hard problems.

6.5 Comparison of algorithms for Noisy Matrix Completion

In this subsection, we compare IHT, FPCA and sIRLS on randomly generated noisy matrix completion problems. We consider the following noisy matrix completion problem,

$$\begin{array}{ll} \text{minimize} & \text{rank}(X) \\ \text{subject to} & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(B), \end{array}$$

where $\mathcal{P}_\Omega(B) = \mathcal{P}_\Omega(X_0) + \mathcal{P}_\Omega(Z)$, X_0 is a low rank matrix of rank r that we wish to recover and $\mathcal{P}_\Omega(Z)$ is the measurement noise. Note that this noise model has been used before for matrix completion (see Cai et al., 2008). Let Z_{ij} be i.i.d Gaussian random variables with distribution $\mathcal{N}(0, \sigma^2)$. We would like the noise to be such that $\|\mathcal{P}_\Omega(Z)\|_F \leq \epsilon \|\mathcal{P}_\Omega(X_0)\|_F$ for a noise parameter ϵ. This would be true if $\sigma \sim \epsilon\sqrt{r}$ (Cai et al., 2008).

We adapt sIRLS for noisy matrix completion by replacing $\mathcal{P}_\Omega(X_0)$ by $\mathcal{P}_\Omega(B)$ in Algorithm 3. For all the algorithms tested in Table 8, we declare the recovery to be successful if $\|X - X_0\|_F / \|X_0\|_F \leq \epsilon = 10^{-3}$, where X is the output of the algorithms. Table 8 shows that sIRLS has successful recovery for easy noisy matrix completion problems with apriori knowledge of rank. The same holds true for hard problems with the true rank known apriori. Thus sIRLS has a competitive performance even for noisy recovery.

6.6 Application to Movie Lens Data Set

In this section, we consider the movie lens data sets with 100,000 ratings taken from (Mov). In particular, we consider four different splits of the 100k ratings into (training set, test set):

	sIRLS	IHT	Optspace
split 1	0.1919	0.1925	0.1887
split 2	0.1878	0.1883	0.1878
split 3	0.1870	0.1872	0.1881
split 4	0.1899	0.1896	0.1882

Table 9: NMAE for sIRLS-0 for different splits of the 100k movie-lens data set.

(`u1.base,u1.test`), (`u2.base,u2.test`), (`u3.base,u3.test`), (`u4.base,u4.test`) for our numerical experiments. Any given set of ratings (e.g. from a data split) can be represented as a matrix. This matrix has rows representing the users and columns representing the movies and an entry (i,j) of the matrix is non-zero if we know the rating of user i for movie j. Thus estimating the remaining ratings in the matrix corresponds to a matrix completion problem. For each data split, we train sIRLS, IHT, and Optspace on the training set and compare their performance on the corresponding test set. The performance metric here is *Normalized Mean Absolute Error* or NMAE given as follows. Let M be the matrix representation corresponding to the actual test ratings and X be the ratings matrix output by an algorithm when input the training set. Then

$$\text{NMAE} = \left(\sum_{i,j \in \text{supp}(M)} \frac{|M_{ij} - X_{ij}|}{|\text{supp}(M)|} \right) / (rt_{\max} - rt_{\min}),$$

where $rt_{\min}$ and $rt_{\max}$ are the minimum and maximum movie ratings possible. The choice of γ^0, η for sIRLS is the same as for the random experiments (described in previous sections). sIRLS is terminated if the maximum number of iterations exceeds 700 or if the relative error between the successive iterates is less than 10^{-3}. We set the rank of the unknown ratings matrix to be equal to 5 while running all the three algorithms. Table 9 shows that the NMAE for sIRLS, IHT, and Optspace are almost the same across different splits of the data. We note that in (Keshavan et al., 2009b), the NMAE of Optspace (Keshavan and Oh, 2009), FPCA (Goldfarb and Ma, 2011), and AdMiRA (Lee and Bresler, 2010) for data split 1 were obtained to be 0.186,0.19 and 0.242 respectively. Thus sIRLS has a NMAE that is as good as Optspace, FPCA, IHT and has a better NMAE than AdMiRA.

7. Conclusions and future directions

We proposed a family of Iterative Reweighted Least Squares algorithms (IRLS-p) for the affine rank minimization problem. We showed that IRLS-1 converges to the global minimum of the smoothed nuclear norm, and that IRLS-p with $p < 1$ converges to a stationary point of the corresponding non-convex yet smooth approximation to the rank function. We gave a matrix recovery guarantee for IRLS-1, showing that it converges to the true low-rank solution if the operator defining the constraints satisfies a certain a null space property. This

null space condition is both necessary and sufficient for low-rank recovery, thus improving on and simplifying the previous analysis for IRLS-1 (Mohan and Fazel, 2010b).

We then focused on the matrix completion problem, a special case of affine rank minimization arising in collaborative filtering among other applications, and presented efficient implementations specialized to this problem. We gave an implementation for IRLP-p for this problem using gradient projections. We also presented a related first-order algorithm, sIRLS-p, for minimizing the smooth Schatten-p function, which serves as a smooth approximation of the rank. Our first set of numerical experiments show that (s)IRLS-0 has a better recovery performance than nuclear norm minimization via SVT. We show that sIRLS-0 has a good recovery performance even when noise is present. Our second set of experiments demonstrate that sIRLS-0 compares favorably in terms of performance and runtime with IHT, Optspace, and IRLSM when the rank of the low rank matrix to be recovered is known. When the rank information is absent, sIRLS-0 shows a distinct advantage in performance over IHT, Optspace and FPCA.

7.1 Future directions

Low-rank recovery problems have recently been pursued in machine learning motivated by applications including collaborative filtering. Iterative reweighted algorithms for low-rank matrix recovery have empirically exhibited improved performance compared to unweighted convex relaxations. However, there has been a relative lack of theoretical results, as well as efficient implementations for these algorithms. This paper takes a step in addressing both of these issues, and opens up several directions for further research.

Low-rank plus sparse decomposition. The problem of decomposing a matrix into a low-rank component and a sparse component has received much attention(Chandrasekaran et al., 2011; Tan et al., 2011), and arises in graphical model identification (Chandrasekaran et al., 2010a) as well as a version of robust PCA (Candes et al., 2011), where problem sizes of practical interest are often very large. The convex relaxation proposed for this problem minimizes a combination of nuclear norm and ℓ_1 norm. An interesting direction for future work is to extend the IRLS algorithms family to this problem, by combining the vector and the matrix weighted updates. A potential feature of such an algorithm can be that the p for the vector part and the matrix part (and hence the weights) can be chosen separately, allowing control over how aggressively to promote the sparsity and the low-rank features.

Distributed IRLS. In the IRLS family, the least squares problem that is solved in every iteration is in fact *separable* in the columns of the matrix X (as also pointed out in (Fornasier et al., 2010)), so it can be solved completely in parallel. This opens the door not just to a fast parallel implementation, but also to the possibility of a partially distributed algorithm. Noting that the weight update step does not appear easy to decompose, an interesting question is whether we can use approximate but decomposable weights, so that the updates would require only local information.

Other applications for the NSP. The simple Null space Property used here, being based on only the singular values of elements in the null space, makes the connection between associated vector and matrix recovery proofs clear and transparent, and may be of independent interest (see Oymak et al., 2011).

Acknowledgements

We would like to acknowledge Ting Kei Pong for helpful discussions on the algorithms and Samet Oymak for pointing out an important singular value inequality. M. F. also acknowledges helpful discussions with Holger Rauhut.

Appendix A. IRLS-p from characterization of Smooth Schatten-p function

We can also derive the IRLS-p algorithm for $0 < p \leq 1$ by defining the following function,

$$\mathcal{F}^p(X, W, \gamma) \quad = \quad \operatorname{Tr} W^{\frac{p-2}{p}}(X^T X + \gamma I)$$

Lemma A.1 ((Argyriou et al., 2007),(Argyriou, 2010)) *Let $\gamma > 0$ and*

$$W^* = \frac{(X^T X + \gamma I)^{\frac{p}{2}}}{\operatorname{Tr}(X^T X + \gamma I)^{\frac{p}{2}}}.$$

Then

$$W^* = \underset{W}{\operatorname{argmin}} \left\{ \mathcal{F}^p(X, W, \gamma) : \; W \succ 0, \operatorname{Tr}(W) \leq 1 \right\}.$$

Thus, $\mathcal{F}^p(X, W^*, \gamma) = (f_p(X))^{\frac{2}{p}}$. Hence the problem of minimizing the *Smooth Schatten-p function*, $f_p(X)$ (3) is equivalent to the following problem:

$$\begin{array}{ll} \text{minimize} & \mathcal{F}^p(X, W, \gamma) \\ \text{subject to} & \mathcal{A}(X) = b, W \succ 0, \operatorname{Tr} W \leq 1, \end{array} \tag{28}$$

where the variables are X and W. As a relaxation to minimizing (28) jointly in X, W, one can consider minimizing (28) alternately with respect to X and W as in Algorithm (4).

Data: $\mathcal{A}$, b
Result: $\widehat{X} : \mathcal{A}(\widehat{X}) = b$
Set $k = 0$. Initialize $W_p^0 = I, \gamma^1 > 0$;
while *not converged* **do**

$$\begin{aligned} X^{k+1} &= \underset{X}{\operatorname{argmin}}\, \mathcal{F}^p(X, W^k) \\ &\quad \text{s.t. } \mathcal{A}(X) = b \end{aligned} \quad ;$$

$$\begin{aligned} W^{k+1} &= \underset{W}{\operatorname{argmin}}\, \mathcal{F}^p(X^{k+1}, W) \\ &\quad \text{s.t. } W \succ 0, \operatorname{Tr}(W) \leq 1 \end{aligned} \quad ;$$

Choose $0 < \gamma^{k+2} \leq \gamma^{k+1}$;
$k = k + 1$;
end

Algorithm 4: Alternative representation of the IRLS-p algorithm. The IRLS-p algorithm can be seen as alternatively minimizing an equivalent *Smooth Schatten-p problem*

Algorithm 4 is nothing but the IRLS-p algorithm. This gives an interpretation to the IRLS-p algorithm as alternatively minimizing an equivalent *Smooth Schatten-p problem* (28). Consider minimization with respect to W of $\mathcal{F}^p(X, W, \gamma)$ with X fixed. This problem can be re-formulated as,

$$\begin{aligned} \text{minimize} \quad & \operatorname{Tr} \widehat{W}(X^T X + \gamma I) \\ \text{subject to} \quad & \operatorname{Tr} \widehat{W}^{\frac{p}{p-2}} \leq 1 \\ & \widehat{W} \succ 0 \end{aligned} \tag{29}$$

where $\widehat{W} = W^{\frac{p-2}{p}}$. The following lemma relates $\mathcal{F}^p(X, W, \gamma)$ with $\mathcal{J}^p(X, W, \gamma)$ (defined in 7).

Lemma A.2 *Let*

$$\widehat{\mathcal{W}}^* = \frac{(X^T X + \gamma I)^{\frac{p}{2}-1}}{(\operatorname{Tr}(X^T X + \gamma I)^{\frac{p}{2}})^{\frac{p-2}{p}}}.$$

Then $\widehat{\mathcal{W}}^*$ is the optimal solution to (29) *as well as the following problem:*

$$\begin{aligned} \textit{minimize} \quad & \operatorname{Tr} \widehat{W}(X^T X + \gamma I) + \lambda(\operatorname{Tr} \widehat{W}^{\frac{p}{p-2}} - 1), \\ \textit{subject to} \quad & \widehat{W} \succ 0, \end{aligned}$$

where $\lambda = \frac{2-p}{p}(\operatorname{Tr}(X^T X + \gamma I)^{\frac{p}{2}})^{\frac{2}{p}}$. Furthermore, let

$$\widetilde{W} = \underset{W \succ 0}{\operatorname{argmin}}\, \mathcal{J}^p(X, W, \gamma).$$

Then $\mathcal{J}^p(X, \widetilde{W}, \gamma) = f_p(X)$ and

$$\underset{X}{\operatorname{argmin}} \left\{ \mathcal{F}^p(X, (\widehat{\mathcal{W}}^*)^{\frac{p}{p-2}}, \gamma) : \ \mathcal{A}(X) = b \right\} = \underset{Z}{\operatorname{argmin}} \left\{ \mathcal{J}^p(Z, \widetilde{W}, \gamma) : \ \mathcal{A}(Z) = b \right\}.$$

Thus Lemma (A.2) shows that alternatively minimizing $\mathcal{F}^p(X, W, \gamma)$ with respect to W followed by X (with constraints on W as in (29) and affine constraints on X) is equivalent to alternatively minimizing $\mathcal{J}^p(X, W, \gamma)$ with respect to W followed by X (with affine constraints on X and $\{W : W \succ 0\}$).

Proof [Proof of Lemma A.2] The Lagrangian for (29) is given by

$$L(\widehat{W}, \lambda) \quad = \quad \mathrm{Tr}\, \widehat{W}(X^T X + \gamma I) + \lambda(\mathrm{Tr}\, \widehat{W}^{\frac{p}{p-2}} - 1)$$

Note that $\lambda^* = \frac{2-p}{p}(\mathrm{Tr}(X^T X + \gamma I)^{\frac{p}{2}})^{\frac{2}{p}}, \widehat{\mathcal{W}}^* = \frac{(X^T X+\gamma I)^{\frac{p}{2}-1}}{(\mathrm{Tr}(X^T X+\gamma I)^{\frac{p}{2}})^{\frac{p-2}{p}}}$ satisfy the KKT conditions to (29). This is so because, $\mathrm{Tr}\,(\widehat{\mathcal{W}}^*)^{\frac{p}{p-2}} = 1$, $\widehat{\mathcal{W}}^* \succ 0$ and $\lambda^* > 0$. The complementary slackness is also true since the primal inequality constraint is tight. Since (29) is a convex problem and $\widehat{\mathcal{W}}^*$ satisfies the KKT conditions, we have that $\widehat{\mathcal{W}}^*$ is the optimal solution to (29). It is also easy to see that $\widetilde{W} = \underset{W \succ 0}{\mathrm{argmin}}\, \mathcal{J}^p(X, W, \gamma)$ where $\widetilde{W} = (X^T X + \gamma I)^{\frac{p}{2}-1}$. Also note that $\mathcal{J}^p(X, \widetilde{W}, \gamma) = f_p(X)$. Now,

$$\begin{aligned}
\underset{Z:\mathcal{A}(Z)=b}{\mathrm{argmin}}\, \mathcal{J}^p(Z, \widetilde{W}, \gamma) &= \underset{Z:\mathcal{A}(Z)=b}{\mathrm{argmin}}\, \mathrm{Tr}\, \widetilde{W}(Z^T Z + \gamma I) \\
&= \underset{Z:\mathcal{A}(Z)=b}{\mathrm{argmin}}\, \mathrm{Tr}(X^T X + \gamma I)^{\frac{p}{2}-1}(Z^T Z + \gamma I) \\
&= \underset{Z:\mathcal{A}(Z)=b}{\mathrm{argmin}}\, \mathrm{Tr}\, \widehat{\mathcal{W}}^*(Z^T Z + \gamma I) \\
&= \underset{Z:\mathcal{A}(Z)=b}{\mathrm{argmin}}\, \mathcal{F}^p(Z, (\widehat{\mathcal{W}}^*)^{\frac{p}{p-2}}, \gamma).
\end{aligned}$$

■

Appendix B. Proof of Theorem 3.1

We present two useful lemmas before we get to the proof

Lemma B.1 *For each $k \geq 1$, we have*

$$\mathrm{Tr}({X^k}^T X^k)^{\frac{p}{2}} \leq \mathcal{J}^p(X^1, W^0, \gamma^0) := D \tag{30}$$

where $W^0 = I$, $\gamma^0 = 1$. Also, $\lambda_j(W^k) \geq D^{(1-\frac{2}{p})}$, $j = 1, 2, \ldots, \min\{m, n\}$

Proof First, notice that

$$\begin{aligned}
\mathrm{Tr}({X^k}^T X^k)^{\frac{p}{2}} &\leq \mathrm{Tr}({X^k}^T X^k + \gamma I)^{p/2} = \mathcal{J}^p(X^k, W^k, \gamma^k) \\
&\leq \mathcal{J}^p(X^1, W^1, \gamma^1) \leq \mathcal{J}^p(X^1, W^0, \gamma^0) = D,
\end{aligned}$$

where the second and third inequalities follow from (7). This proves (30).

Furthermore, from the above chain of inequalities, we see that

$$(\|X^{k^T}X^k\|_2+\gamma)^{\frac{p}{2}} = \|(X^{k^T}X^k+\gamma I)^{\frac{p}{2}}\|_2 \leq D.$$

Using this and the definition of W^k, we obtain that

$$\|(W^k)^{-1}\|_2 = \|(X^{k^T}X^k+\gamma I)^{1-\frac{p}{2}}\|_2 = (\|X^{k^T}X^k\|_2+\gamma)^{1-\frac{p}{2}} \leq D^{(\frac{2}{p}-1)}.$$

This last relation shows that $\lambda_j(W^k) = \sigma_j(W^k) \geq 1/\|(W^k)^{-1}\|_2 \geq D^{(1-\frac{2}{p})}$ for all j. ■

Lemma B.2 *A matrix X^* is a minimizer of*

$$\begin{array}{ll} \textit{minimize} & \operatorname{Tr} WX^TX \\ \textit{subject to} & \mathcal{A}(X) = b \end{array}$$

if and only if $\operatorname{Tr}(WX^{*T}Z) = 0$ *for all* $Z \in \mathcal{N}(A)$.

Proof [Proof of Theorem (3.1)] For each $k \geq 1$, we have that

$$\begin{aligned} 2[\mathcal{J}^p(X^k,W^k,\gamma^k) - \mathcal{J}^p(X^{k+1},W^{k+1},\gamma^{k+1})] &\geq 2[\mathcal{J}^1(X^k,W^k,\gamma^k) - \mathcal{J}^1(X^{k+1},W^k,\gamma^k)] \\ &= \langle X^k, X^k\rangle_{W^k} - \langle X^{k+1}, X^{k+1}\rangle_{W^k} \\ &= \langle X^k + X^{k+1}, X^k - X^{k+1}\rangle_{W^k} \\ &= \langle X^k - X^{k+1}, X^k - X^{k+1}\rangle_{W^k} \\ &= \operatorname{Tr} W^k(X^k - X^{k+1})^T(X^k - X^{k+1}) \\ &\geq D^{(1-\frac{2}{p})}\|X^k - X^{k+1}\|_F^2 \end{aligned}$$

where the above expressions use Lemma B.2 and Lemma B.1. Summing the above inequalities over all $k \geq 1$, we have that $\lim_{n\to\infty}\left(X^n - X^{n+1}\right) = 0$.

■

Appendix C. Proof of Theorem 5.1

For any two matrices X, Y we denote $\langle X, Y\rangle_W = \operatorname{Tr} WX^TY$. We first note that the iterates of sIRLS-p satisfy

$$\begin{aligned} \mathcal{J}^p(X^{k+1},W^{k+1},\gamma^{k+1}) &\leq \mathcal{J}^p(X^{k+1},W^k,\gamma^{k+1}) \\ &\leq \mathcal{J}^p(X^{k+1},W^k,\gamma^k) \\ &\leq \mathcal{J}^p(X^k,W^k,\gamma^k). \end{aligned}$$

The last inequality follows from the Lipschitz continuity (with $L^k = 2\gamma^{k^{\frac{p}{2}-1}}$) of the gradient of $\operatorname{Tr}(W^kX^TX)$, i.e.

$$\operatorname{Tr} W^kX^TX \leq \operatorname{Tr} W^kX^{k^T}X^k + \langle 2X^kW^k, X - X^k\rangle + \frac{L^k}{2}\|X - X^k\|_F^2 \;\forall X, X^k$$

and the fact that

$$\begin{aligned} X^{k+1} &= \arg\min_{X} \|X - (X^k - X^k W^k)\|_F^2 \\ &\quad \text{s.t. } \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0). \end{aligned}$$

The convergence of $\{\mathcal{J}^p(X^k, W^k, \gamma^k)\}$ follows from Monotone convergence theorem. This also implies that the sequence $\{X^k\}$ is bounded. Hence there exists a convergent subsequence, $\{X^{n_i}\} \to X^*$. Also let $\{X^{n_i+1}\} \to \widehat{X}$. If X^* is a stationary point, we are done. Conversely, if X^* is not a stationary point to (26) then it follows that, $\widehat{X} \neq X^*$. But $\widehat{X} \neq X^*$ implies (using strict convexity) that $\mathrm{Tr}(W^*\widehat{X}^T\widehat{X}) < \mathrm{Tr}(W^*X^{*T}X^*)$ which also implies that $\mathcal{J}^p(\widehat{X}, \widehat{W}, \gamma_{\min}) < \mathcal{J}^p(X^*, W^*, \gamma_{\min})$. However since

$$\begin{aligned} \lim \mathcal{J}^p(X^i, W^i, \gamma^i) &= \lim \mathcal{J}^p(X^{n_i}, W^{n_i}, \gamma^{n_i}) \\ &= \lim \mathcal{J}^p(X^{n_{i+1}}, W^{n_{i+1}}, \gamma^{n_{i+1}}), \end{aligned}$$

we have a contradiction. Therefore, X^* is a stationary point to (26) and the theorem follows.

References

"Movie Lens data". `http://www.grouplens.org/node/73`.

A. Argyriou. A study of convex regularizers for sparse recovery and feature selection. 2010. Technical Report. Available at `http://ttic.uchicago.edu/~argyriou/papers/sparse_report.pdf`.

A. Argyriou, C.A. Micchelli, M. Pontil, and Y. Ying. A spectral regularization framework for multi-task structure learning. In *Proc. of Neural Information Processing Systems (NIPS)*, 2007.

J.F. Cai, E.J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. on Optimization*, 20(4):1956 – 1982, 2008.

E. J. Candes and S. Becker. Software for singular value thresholding algorithm for matrix completion. 2010. Available at http://svt.caltech.edu/code.html.

E. J. Candes and B. Recht. Exact matrix completion via convex optimziation. *Foundations of Computational Mathematics*, 9:717–772, 2009.

E. J. Candes and T. Tao. Decoding by linear programming. *IEEE Trans Info. Theory*, 2004.

E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

E.J. Candes, M.B. Wakin, and S. Boyd. Enhancing sparsity by reweighted l_1 minimization. *Journal of Fourier Analysis and Applications*, 14:877–905, 2008.

V. Chandrasekaran, P.A. Parrilo, and A.S. Willsky. Latent variable graphical model selection via convex optimization. 2010a. Available at http://arxiv.org/abs/1008.1290.

V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex geometry of linear inverse problems. 2010b. Available at `http://arxiv.org/abs/1012.0621`.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

R. Chartrand and V. Staneva. Restricted isometry properties and nonconvex compressive sensing. *Inverse Problems*, 24(035020):1–14, 2008.

R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *33rd International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2008.

I. Daubechies, R. DeVore, M. Fornasier, and C.S. Gunturk. Iteratively re-weighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math*, 63(1):1 – 38, 2010.

P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices ii: Computing a low rank approximation to a matrix. *SIAM Journal on Computing*, 36:158 – 183, 2006.

M. Fazel, H. Hindi, and S. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proc. American Control Conference*, Arlington, VA, 2001.

M. Fazel, H. Hindi, and S. Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *Proc. American Control Conference*, pages 2156–2162, Denver, CO, 2003.

M. Fornasier, H. Rauhut, and R. Ward. Low-rank matrix recovery via iteratively reweighted least squares minimization, 2010. Submitted to SIAM Journal of Optimization. Preprint available at `http://arxiv.org/PS_cache/arxiv/pdf/1010/1010.2471v2.pdf`.

R. Garg and R. Khandekar. Gradient descent with sparsification: An iterative algorithm for sparse recovery with restricted isometry property. In *Proc. of 26th Intl. Conf. on Machine Learning (ICML)*, 2009.

D. Goldfarb and S. Ma. Convergence of fixed point continuation algorithms for matrix rank minimization. *Foundations of Computational Mathematics*, 11(2), 2011.

D. Goldfarb and S. Ma. FPCA code. 2009. Available at http://www.columbia.edu/ sm2756/FPCA.htm.

D. Gross, Y. K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Physical Review Letters*, 105, 2010.

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2): 217–288, 2011.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

R. H. Keshavan and S. Oh. A gradient descent algorithm on the grassman manifold for matrix completion. 2009.

R. H. Keshavan, A. Montanari, and S. Oh. OPTSPACE algorithm. 2009a. `http://www.stanford.edu/~raghuram/optspace/index.html`.

R. H. Keshavan, A. Montanari, and S. Oh. Low-rank matrix completion with noisy observations: a quantitative comparison. In *Proc. 47th Annual Allerton Conference on Communication, Control, and Computing*, 2009b.

K. Lee and Y. Bresler. Admira: Atomic decomposition for minimum rank approximation. *IEEE Tran. Info. Theory*, 56(9), 2010.

A.S. Lewis. Derivatives of spectral functions. *Mathematics of Operations Research*, 21(3): 576 – 588, 1996.

Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Analysis and Appl.*, 31(3), 2008.

M. S. Lobo, M. Fazel, and S. Boyd. Portfolio optimization with linear and fixed transaction costs. *Annals of Operations Research*, 152:341–365, 2006.

A. W. Marshall and I. Olkin. *Inequalities: Theory of Majorization and Its Applications*. 1979.

R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287 – 2322, 2010.

R. Meka, P. Jain, and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In *Proc. of Neural Information Processing Systems (NIPS)*, 2010.

K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *Proc. American Control Conference*, Baltimore, MA, 2010a.

K. Mohan and M. Fazel. Iterative reweighted least squares for matrix rank minimization. In *Proc. 48th Allerton conference on Controls and communications*, Allerton, IL, 2010b.

D. Needell. Noisy signal recovery via iterative reweighted l1-minimization. In *Proc. Asilomar conference on Signals, Systems and Computers*, 2009.

D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301 – 321, 2008.

S. Oymak and B. Hassibi. New null space results and recovery thresholds for matrix rank minimization. 2010. Available at `http://arxiv.org/abs/1011.6326`.

S. Oymak, K. Mohan, M. Fazel, and B. Hassibi. A simplified approach to recovery conditions for low rank matrices. In *Proc. International Symposium on Information Theory*, 2011.

B. D. Rao and K. Kreutz-Delgado. An affine scaling methodology for best basis selection. *IEEE Transactions on Signal Processing*, 47:187 – 200, 1999.

N. Srebro, J.D.M. Rennie, and T.S. Jakkola. Maximum-Margin Matrix Factorization. In *Advances in Neural Information Processing Systems*, 2005b.

W. Tan, G. Cheung, and Y. Ma. Face recovery in conference video streaming using robust principal component analysis. In *In Proc. IEEE Intl. Conf. on Image Processing*, 2011.

K. C. Toh and S. W. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problem. *Pacific Journal of Optimization*, 6:615–640, 2010.

D. P. Wipf and S. Nagarajan. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *Journal of Selected Topics in Signal Proessing (Special Issue on Compressive Sensing)*, 4(2), 2010.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010.