Open access • Proceedings Article • DOI:10.1109/ALLERTON.2010.5706969

Iterative reweighted least squares for matrix rank minimization — Source link

Karthik Mohan, Maryam Fazel

Institutions: University of Washington

Published on: 01 Sep 2010 - Allerton Conference on Communication, Control, and Computing

Topics: Iterative method, Approximation algorithm, Underdetermined system, Singular value decomposition and Convex optimization

Forty-Eighth Annual Allerton Conference
Allerton House, UIUC, Illinois, USA
September 29 - October 1, 2010

Iterative Reweighted Least Squares for Matrix Rank Minimization

Karthik Mohan and Maryam Fazel

***Abstract*—The classical compressed sensing problem is to find the sparsest solution to an underdetermined system of linear equations. A good convex approximation to this problem is to minimize the ℓ_1 norm subject to affine constraints. The Iterative Reweighted Least Squares (IRLS-p) algorithm ($0 < p \leq 1$), has been proposed as a method to solve the ℓ_p ($p \leq 1$) minimization problem with affine constraints. Recently Chartrand et al observed that IRLS-p with $p < 1$ has better empirical performance than ℓ_1 minimization, and Daubechies et al gave 'local' linear and super-linear convergence results for IRLS-p with $p = 1$ and $p < 1$ respectively. In this paper we extend IRLS-p as a family of algorithms for the *matrix rank minimization problem* and we also present a related family of algorithms, sIRLS-p. We present guarantees on recovery of low-rank matrices for IRLS-1 under the Null Space Property (NSP). We also establish that the difference between the successive iterates of IRLS-p and sIRLS-p converges to zero and that the IRLS-0 algorithm converges to the stationary point of a non-convex rank-surrogate minimization problem. On the numerical side, we give a few efficient implementations for IRLS-0 and demonstrate that both sIRLS-0 and IRLS-0 perform better than algorithms such as Singular Value Thresholding (SVT) on a range of 'hard' problems (where the ratio of number of degrees of freedom in the variable to the number of measurements is large). We also observe that sIRLS-0 performs better than Iterative Hard Thresholding algorithm (IHT) when there is no apriori information on the low rank solution.**

I. Introduction and motivation

The *Affine Rank Minimization Problem* (ARMP), or the problem of finding the minimum rank matrix in an affine set, arises in a broad set of applications such as model order reduction [18], matrix completion, collaborative filtering[4], and quantum tomography [12]. The problem is as follows,

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

where the action of the linear operator, $\mathcal{A} : \mathbb{R}^{m\times n} \rightarrow \mathbb{R}^q$ on X is described by $\text{Tr}\, A_i^T X$, $i = 1, \ldots, p$ for $A_1, \ldots, A_p \in \mathbb{R}^{m\times n}$. Note that when X is restricted to be a diagonal matrix, ARMP reduces to the classical compressed sensing problem,

$$\begin{array}{ll} \text{minimize} & \text{card}(x) \\ \text{subject to} & Ax = b \end{array} \tag{1}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ and $\text{card}(x)$ denotes the cardinality or number of non-zeros entries of x. Many algorithms have been proposed as a relaxation to (1) including ℓ_1 minimization, greedy algorithms (e.g. CoSaMP [19]) and message-passing based algorithms [1]. Iterative reweighted ℓ_1 [5] and Iterative reweighted least squares (IRLS-p, [20]) with $0 < p \leq 1$ have been proposed to improve on the recovery performance of ℓ_1 minimization. In this paper, we are interested in the IRLS-p family of algorithms, with the $(k+1)$th iteration of the algorithm is given by

$$\begin{array}{rcll} x^{k+1} & = & \arg\min\limits_{x} & \sum\limits_i w_i^k x_i^2 \\ & & \text{s.t.} & Ax = b \end{array} \tag{2}$$

where w^k is a weighting vector with $w_i^k = (|x_i^k|^2 + \gamma)^{p/2-1}$, with $\gamma > 0$ being a regularization parameter added to ensure that w^k is well defined. x^0 is set to zero, so that first iterate is the least norm solution to $Ax = b$.

For the ARMP, algorithms with analogies to the vector case have been proposed including the Nuclear Norm Heuristic [9], AdMiRA [15], Reweighted Nuclear Norm Heuristic [18], etc. Developing efficient implementations for nuclear norm minimization is an important research area (see e.g. [2],[10]) since standard semidefinite programming solvers cannot handle large problem sizes.

We propose an IRLS-p family of algorithms for ARMP that minimizes a weighted Frobenius norm objective at each iteration. It serves to give an efficient implementation for the nuclear norm minimization heuristic for $p = 1$, and attempts to improve on its recovery performance for $p < 1$. Our focus on the IRLS-p algorithm for ARMP is further motivated by the good convergence properties of the IRLS-p for the compressed sensing problem along with recovery performance guarantees (see e.g. [8]). Chartrand et al [6], [7] showed that IRLS-p shows better empirical recovery performance than ℓ_1

The authors are with the Electrical Engineering Department, University of Washington, Seattle. Email: {karna,mfazel }@uw.edu

Research funded in part by NSF CAREER grant ECCS-0847077.

minimization for $p < 1$ and a similar performance with the reweighted ℓ_1 algorithm.

We show, under an assumption on the null space of $\mathcal{A}$, that IRLS-1 outputs the solution to nuclear norm minimization, which coincides with the lowest rank solution to $\mathcal{A}(X) = b$. Regarding convergence, we show the difference between successive iterates of IRLS-p (with $p = 0, 1$) algorithm converges to zero, and the iterates of the IRLS-0 algorithm converge to a stationary point of a non-convex rank-surrogate minimization problem. We also examine a variant of this algorithm called sIRLS(a.k.a short IRLS), that instead of minimizing the weighted quadratic objective at each iteration simply decreases the objective by taking one gradient projection step. Numerical experiments demonstrate that IRLS-0 and sIRLS applied to the matrix completion problem performs better than Singular Value Thresholding algorithm (SVT [2]) when the ratio of number of degrees of freedom to the number of measurements is large, and performs better than Iterative Hard Thresholding (IHT [11],[17]) when there is no apriori knowledge of the rank of the solution. The paper is organized as follows. In Section II, we propose the IRLS-p algorithm for ARMP and give convergence and recovery results. In Section III, we give a few implementations for IRLS-0 that are tailored for the matrix completion problem. We demonstrate the numerical performance of IRLS-0 for the matrix completion problem and compare it with SVT and IHT. The last section discusses future research directions.

II. IRLS-p FOR ARMP

Notation

Let $\mathcal{F}(b)$ be the set of all solutions to $\mathcal{A}(X) = b$. Denote by R_L the set of all rank L matrices, and by $e_L(X)$ the best rank L approximation error of X, i.e., $e_L(X) = \min_{Y \in R_L} \|Z - Y\|_*$. Denote by $\mathcal{P}_T(H)$ the projection of a matrix H onto the set T. Let $G_L(U, V)$ be a subspace of $m \times n$ matrices of rank at most L, whose row space belongs to the span of $V \in \mathbb{R}^{m \times L}$ and whose column space belongs to the span of $U \in \mathbb{R}^{n \times L}$. Let $S_L = \{G_L(U, V) : U \in R^{m \times L}, V \in \mathbb{R}^{n \times L}, U^T U = V^T V = I\}$. Let $\mathcal{N}(\mathcal{A})$ denote the null space of the operator $\mathcal{A}$. Let $\sigma_i(X)$ denote the i^{th} largest singular value of X and $\|X\|$ denote its largest singular value. The nuclear norm of a matrix X is defined as $\|X\|_* = \sum_i \sigma_i(X)$. I_k denotes the identity matrix of size $k \times k$.

A. IRLS-p

In this subsection, we give the IRLS-p algorithm for ARMP. Replacing the objective function in (1) by $\|X\|_*$, we get the nuclear norm minimization heuristic,

$$\begin{array}{ll} \min & \|X\|_* \\ \text{s.t.} & \mathcal{A}(X) = b. \end{array} \tag{3}$$

This heuristic is analogous to ℓ_1 minimization and many algorithms have been proposed to implement the heuristic efficiently for large scale problems, e.g. Singular Value Thresholding (SVT), Fixed Point Continuation algorithm (FPCA), etc. We would like to improve on the recovery performance of the nuclear norm heuristic by considering non-convex approximations to the rank function. Define the *smooth Schatten-p function* as $f_p(X) = \mathrm{Tr}(X^T X + \gamma I)^{p/2}$. 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 following problem,

$$\begin{array}{ll} \text{minimize} & f_p(X) \\ \text{subject to} & \mathcal{A}(X) = b \end{array} \tag{4}$$

We note that $\nabla f_p(X) = pX(X^T X + \gamma I)^{p/2-1}$ (see e.g. section 4.2 of [16]). The KKT conditions for (4) are equivalent to

$$\begin{aligned} X(X^T X + \gamma I)^{p/2-1} + \mathcal{A}^*(\lambda) &= 0 \\ \mathcal{A}(X) &= b \end{aligned} \tag{5}$$

Let $W_p^k = ({X^k}^T X^k + \gamma I)^{p/2-1}$. The first KKT condition is equivalent to $X = -\frac{1}{2}\mathcal{A}^*(\lambda)(X^T X + \gamma I)^{1-p/2}$. This is a fixed point equation and a solution can be obtained by iteratively solving for it as $X^{k+1} = \frac{1}{2}\mathcal{A}^*(\lambda){W_p^k}^{-1}$ along with the second KKT condition, $\mathcal{A}(X^{k+1}) = b$. Note that X^{k+1} as just defined satisfies the KKT conditions to the following convex optimization problem,

$$\begin{array}{ll} \text{minimize} & \mathrm{Tr}\, W_p^k X^T X \\ \text{subject to} & \mathcal{A}(X) = b. \end{array} \tag{6}$$

This idea leads to the IRLS-p algorithm as described in Table I. Note that we do let $p = 0$ in the algorithm in Table I, although it can't be derived from (4). For $p = 0$, IRLS-0 can be seen as the algorithm coming out of solving iteratively (as outlined previously) the KKT conditions for the non-convex problem,

$$\begin{array}{ll} \min & \log\det(X^T X + \gamma I) \\ \text{s.t.} & \mathcal{A}(X) = b \end{array} \tag{7}$$

with, $\gamma > 0$ being a regularization parameter. In the following subsections, we give convergence and guaranteed performance results for the IRLS-1 algorithm. Some of the results directly extend those for the IRLS-p algorithm for compressed sensing problem [8]. We also

Set $k = 0$, $X^0 = 0$. Do until convergence,

1) $W_p^k = (X^{k^T} X^k + \gamma^k I)^{\frac{p}{2}-1}$.

2) $X^{k+1} = \arg\min_X \operatorname{Tr}(W_p^k X^T X)$ s.t. $\mathcal{A}(X) = b$ (8)

3) Set $k = k+1$.

TABLE I
IRLS-P ALGORITHM FOR MATRIX RANK MINIMIZATION WITH $0 \le p \le 1$

study the convergence of IRLS-0. Note that the IRLS-p family of algorithms for rank minimization are an appropriate extension of the corresponding algorithms for compressed sensing. While exploring efficient implementations for IRLS-p, we found a more efficient class of gradient based algorithms which we call sIRLS-p (a.k.a short IRLS) which have similar convergence properties as IRLS-p. We described these subsequently.

B. IRLS-1 algorithm

We show that the difference between successive iterates of the IRLS-1 algorithm converges to zero. We also show under certain conditions on the null space of the operator $\mathcal{A}$ that the IRLS-1 algorithm outputs the minimum nuclear norm solution, which coincides with the lowest rank solution to $\mathcal{A}(X) = b$. In the following paragraphs, we drop the subscript on W_1^k for ease of notation.

To analyze the convergence of the iterates, we define a function,

$$\mathcal{J}^1(Z, W, \gamma) = \frac{1}{2}(\operatorname{Tr}(W Z^T Z) + \gamma \operatorname{Tr}(W) + \operatorname{Tr}(W^{-1})).$$

The second step of the IRLS-1 algorithm (Table I with $p = 1$) is equivalent to:

$$X^{k+1} = \arg\min_{Z \in \mathcal{F}(b)} \mathcal{J}^1(Z, W^k, \gamma^k)$$

We choose γ^{k+1} as $\gamma^{k+1} = \min\{\gamma^k, \sigma_{K+1}(X^{k+1})/N\}$, where K, N are fixed integers to be described later. Also note that

$$W^{k+1} = \arg\min_{W \succ 0} \mathcal{J}^1(X^{k+1}, W, \gamma^{k+1}),$$

and

$$\begin{aligned}\mathcal{J}^1(X^{k+1}, W^{k+1}, \gamma^{k+1}) &\le \mathcal{J}^1(X^{k+1}, W^k, \gamma^{k+1}) \\ &\le \mathcal{J}^1(X^{k+1}, W^k, \gamma^k) \\ &\le \mathcal{J}^1(X^k, W^k, \gamma^k). \end{aligned} \quad (9)$$

We now have the following lemmas that can be easily shown.

Lemma II.1. *For each $k \ge 1$, we have*

$$\|X^k\|_* \le \mathcal{J}^1(X^1, W^0, \gamma^0) := D \quad (10)$$

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

Lemma II.2. *The necessary and sufficient condition for X^* to be the minimizer of*

$$\begin{array}{ll}\min & \operatorname{Tr} W X^T X \\ \textit{s.t.} & \mathcal{A}(X) = b \end{array} \quad (11)$$

is $\operatorname{Tr}(W X^{*T} Z) = 0$ for all $Z \in \mathcal{N}(A)$.

The following result shows that the difference between the succesive iterates of the IRLS-1 converges to zero. The proof parallels the result in [8].

Theorem II.3. *Given any $b \in \mathbb{R}^p$, the iterates of IRLS-1, $\{X^n\}$ satisfy $\sum_{n=1}^{\infty} \|X^{n+1} - X^n\|_F^2 \le 2D^2$, where D is as defined in (10). In particular we have that $\lim_{n\to\infty} \left(X^n - X^{n+1}\right) = 0$*

The following definition plays an important role in the performance analysis of IRLS-1.

Definition II.4. *We say that the map $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$ has* Null Space Property (NSP) *of order L for $\delta > 0$ if*

$$\|\mathcal{P}_T(H)\|_* \le \delta \|\mathcal{P}_{T^\perp}(H)\|_* \quad (12)$$

holds for all $T \in S_L$ and for all $H \in \mathcal{N}(\mathcal{A})$.

Lemma II.5. *Assume that NSP (12) holds for order $2L$ (for some positive integer L) and $\gamma < 1$. Then, for any $Z, Z' \in \mathcal{F}(b)$, we have*

$$\|Z' - Z\|_* \le \frac{1+\delta}{1-\delta}\left(\|Z'\|_* - \|Z\|_* + 2e_L(Z)\right) \quad (13)$$

Proof: Let the SVD of Z be $Z = \begin{bmatrix} U_L & \tilde{U} \end{bmatrix} \begin{bmatrix} \Sigma_L & 0 \\ 0 & \tilde{\Sigma} \end{bmatrix} \begin{bmatrix} V_L & \tilde{V} \end{bmatrix}^T$, where Σ_L corresponds to the top L singular values of Z. Define the projection operators, $\mathcal{P}_U = U_L U_L^T, \mathcal{P}_V = V_L V_L^T$, $\mathcal{P}_{U^\perp} = \tilde{U}\tilde{U}^T$ and $\mathcal{P}_{V^\perp} = \tilde{V}\tilde{V}^T$. Let $T = \{X : X = \mathcal{P}_U Y \mathcal{P}_V + \mathcal{P}_U Y \mathcal{P}_{V^\perp} + \mathcal{P}_{U^\perp} Y \mathcal{P}_V \ \forall Y \in \mathbb{R}^{m\times n}\}$ and let $T^\perp$ be the orthogonal complement of T. Let $S = \{X : X = \mathcal{P}_U Y \mathcal{P}_V \ \forall Y \in \mathbb{R}^{m\times n}\}$. Then,

$$\begin{aligned}\|\mathcal{P}_{T^\perp}(Z - Z')\|_* &\le \|\mathcal{P}_{T^\perp}(Z')\|_* + \|\mathcal{P}_{T^\perp}(Z)\|_* \\ &\le \|Z'\|_* - \|\mathcal{P}_S(Z')\|_* + e_L(Z) \\ &\le \|Z\|_* + \|Z'\|_* - \|Z\|_* \\ &\quad -\|\mathcal{P}_S(Z')\|_* + e_L(Z) \\ &\le \|\mathcal{P}_S(Z)\|_* - \|\mathcal{P}_S(Z')\|_* + \|Z'\|_* \\ &\quad -\|Z\|_* + 2e_L(Z) \\ &\le \|\mathcal{P}_T(Z - Z')\|_* + \|Z'\|_* \\ &\quad -\|Z\|_* + 2e_L(Z)\end{aligned}$$

where the second inequality follows from the fact that nuclear norm of a 2×2 block matrix is lower bounded by the sum of the nuclear norms of the diagonal blocks (Note that $\|Z^{'}\|_* = \|\begin{bmatrix} U_L & \tilde{U} \end{bmatrix}^T Z^{'} \begin{bmatrix} V_L & \tilde{V} \end{bmatrix}\|_* \geq \|U_L^T Z^{'} V_L\|_* + \|\tilde{U}^T Z^{'} \tilde{V}\|_* = \|\mathcal{P}_S(Z^{'})\|_* + \|\mathcal{P}_{T^\perp}(Z^{'})\|_*$). Using (12), we have that

$$\begin{aligned}\|\mathcal{P}_T(Z-Z^{'})\|_* &\leq \delta \|\mathcal{P}_{T^\perp}(Z-Z^{'})\|_* \\ &\leq \delta(\|\mathcal{P}_T(Z-Z^{'})\|_* + \|Z^{'}\|_* \\ &\quad -\|Z\|_* + 2e_L(Z))\end{aligned}$$

Thus,

$$\begin{aligned}\|\mathcal{P}_T(Z-Z^{'})\|_* &\leq \frac{\delta}{1-\delta}\left(\|Z^{'}\|_* - \|Z\|_*\right) \\ &\quad + \frac{2\delta}{1-\delta} e_L(Z)\end{aligned}$$

(14) together with (14) implies that

$$\begin{aligned}\|Z-Z^{'}\|_* &\leq \|\mathcal{P}_T(Z-Z^{'})\|_* + \|\mathcal{P}_{T^\perp}(Z-Z^{'})\|_* \\ &\leq \frac{1+\delta}{1-\delta}(\|Z^{'}\|_* - \|Z\|_* + 2e_L(Z))\end{aligned}$$

■

The following lemma gives sufficient conditions for recovery of low-rank matrices using nuclear norm minimization and can be shown using the previous lemma.

Lemma II.6. *Assume that NSP (12) holds for order $2L$ and $\delta < 1$. Suppose that $\mathcal{F}(b)$ contains a rank L matrix. Then this matrix is the unique nuclear norm minimizer in $\mathcal{F}(b)$. We denote this minimizer by X^*. Then, we have that $\forall\, Y \in \mathcal{F}(b)$,*

$$\|Y - X^*\|_* \leq 2\frac{1+\delta}{1-\delta} e_L(Y) \tag{14}$$

We now show that the iterates of the IRLS-1 converge to the nuclear norm solution which also turns out to be the unique low rank solution under NSP.

Theorem II.7. *Let K be chosen so that $\mathcal{A}$ satisfies NSP of order $2K$, with $\delta < 1$. Also assume that $\lim_{n\to\infty} \gamma^n = 0$. Then, for each $b \in \mathbb{R}^q$, the output of IRLS-1 converges to $\bar{X}$, with $\bar{X}$ being of rank K. Also in this case, $\bar{X} = X^*$, the unique nuclear norm minimizer and $\bar{X}$ is also the unique rank K solution to $\mathcal{A}(X) = b$.*

Proof: We give a proof for the simpler case where $\gamma^n = 0$ for $n \geq n_0 + 1$. By definition, $\gamma^{n_0+1} = \min\{\gamma^{n_0}, \sigma_{K+1}/N\}$. Since $\gamma^{n_0} \neq 0$, $\sigma_{K+1}(X^{n_0}) = 0$ implying that X^{n_0} is of rank K. Thus, by Lemma II.6 we have that X^{n_0} is the unique nuclear norm minimizer. The proof for the case where $\gamma^n > 0\ \forall\, n$ can be shown using standard convergence arguments. ■

As an extension of the above theorem, one can also consider the case where $\lim_{n\to\infty} \gamma^n = \bar{\gamma} > 0$.

C. IRLS-0

In this section we give a convergence result for the IRLS-0 algorithm. We define an appropriate function that can be used to show convergence as follows:

$$\mathcal{J}^0(X, W, \gamma) = \operatorname{Tr}(W X^T X) + \gamma \operatorname{Tr} W - \log\det W$$

Note that the above function is strictly convex in W and also in X (because W is positive definite). As in (9),

$$\begin{aligned}\mathcal{J}^0(X^{k+1}, W^{k+1}, \gamma^{k+1}) &\leq \mathcal{J}^0(X^{k+1}, W^k, \gamma^{k+1}) \\ &\leq \mathcal{J}^0(X^{k+1}, W^k, \gamma^k) \\ &\leq \mathcal{J}^0(X^k, W^k, \gamma^k)\end{aligned} \tag{15}$$

Analogous to Lemma II.1 we have:

Lemma II.8. *For each $k \geq 1$, we have*

$$\begin{array}{rcl}\log\det(X^{k^T} X^k) & \leq & \mathcal{J}^0(X^1, W^0, \gamma^0) := E \\ \sigma_j(W^k) & \geq & e^{-E},\ j = 1, 2, \ldots, t\end{array} \tag{16}$$

where, $t = \min\{m, n\}$.

Analogous to Theorem II.3 we have the following:

Theorem II.9. *Given any $b \in \mathbb{R}^q$, the iterates of IRLS-0, $\{X^n\}$ satisfy*

$$\sum_{n=1}^{\infty} \|X^{n+1} - X^n\|_F^2 \leq 2e^{2E}, \tag{17}$$

where E is as defined in (16). In particular we have that $\lim_{n\to\infty} \left(X^n - X^{n+1}\right) = 0$

Using this theorem we can show the following:

Theorem II.10. *Every cluster point of the iterates $\{X^k\}$ of IRLS-0 is a stationary point of (7) with $\gamma = \gamma_{\min} = \lim \gamma^k$.*

D. sIRLS-p

In this subsection, we describe the sIRLS-p (a.k.a short IRLS) algorithm and show its convergence for the matrix completion problem, where we would like to complete a low-rank matrix given only a subset of its entries. Although we don't discuss it here, the algorithm can easily be extended to solve for general affine constraints. The matrix completion problem is as follows,

$$\begin{array}{ll}\text{minimize} & \operatorname{rank}(X) \\ \text{subject to} & \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0),\end{array}$$

Set $k = 0$. $X^0 = 0$.
Do until convergence,
1) $W_p^k = ({X^k}^T X^k + \gamma^k I)^{\frac{p}{2}-1}$.
2) $X^{k+1} = \mathcal{P}_{\Omega^c}(X^k - \alpha^k X^k W_p^k) + \mathcal{P}_\Omega(X_0)$
3) Set $k = k + 1$.

TABLE II
sIRLS-p FOR MATRIX COMPLETION PROBLEM

where X_0 is a matrix we would like to recover, $\mathcal{P}_\Omega : \mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{n\times n}$ is an operator that samples entries X_{ij} from X where $(i,j) \in \Omega$. sIRLS has a simple structure for the matrix completion problem and is given as in Table II. In the next section, we give a fast implementation (IRLS-GP, Table III) for IRLS-p where each iterate is obtained through a gradient projection algorithm. Thus, sIRLS-p can be thought of as IRLS-p with each of its iterates solved approximately (i.e. terminating at the first iteration in the gradient projection algorithm instead of until convergence). Hence the name *short IRLS* even though sIRLS doesn't involve solving a least squares problem. We observe in the next section that sIRLS-0 is much faster than IRLS-0 for matrix completion with little or no loss in performance as evidenced by successful recovery in most easy and hard problem instances (a precise notion of easy/hard will be given in the implementation section). We now give convergence results for sIRLS-p.

Theorem II.11. *The difference between successive iterates of the sIRLS-p converges to zero.*

Interestingly, for a fixed γ, sIRLS-p ($0 < p \leq 1$) can be viewed as a Gradient Projection algorithm applied to (4) and sIRLS-0 can be viewed as a Gradient Projection algorithm applied to (7) with matrix completion constraints. We therefore have the following theorem for sIRLS-p.

Theorem II.12. *Every cluster point of sIRLS-p is a stationary point of the* smooth Schatten-p function, $f_p(X)$ *over the constraint set,* $\{X : \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0)\}$ *with* $\gamma = \gamma_{\min}$.

It can be also shown that every cluster point of sIRLS-0 is a stationary point of (7) with matrix-completion constraints. Note that for $p = 1$, both IRLS-1 and sIRLS-1 converge to the global minimum of the *smooth Schatten-*1 *problem* (4).

III. ALGORITHM IMPLEMENTATION AND NUMERICAL RESULTS

In this section, we give a fast implementation of the IRLS-p algorithm for the Matrix Completion problem.

Set $k = 0$, $X^0 = 0$. Do until IRLS iterates converge,
1) $W_p^k = ({X^k}^T X^k + \gamma^k I)^{\frac{p}{2}-1}$. Set $X_{\text{old}} = X^k$.
2) Do until gradient projection iterates converge,
   a) $X_{\text{new}} = \mathcal{P}_{\Omega^c}(X_{\text{old}} - \frac{2}{L^k} X_{\text{old}} W_p^k) + \mathcal{P}_\Omega(X_0)$
   b) $X_{\text{old}} = X_{\text{new}}$
3) Set $X^{k+1} = X_{\text{new}}$, $k = k + 1$.

TABLE III
IRLS$_p$-GP FOR MATRIX COMPLETION

We also give numerical comparisons of sIRLS-p with other algorithms. For ease of notation throughout this section, we shall refer to IRLS-p and sIRLS-p as IRLS and sIRLS respectively unless we specifically refer to the algorithms with $p = 0$ or $p = 1$.

A. A fast gradient projection based implementation of IRLS for Matrix Completion

IRLS for matrix completion problem is similar to that in Table I with the constraints $\mathcal{A}(X) = b$ replaced by $\mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(X_0)$. The implementation we describe in this section solves each iteration of IRLS approximately so that the overall computational time is smaller but at the same time the performance in terms of recovering low-rank solutions for different problem instances is preserved. Now, each iteration of IRLS solves a quadratic program (QP). We note that calculating $\mathcal{P}_\Omega(X)$ is computationally cheap. Thus, the gradient projection algorithm could be used to solve the quadratic program (QP) in each iteration of the IRLS. The implementation IRLS$_p$-GP is as in Table III. The step size used in the gradient descent step is $2/L^k$, where L^k 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 (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 (see e.g. [13]) can be used to compute the top r singular vectors and singular values of a matrix X 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 $W^{k^{-1}} \sim (U\Sigma V^T)^T(U\Sigma V^T) + \gamma^k I$. It is easy to check that $W^k \sim V(\Sigma^2 - \frac{1}{\gamma^k}I_r)V^T + \frac{1}{\gamma^k}I_n$. Thus the computation of the weighting matrix is of $O(nr^2)$ saving significant computational costs. We choose r to be $\min\{r_{\max}, \hat{r}\}$ where $\hat{r}$ is the largest integer such that $\sigma_{\hat{r}}(X^k) > 10^{-2} \times \sigma_1(X^k)$ (this is justified since in our experiments 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.

B. Behavior of IRLS-p

We begin our numerical experiments by examining the behavior of IRLS-0 and its sensitivity to γ^k (regularization parameter in the weighting matrix, W^k). We then compare sIRLS-1, IRLS-1 with sIRLS-0, IRLS-0 and Singular Value Thresholding (SVT), an algorithm for nuclear norm minimization. Note that in this subsection, by IRLS-p we refer to the implementation IRLS$_p$-GP given in Table III.

We find that the IRLS-0 works well when γ^k are chosen appropriately. We let $\gamma^k = \gamma^0/(\eta)^k$, where γ^0 is the initial regularization parameter and η is a scaling parameter. For this sensitivity experiment (and subsequent experiments), the support set Ω is generated using bernoulli $\{0,1\}$ random variables with a mean support size of q where q/n^2 is the bernoulli probability for an index (i,j) to belong to the support set. X_0 of rank r is generated as YY^T, where $Y \in \mathbb{R}^{n\times r}$ is generated using iid gaussian entries, X_0 is normalized so that its maximum singular value is 1. All experiments are conducted in Matlab on a Intel 3 Ghz core 2 duo processor with 3.25 GB RAM.

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 (with respect to γ^0 and η), we consider recovery of matrices of size 500×500. As can be seen from figure 1, choosing γ_c is critical to the performance of IRLS-0. Small values of γ_c ($< 10^{-3}$) don't give good recovery results (premature convergence). However larger values of γ_c might lead to delayed convergence.

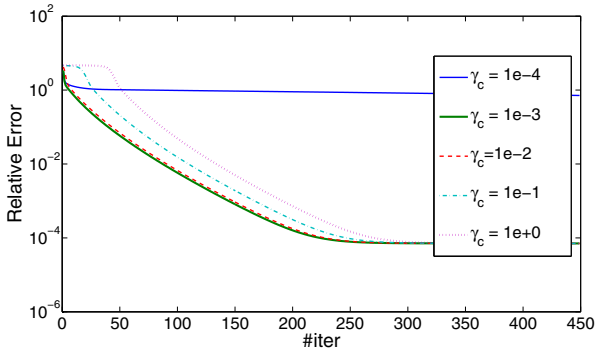

Fig. 1. $n = 500$, $rank = 5$,$\eta = 1.15$. Sensitivity of IRLS to γ^0. $\gamma^0 = \gamma_c * \|X_0\|_2^2$. *Top to bottom*: Recovery error using IRLS for $\|X_0\|_2 = 1$

5

Hence as a heuristic, we observe that $\gamma_c = 10^{-2}$ works well for IRLS-0. We use a similar heuristic for IRLS-1.

Note that for a $n \times n$ matrix of rank r, $r(2n-r)$ is the number of degrees of freedom in the matrix. Define, FR (degrees of freedom ratio) to be $\frac{r(2n-r)}{q}$ and SR (sampling ratio) to be $\frac{q}{n^2}$. 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. Based on this observation, we conduct experiments over *Easy problems* ($FR < 0.4$) and *Hard problems* ($FR > 0.4$). Figure 2 looks at the sensitivity of the IRLS algorithm to the scaling parameter, η. We observe that for a good choice of γ^0 (described earlier), η depends on the hardness of the problem (i.e. on rank of X_0 and SR). More specifically, η seems to have an inverse relationship with FR. From Figure 2 it is clear that $\eta = 1.15$ works well if rank of X_0 equals 5 (i.e. FR = 0.17). We also observed that $\eta = 1.1$ and $\eta = 1.05$ work well when rank of X_0 equals 10 (FR = 0.2) and 15 (FR = 0.33) respectively. To simplify the choice of η, we fix $\eta = 1.1$ if $FR < 0.4$ and $\eta = 1.03$ if $FR > 0.4$. We define the recovery to be successful when the relative error, $\|X - X_0\|_F/\|X_0\|_F \leq 10^{-3}$ and unsuccessful recovery otherwise. For each problem (easy or hard) we consider, 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.

IRLS-0, sIRLS-0, IRLS-1, sIRLS-1 are successful in recovering all problem instances for all the easy problems considered while SVT is successful in all problems except 4,7 and 9 (Table IV). IRLS-0 takes fewer iterations to converge successfully than IRLS-1 for the easy problems and has a lower computational time.

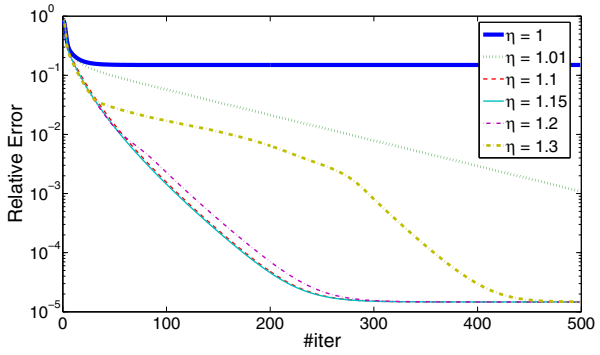

Fig. 2. $n = 500$ $\gamma_c = 1e-2$. *Top to bottom*: Recovery error using IRLS for rank equal to 5.

sIRLS-1 takes more number of iterations to converge as compared to IRLS-1 but because it has a lower per iteration cost, it takes significantly lower computational time than IRLS-1. The same holds true for sIRLS-0. Both sIRLS-0 and sIRLS-1 have computational times comparable with SVT. We use the implementation of SVT available at [3]. For hard problems, Table V shows that sIRLS-0 and IRLS-0 are successful in almost all problems considered, while sIRLS-1 doesn't successfully recover 4 problems. We also found that SVT wasn't successful in recovering any of the hard problems. Also, sIRLS-0 takes few more iterations to converge than IRLS-0 but has siginificantly lower computational times as compared to both IRLS-0 and sIRLS-1.

C. Comparison of algorithms for Exact Matrix Completion

From the previous results, it is clear that sIRLS-0 is both fast and has a good performance among the family of sIRLS-p, IRLS-p algorithms. Hence in the experiments below, we compare sIRLS-0 with IHT [11] over both easy and hard problems. We observed in our experiments that when the rank of X_0 is known, sIRLS-0 is as good as IHT in performance and computational time. A possible disadvantage of IHT is that it can be sensitive to the knowledge of the rank of the low rank solution X_0. Thus, our experiments compare sIRLS-0 and IHT over easy and hard problems without any prior knowledge on the rank of X_0. 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-0 and IHT. We choose r (the rank at which the SVD of X^k is truncated) to be $\min\{r_{\max}, \hat{r}\}$ where $\hat{r}$ is the largest integer such that $\sigma_{\hat{r}}(X^k) > \alpha \times \sigma_1(X^k)$. For IHT and sIRLS-0 we find that $\alpha = 10^{-1}$ works well for easy problems, $\alpha = 5 \times 10^{-2}, 10^{-2}$ respectively work well for hard problems. The justification this choice was mentioned previously. The SVD computations in IHT and sIRLS-0 are based on a fast yet accurate randomized algorithm for SVD computations [14]. Also, we find that a step-size of 1.5 seems to work very well for IHT. As can be seen from Table VI, the two algorithms are successful on all easy problems and also have a comparable computational times. For hard problems, however, sIRLS-0 has a distinct advantage over IHT in recovery. IHT has unsuccessful recovery in four of the problems, while sIRLS-0 is not fully successful in only the second problem which has a high FR. Thus, when the rank of X_0 is not known apriori, sIRLS-0 has a distinct advantage over IHT in successfully recovering X_0 for hard problems.

IV. Summary and Future Work

In summary, we presented the IRLS-p family of algorithms to the affine rank minimization problem. We considered the convergence properties of IRLS-1 and IRLS-0 showing that the difference between successive iterates of both the algorithms converge to zero. Under some assumptions on null space of the operator, we also showed that IRLS-1 converges to the unique nuclear norm solution which also coincides with the lowest rank solution satisfying the affine constraints. We also showed that IRLS-0 (as well as sIRLS$-p$) converges to the stationary point of the problem of minimizing a smooth rank-surrogate function. We gave an efficient gradient projection based implementation for IRLS-0, making use of the structure of the matrix completion operator.

Our first set of numerical experiments show that IRLS-0 and sIRLS-0 have a better recovery performance than SVT (an efficient implementation for nuclear norm minimization). We also give a heuristic for tuning the parameters of IRLS-0 algorithm for better performance. Our second set of experiments demonstrate that sIRLS-0 compares favorably in terms of performance and computational time with IHT when the rank of the low rank matrix to be recovered is known. When information on rank is absent, sIRLS-0 seems to have a distinct advantage in performance over IHT. Future work could focus on giving performance guarantees and convergence rate results for IRLS-0 and sIRLS-0. Other non-convex formulation ideas (e.g. decomposing the variable X into a product of two low rank matrices) may possibly be combined with IRLS-0 algorithm to make way for even faster algorithms. A unified perspective on different non-convex heuristics for rank minimization is desirable, and insights in this direction would be useful.

Problem				IRLS-1		sIRLS-1		IRLS-0		sIRLS-0		SVT	
n	r	$\frac{q}{n^2}$	FR	# iter	Time	# iter	Time	# iter	Time	# iter	Time	# iter	Time
100	10	0.57	0.34	133	4.49	132	1.63	54	0.79	59	0.84	170	5.69
200	10	0.39	0.25	140	4.49	140	2.41	60	1.34	63	1.31	109	3.74
500	10	0.2	0.2	160	24.46	163	8	77	9.63	98	4.97	95	5.9
500	10	0.12	0.33	271	37.47	336	13.86	220	22.74	280	11.03	-	-
1000	10	0.12	0.17	180	113.72	195	32.21	109	55.42	140	20.80	85	10.71
1000	50	0.39	0.25	140	134.30	140	102.64	51	59.74	60	61.32	81	49.17
1000	20	0.12	0.33	241	156.09	284	57.85	188	96.20	241	43.11	-	-
2000	20	0.12	0.17	180	485.24	190	166.28	100	235.94	130	98.55	73	42.31
2000	40	0.12	0.33	236	810.13	270	322.96	170	432.34	220	227.07	-	-

TABLE IV
COMPARISON OF IRLS(IRLS-1,sIRLS-1,IRLS-0,sIRLS-0) 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 V
COMPARISON OF SIRLS-1,IRLS-0 AND SIRLS-0. PERFORMANCE ON HARD PROBLEMS $FR \geq 0.4$

Problem				sIRLS-0			IHT		
n	r	$\frac{q}{n^2}$	FR	# iter	NS	Time	# iter	NS	Time
100	10	0.57	0.34	59	10	0.84	37	10	0.79
200	10	0.39	0.25	63	10	1.31	44	10	1.49
500	10	0.2	0.2	98	10	4.97	70	10	5.16
500	10	0.12	0.33	280	10	11.03	204	10	8.26
1000	10	0.12	0.17	140	10	20.80	103	10	17.71
1000	50	0.39	0.25	60	10	61.32	34	10	80.24
1000	20	0.12	0.33	241	10	43.11	177	10	34.81
2000	20	0.12	0.17	130	10	98.55	97	10	90.21
2000	40	0.12	0.33	220	10	227.07	166	10	202.2

TABLE VI
COMPARISON OF SIRLS-0 AND IHT. PERFORMANCE OF SIRLS ON EASY PROBLEMS $FR < 0.4$.

Problem				sIRLS-0			IHT		
n	r	$\frac{q}{n^2}$	FR	# iter	NS	Time	# iter	NS	Time
40	9	0.5	0.8	2364	9	30.22	5000	0	51.40
100	14	0.3	0.87	5039	7	114.54	5000	0	75.63
500	20	0.1	0.78	5140	10	315.57	5000	0	583.04
1000	20	0.1	0.40	406	10	97.15	280	10	72.67
1000	20	0.06	0.66	2925	10	484.84	10000	0	1175.45
1000	30	0.1	0.59	915	10	244.23	660	10	213.95
1000	50	0.2	0.49	270	10	234.25	203	10	186.15

TABLE VII
COMPARISON OF SIRLS-0 AND IHT. PERFORMANCE ON HARD PROBLEMS $FR \geq 0.4$.

REFERENCES

[1] S. Arora, C. Daskalakis, and D. Steurer. Message-passing algorithms and improved lp decoding. In *Proc. 41st annual ACM symposium on Theory of Computing*, 2009.

[2] J.F. Cai, E.J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. 2009. http://arxiv.org/abs/0810.3286.

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

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

[5] 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.

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

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

[8] I. Daubechies, R. DeVore, M. Fornasier, and C.S. Gunturk. Iteratively re-weighted least squares minimization for sparse recovery, 2008. http://arXiv.org/abs/0807.0575.

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

[10] D. Goldfarb and S. Ma. Convergence of fixed point continuation algorithms for matrix rank minimization. Technical report, Department of IEOR, Columbia University, 2009. Available at http://www.columbia.edu/~sm2756/FPCA-convergence.pdf.

[11] D. Goldfard and S. Ma. Convergence of fixed point continuation algorithms for matrix rank minimization. 2009. Technical Report. Available at http://www.columbia.edu/ sm2756/FPCA-convergence.pdf.

[12] D. Gross, Y.K. Liu, S.T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. 2010. Preprint available at http://arxiv.org/PS_cache/arxiv/pdf/0909/0909.3304v2.pdf.

[13] N. Halko, P.G. Martinsson, and J.A. Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. Technical Report 2009-05, California Institute of Technology, 2009.

[14] N. Halko, P.G. Martinsson, and J.A. Tropp. Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions. 2009. Technical Report 2009-05. Available at http://www.acm.caltech.edu/ jtropp/reports/HMT09-Finding-Structure-TR.pdf.

[15] K. Lee and Y. Bresler. Admira: Atomic decomposition for minimum rank approximation. 2009. Available at http://arxiv.org/abs/0905.0044.

[16] Z. Lu and T.K. Pong. Interior point methods for computing optimal design. 2010. Available at http://arxiv.org/PS_cache/arxiv/pdf/1009/1009.1909v1.pdf.

[17] R. Meka, P. Jain, and I.S. Dhillon. Guaranteed rank minimization via singular value projection. 2009. Available at http://arxiv.org/abs/0909.5457.

[18] K. Mohan and M. Fazel. Reweighted nuclear norm minimization with application to system identification. In *Proc. American Control Conference*, 2010.

[19] D. Needell and J.A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. Available at http://arxiv.org/abs/0905.0044Submittedon17Mar2008.

[20] B.D. Rao and K.K.Delgado. An affine scaling methodology for best basis selection. *IEEE Transactions on Signal Processing*, 1999.