Spectral Regularization Algorithms for Learning Large Incomplete Matrices

Rahul Mazumder RAHULM@STANFORD.EDU
Department of Statistics
Stanford University

Trevor Hastie HASTIE@STANFORD.EDU
Statistics Department and Department of Health, Research and Policy
Stanford University

Robert Tibshirani TIBS@STANFORD.EDU
Department of Health, Research and Policy and Statistics Department
Stanford University

Editor:

Abstract

We use convex relaxation techniques to provide a sequence of regularized low-rank solutions for large-scale matrix completion problems. Using the nuclear norm as a regularizer, we provide a simple and very efficient convex algorithm for minimizing the reconstruction error subject to a bound on the nuclear norm. Our algorithm SOFT-IMPUTE iteratively replaces the missing elements with those obtained from a soft-thresholded SVD. With warm starts this allows us to efficiently compute an entire regularization path of solutions on a grid of values of the regularization parameter. The computationally intensive part of our algorithm is in computing a low-rank SVD of a dense matrix. Exploiting the problem structure, we show that the task can be performed with a complexity linear in the matrix dimensions. Our semidefinite-programming algorithm is readily scalable to large matrices: for example it can obtain a rank-80 approximation of a $10^6 \times 10^6$ incomplete matrix with 10^5 observed entries in 2.5 hours, and can fit a rank 40 approximation to the full Netflix training set in 6.6 hours. Our methods show very good performance both in training and test error when compared to other competitive state-of-the art techniques.

1. Introduction

In many applications measured data can be represented in a matrix $X_{m\times n}$, for which only a relatively small number of entries are observed. The problem is to "complete" the matrix based on the observed entries, and has been dubbed the matrix completion problem [CCS08, CR08, RFP07, CT09, KOM09, RS05]. The "Netflix" competition (e.g. [SN07]) is a popular example, where the data is the basis for a recommender system. The rows correspond to viewers and the columns to movies, with the entry X_{ij} being the rating $\in \{1, \ldots, 5\}$ by viewer i for movie j. There are 480K viewers and 18K movies, and hence 8.6 billion (8.6×10^9) potential entries. However, on average each viewer rates about 200 movies, so only 1.2% or 10^8 entries are observed. The task is to predict the ratings that viewers would give to movies they have not yet rated.

These problems can be phrased as learning an unknown parameter (a matrix $Z_{m\times n}$) with very high dimensionality, based on very few observations. In order for such inference to be meaningful, we assume that the parameter Z lies in a much lower dimensional manifold. In this paper, as is relevant in many real life applications, we assume that Z can be well represented by a matrix of low rank, i.e. $Z \approx V_{m\times k}G_{k\times n}$, where $k \ll \min(n, m)$. In this recommender-system example, low rank structure suggests that movies can be grouped into a small number of "genres", with $G_{\ell j}$ the relative score for movie j in genre ℓ. Viewer i on the other hand has an affinity $V_{i\ell}$ for genre ℓ, and hence the modeled score for viewer i on movie j is the sum $\sum_{\ell=1}^{k} V_{i\ell}G_{\ell j}$ of genre affinities times genre scores. Typically we view the observed entries in X as the corresponding entries from Z contaminated with noise.

Recently [CR08, CT09, KOM09] showed theoretically that under certain assumptions on the entries of the matrix, locations, and proportion of unobserved entries, the true underlying matrix can be recovered within very high accuracy. [SAJ05] studied generalization error bounds for learning low-rank matrices.

For a matrix $X_{m\times n}$ let $\Omega \subset \{1, \ldots, m\}\times\{1, \ldots, n\}$ denote the indices of observed entries. We consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \operatorname{rank}(Z) \\ \text{subject to} \quad & \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \leq \delta, \end{aligned} \tag{1}$$

where $\delta \geq 0$ is a regularization parameter controlling the tolerance in training error. The rank constraint in (1) makes the problem for general Ω combinatorially hard [SJ03]. For a fully-observed X on the other hand, the solution is given by a truncated singular value decomposition (SVD) of X. The following seemingly small modification to (1),

$$\begin{aligned} \text{minimize} \quad & \|Z\|_* \\ \text{subject to} \quad & \sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 \leq \delta, \end{aligned} \tag{2}$$

makes the problem convex [Faz02]. Here $\|Z\|_*$ is the nuclear norm, or the sum of the singular values of Z. Under many situations the nuclear norm is an effective convex relaxation to the rank constraint [Faz02, CR08, CT09, RFP07]. Optimization of (2) is a semi-definite programming problem [BV04, Faz02] and can be solved efficiently for small problems, using modern convex optimization software like SeDuMi and SDPT3. However, since these algorithms are based on second order methods [LV08], they can become prohibitively expensive if the dimensions of the matrix get large [CCS08]. Equivalently we can reformulate (2) in *Lagrange* form

$$\underset{Z}{\text{minimize}}\ \tfrac{1}{2}\sum_{(i,j)\in\Omega} (X_{ij} - Z_{ij})^2 + \lambda\|Z\|_*. \tag{3}$$

Here $\lambda \geq 0$ is a regularization parameter controlling the nuclear norm of the minimizer $\hat{Z}_\lambda$ of (3); there is a 1-1 mapping between $\delta \geq 0$ and $\lambda \geq 0$ over their active domains.

In this paper we propose an algorithm Soft-Impute for the nuclear norm regularized least-squares problem (3) that scales to large problems with $m, n \approx 10^5$–10^6 with around 10^4–10^5 or more observed entries. At every iteration Soft-Impute decreases the value

of the objective function towards its minimum, and at the same time gets closer to the set of optimal solutions of the problem 2. We study the convergence properties of this algorithm and discuss how it can be extended to other more sophisticated forms of spectral regularization.

To summarize some performance results[1]:

- We obtain a rank-11 solution to (2) for a problem of size $(5 \times 10^5) \times (5 \times 10^5)$ and $|\Omega| = 10^4$ observed entries in under 11 minutes.
- For the same sized matrix with $|\Omega| = 10^5$ we obtain a rank-52 solution in under 80 minutes.
- For a $10^6 \times 10^6$ sized matrix with $|\Omega| = 10^5$ a rank-80 solution is obtained in approximately 2.5 hours.
- We fit a rank-40 solution for the Netflix data in 6.6 hours. Here there are 10^8 observed entries in a matrix with 4.8×10^5 rows and 1.8×10^4 columns. A rank 60 solution takes 9.7 hours.

The paper is organized as follows. In Section 2, we discuss related work and provide some context for this paper. In Section 3 we introduce the SOFT-IMPUTE algorithm and study its convergence properties. The computational aspects of the algorithm are described in Section 4, and Section 5 discusses how nuclear norm regularization can be generalized to more aggressive and general types of spectral regularization. Section 6 describes post-processing of "selectors" and initialization. We discuss simulations and experimental studies in Section 7 and application to the Netflix data in Section 8.

2. Context and related work

[CT09, CCS08, CR08] consider the criterion

$$\begin{aligned} \text{minimize} \quad & \|Z\|_* \\ \text{subject to} \quad & Z_{ij} = X_{ij},\ \forall (i,j) \in \Omega \end{aligned} \tag{4}$$

With $\delta = 0$, the criterion (1) is equivalent to (4), in that it requires the training error to be zero. Cai et. al. [CCS08] propose a first-order singular-value-thresholding algorithm SVT scalable to large matrices for the problem (4). They comment on the problem (2) with $\delta > 0$, but dismiss it as being computationally prohibitive for large problems.

We believe that (4) will almost always be too rigid and will result in overfitting. If minimization of prediction error is an important goal, then the optimal solution Z^* will typically lie somewhere in the interior of the path indexed by δ (Figures 1,2, and 3).

In this paper we provide an algorithm for computing solutions of (3) on a grid of λ values, based on warm restarts. The algorithm is inspired by Hastie et al.'s SVD- impute [HTS+99, TCS+01], and is very different from the proximal forward-backward splitting method of [CCS08, CW05] as well as the Bregman iterative method proposed in [MGC09].

1. all times are reported based on computations done in a Intel Xeon Linux 3GHz processor using MATLAB, with no C or Fortran interlacing

The latter is motivated by an analogous algorithm used for the ℓ_1 penalized least-squares problem. All these algorithms [CCS08, CW05, MGC09] require the specification of a step size, and can be quite sensitive to the chosen value. Our algorithm does not require a step-size, or any such parameter.

In [MGC09] the SVD step becomes prohibitive, so randomized algorithms are used in the computation. Our algorithm SOFT-IMPUTE also requires an SVD computation at every iteration, but by exploiting the problem structure, can easily handle matrices of dimensions much larger than those in [MGC09]. At each iteration the non-sparse matrix has the structure:

$$Y = Y_{SP} \text{ (Sparse)} \quad + \quad Y_{LR} \text{ (Low Rank)} \tag{5}$$

In (5) Y_{SP} has the same sparsity structure as the observed X, and Y_{LR} has the rank $r \ll m, n$ of the estimated Z. For large scale problems, we use iterative methods based on Lanczos bidiagonalization with partial re-orthogonalization (as in the PROPACK algorithm [Lar98]), for computing the first r singular vectors/values of Y. Due to the specific structure of (5), multiplication by Y and Y' can both be achieved in a cost-efficient way. More precisely, in the sparse + low-rank situation, the computationally burdensome work in computing the SVD is of an order that depends linearly on the matrix dimensions — $O((m+n)r)+O(|\Omega|)$. In our experimental studies we find that our algorithm converges in very few iterations; with warm-starts the entire regularization path can be computed very efficiently along a dense series of values for λ.

Although the nuclear norm is motivated here as a convex relaxation to a rank constraint, we believe in many situations it will outperform the rank-restricted estimator. This is supported by our experimental studies and explored in [SAJ05, RS05]. We draw the natural analogy with model selection in linear regression, and compare best-subset regression (ℓ_0 regularization) with the LASSO (ℓ_1 regularization)[Tib96, HTF09]. There too the ℓ_1 penalty can be viewed as a convex relaxation of the ℓ_0 penalty. But in many situations with moderate sparsity, the LASSO will outperform best subset in terms of prediction accuracy [Fri08, HTF09]. By shrinking the parameters in the model (and hence reducing their variance), the lasso permits more parameters to be included. The nuclear norm is the ℓ_1 penalty in matrix completion, as compared to the ℓ_0 rank. By shrinking the singular values, we allow more dimensions to be included without incurring undue estimation variance.

Another class of techniques used in collaborative filtering problems are close in spirit to (2). These are known as *maximum margin factorization* methods, and use a factor model for the matrix Z [SRJ05]. Let $Z = UV'$ where $U_{m\times r}$ and $V_{n\times r}$ (U, V are not orthogonal), and consider the following problem

$$\underset{U,V}{\text{minimize}} \quad \sum_{(i,j)\in\Omega} (X_{ij} - (UV')_{ij})^2 + \lambda(\|U\|_F^2 + \|V\|_F^2). \tag{6}$$

It turns out that (6) is equivalent to (3), since

$$||Z||_* = \underset{U,V:\ Z=UV'}{\text{minimize}} \tfrac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right) \tag{7}$$

This problem formulation and related optimization methods have been explored by [SRJ05, RS05, TPNT09]. A very similar formulation is studied in [KOM09]. However (6) is a non-convex optimization problem in (U, V). It has been observed empirically and theoretically

[BM05, RS05] that bi-convex methods used in the optimization of (6) get stuck in sub-optimal local minima if the rank r is small. For a large number of factors r and large dimensions m, n the computational cost may be quite high [RS05]. Moreover the factors (U, V) are not orthogonal, and if this is required, additional computations are required $[O(r(m+n)+r^3)]$.

Our criterion (3), on the other hand, is convex in Z for every value of λ (and hence rank r) and it outputs the solution $\hat{Z}$ in the form of its SVD, implying that the "factors" U, V are already orthogonal. Additionally the formulation (6) has two different tuning parameters r and λ, both of which are related to the rank or spectral properties of the matrices U, V. Our formulation has only one tuning parameter λ. The presence of two tuning parameters is problematic:

- It results in a significant increase in computational burden, since for every given value of r, one needs to compute an entire system of solutions by varying λ.
- In practice when neither the optimal values of r and λ are known, a two-dimensional search (eg by cross validation) is required to select suitable values.

3. Algorithm and Convergence analysis

3.1 Notation

We adopt the notation of [CCS08]. Define a matrix $P_\Omega(Y)$ (with dimension $n \times m$)

$$P_\Omega(Y)\ (i,j) = \begin{cases} Y_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega, \end{cases} \tag{8}$$

which is a projection of the matrix $Y_{m\times n}$ onto the observed entries. In the same spirit, define the complementary projection $P_\Omega^\perp(Y)$ via $P_\Omega^\perp(Y) + P_\Omega(Y) = Y$. Using (8) we can rewrite $\sum_{(i,j)\in\Omega}(X_{ij} - Z_{ij})^2$ as $\|P_\Omega(X) - P_\Omega(Z)\|_F^2$.

3.2 Nuclear norm regularization

We present the following lemma, given in [CCS08], which forms a basic ingredient in our algorithm.

Lemma 1. *Suppose the matrix $W_{m\times n}$ has rank r. The solution to the optimization problem*

$$\underset{Z}{\text{minimize}} \quad \tfrac{1}{2}\|W - Z\|_F^2 + \lambda\|Z\|_* \tag{9}$$

is given by $\hat{Z} = \mathbf{S}_\lambda(W)$ where

$$\mathbf{S}_\lambda(W) \equiv UD_\lambda V' \quad \textit{with} \quad D_\lambda = \operatorname{diag}\left[(d_1-\lambda)_+, \ldots, (d_r-\lambda)_+\right], \tag{10}$$

UDV' is the SVD of W, $D = \operatorname{diag}[d_1, \ldots, d_r]$, and $t_+ = \max(t, 0)$.

The notation $\mathbf{S}_\lambda(W)$ refers to *soft-thresholding* [DJKP95]. Lemma 1 appears in [CCS08, MGC09] where the proof utilizes the sub-gradient characterization of the nuclear norm. In Appendix A.1 we present an entirely different proof, which can be extended in a relatively straightforward way to other complicated forms of spectral regularization discussed in Section 5. Our proof is followed by a remark that covers these more general cases.

3.3 Algorithm

Using the notation in 3.1, we rewrite (3)

$$\underset{Z}{\text{minimize}} \quad \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_*. \tag{11}$$

Let $f_\lambda(Z) = \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_*$ denote the objective in (11).

We now present Algorithm 1—SOFT-IMPUTE—for computing a series of solutions to (11) for different values of λ using warm starts.

Algorithm 1 SOFT-IMPUTE

1. Initialize $Z^{\text{old}} = 0$ and create a decreasing grid Λ of values $\lambda_1 > \ldots > \lambda_K$.
2. For every fixed $\lambda = \lambda_1,\ \lambda_2, \ldots \in \Lambda$ iterate till convergence:
   (a) Compute $Z^{\text{new}} \leftarrow \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z^{\text{old}}))$
   (b) If $\frac{\|Z^{\text{new}} - Z^{\text{old}}\|_F^2}{\|Z^{\text{old}}\|_F^2} < \epsilon$, go to step 2d.
   (c) Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$ and go to step 2a.
   (d) Assign $\hat{Z}_\lambda \leftarrow Z^{\text{new}}$ and go to step 2.
3. Output the sequence of solutions $\hat{Z}_{\lambda_1}, \ldots, \hat{Z}_{\lambda_K}$.

The algorithm repeatedly replaces the missing entries with the current guess, and then updates the guess by solving (9). Figures 1, 2 and 3 show some examples of solutions using SOFT-IMPUTE (blue curves). We see test and training error in the top rows as a function of the nuclear norm, obtained from a grid of values Λ. These error curves show a smooth and very competitive performance.

3.4 Convergence analysis

In this section we study the convergence properties of Algorithm 1. We prove that SOFT-IMPUTE converges to the solution to (11). It is an iterative algorithm that produces a sequence of solutions for which the criterion decreases to the optimal solution with every iteration. This aspect is absent in many first order convex minimization algorithms [Boy08]. In addition the successive iterates get closer to the optimal set of solutions of the problem 2. Unlike many other competitive first-order methods [CCS08, CW05, MGC09], SOFT-IMPUTE does not involve the choice of any step-size. Most importantly our algorithm is readily scalable for solving large scale semidefinite programming problems (2,11) as will be explained later in Section 4.

For an arbitrary matrix $\tilde{Z}$, define

$$Q_\lambda(Z|\tilde{Z}) = \tfrac{1}{2}\|P_\Omega(X) + P_\Omega^\perp(\tilde{Z}) - Z\|_F^2 + \lambda\|Z\|_*, \tag{12}$$

a surrogate of the objective function $f_\lambda(z)$. Note that $f_\lambda(\tilde{Z}) = Q_\lambda(\tilde{Z}|\tilde{Z})$ for any $\tilde{Z}$.

Lemma 2. *For every fixed $\lambda \geq 0$, define a sequence Z_λ^k by*

$$Z_\lambda^{k+1} = \arg\min_Z Q_\lambda(Z|Z_\lambda^k) \tag{13}$$

with any starting point Z_λ^0. The sequence Z_λ^k satisfies

$$f_\lambda(Z_\lambda^{k+1}) \leq Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^k) \leq f_\lambda(Z_\lambda^k) \tag{14}$$

Proof. Note that

$$Z_\lambda^{k+1} = \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)) \tag{15}$$

by Lemma 1 and the definition (12) of $Q_\lambda(Z|Z_\lambda^k)$.

$$\begin{aligned}
f_\lambda(Z_\lambda^k) &= Q_\lambda(Z_\lambda^k|Z_\lambda^k) \\
&= \tfrac{1}{2}\|P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) - Z_\lambda^k\|_F^2 + \lambda\|Z_\lambda^k\|_* \\
&\geq \min_Z \tfrac{1}{2}\left\{\|P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) - Z\|_F^2\right\} + \lambda\|Z\|_* \\
&= Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^k) \\
&= \tfrac{1}{2}\|\left\{P_\Omega(X) - P_\Omega(Z_\lambda^{k+1})\right\} + \left\{P_\Omega^\perp(Z_\lambda^k) - P_\Omega^\perp(Z_\lambda^{k+1})\right\}\|_F^2 + \lambda\|Z_\lambda^{k+1}\|_* \\
&= \tfrac{1}{2}\left\{\|P_\Omega(X) - P_\Omega(Z_\lambda^{k+1})\|_F^2 + \|P_\Omega^\perp(Z_\lambda^k) - P_\Omega^\perp(Z_\lambda^{k+1})\|_F^2\right\} + \lambda\|Z_\lambda^{k+1}\|_* \quad (16) \\
&\geq \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(Z_\lambda^{k+1})\|_F^2 + \lambda\|Z_\lambda^{k+1}\|_* \quad (17) \\
&= Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^{k+1}) \\
&= f(Z_\lambda^{k+1})
\end{aligned}$$

□

Lemma 3. *The nuclear norm shrinkage operator $\mathbf{S}_\lambda(\cdot)$ satisfies the following for any W_1, W_2 (with matching dimensions)*

$$\|\mathbf{S}_\lambda(W_1) - \mathbf{S}_\lambda(W_2)\|_F^2 \leq \|W_1 - W_2\|_F^2 \tag{18}$$

In particular this implies that $\mathbf{S}_\lambda(W)$ is a continuous map in W.

Lemma 3 is proved in [MGC09]; their proof is complex and based on trace inequalities. We give a concise proof in Appendix A.2.

Lemma 4. *The successive differences $\|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2$ of the sequence Z_λ^k are monotone decreasing:*

$$\|Z_\lambda^{k+1} - Z_\lambda^k\|_F^2 \leq \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2 \ \forall k \tag{19}$$

Moreover the difference sequence converges to zero. That is

$$Z_\lambda^{k+1} - Z_\lambda^k \to 0 \text{ as } k \to \infty$$

The proof of Lemma 4 is given in Appendix A.3.

Lemma 5. *Every limit point of the sequence Z_λ^k defined in Lemma 2 is a stationary point of*

$$\tfrac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_* \tag{20}$$

Hence it is a solution to the fixed point equation

$$Z = \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z)) \tag{21}$$

The proof of Lemma 5 is given in Appendix A.4.

Theorem 1. *The sequence Z_λ^k defined in Lemma 2 converges to a limit Z_λ^∞ that solves*

$$\underset{Z}{\text{minimize}} \quad \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_* \tag{22}$$

Proof. It suffices to prove that Z_λ^k converges; the theorem then follows from Lemma 5.

Let $\hat{Z}_\lambda$ be a limit point of the sequence Z_λ^k. There exists a subsequence m_k such that $Z_\lambda^{m_k} \to \hat{Z}_\lambda$. By Lemma 5, $\hat{Z}_\lambda$ solves the problem (22) and satisfies the fixed point equation (21).

Hence

$$\begin{aligned}
\|\hat{Z}_\lambda - Z_\lambda^k\|_F^2 &= \|\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(\hat{Z}_\lambda)) - \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2 &(23)\\
&\leq \|(P_\Omega(X) + P_\Omega^\perp(\hat{Z}_\lambda)) - (P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2\\
&= \|P_\Omega^\perp(\hat{Z}_\lambda - Z_\lambda^{k-1})\|_F^2\\
&\leq \|\hat{Z}_\lambda - Z_\lambda^{k-1}\|_F^2 &(24)
\end{aligned}$$

In (23) two substitutions were made; the left one using (21) in Lemma 5, the right one using (15). Inequality (24) implies that the sequence $\|\hat{Z}_\lambda - Z_\lambda^{k-1}\|_F^2$ converges as $k \to \infty$. To show the convergence of the sequence Z_λ^k it suffices to prove that the sequence $\hat{Z}_\lambda - Z_\lambda^k$ converges to zero. We prove this by contradiction.

Suppose the sequence Z_λ^k has another limit point $Z_\lambda^+ \neq \hat{Z}_\lambda$. Then $\hat{Z}_\lambda - Z_\lambda^k$ has two distinct limit points 0 and $Z_\lambda^+ - \hat{Z}_\lambda \neq 0$. This contradicts the convergence of the sequence $\|\hat{Z}_\lambda - Z_\lambda^{k-1}\|_F^2$. Hence the sequence Z_λ^k converges to $\hat{Z}_\lambda := Z_\lambda^\infty$. □

The inequality in (24) implies that at every iteration Z_λ^k gets closer to an optimal solution for the problem (22)[2]. This property holds in addition to the decrease of the objective function (Lemma 2) at every iteration. This is a very nice property of the algorithm and is in general absent in many first order methods such as projected sub-gradient minimization [Boy08].

4. Computation

The computationally demanding part of Algorithm 1 is in $\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k))$. This requires calculating a low-rank SVD of a matrix, since the underlying model assumption is that $\text{rank}(Z) \ll \min\{m, n\}$. In Algorithm 1, for fixed λ, the entire sequence of matrices Z_λ^k

2. In fact this statement can be strengthened further — at every iteration the distance of the estimate decreases from the set of optimal solutions

have explicit low-rank representations of the form $U_k D_k V_k'$ corresponding to $\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))$

In addition, observe that $P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)$ can be rewritten as

$$\begin{array}{rcccc} P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) & = & \left\{P_\Omega(X) - P_\Omega(Z_\lambda^k)\right\} & + & Z_\lambda^k \\ & = & \textit{Sparse} & + & \textit{Low Rank} \end{array} \tag{25}$$

In the numerical linear algebra literature, there are very efficient direct matrix factorization methods for calculating the SVD of matrices of moderate size (at most a few thousand). When the matrix is sparse, larger problems can be solved but the computational cost depends heavily upon the sparsity structure of the matrix. In general however, for large matrices one has to resort to indirect iterative methods for calculating the leading singular vectors/values of a matrix. There is a lot research in numerical linear algebra for developing sophisticated algorithms for this purpose. In this paper we will use the PROPACK algorithm [Lar, Lar98] because of its low storage requirements, effective flop count and its well documented MATLAB version. The algorithm for calculating the truncated SVD for a matrix W (say), becomes efficient if multiplication operations Wb_1 and $W'b_2$ (with $b_1 \in \Re^n,\ b_2 \in \Re^m$) can be done with minimal cost.

Algorithm Soft-Impute requires repeated computation of a truncated SVD for a matrix W with structure as in (25). Note that in (25) the term $P_\Omega(Z_\lambda^k)$ can be computed in $O(|\Omega|r)$ flops using only the required outer products (i.e. our algorithm does not compute the matrix explicitly).

The cost of computing the truncated SVD will depend upon the cost in the operations Wb_1 and $W'b_2$ (which are equal). For the sparse part these multiplications cost $O(|\Omega|)$. Although it costs $O(|\Omega|r)$ to create the matrix $P_\Omega(Z_\lambda^k)$, this is used for each of the r such multiplications (which also cost $O(|\Omega|r)$), so we need not include that cost here. The LowRank part costs $O((m+n)r)$ for the multiplication by b_1. Hence the cost is $O(|\Omega|) + O((m+n)r)$ per vector multiplication.

For the reconstruction problem to be theoretically meaningful in the sense of [CT09] we require that $|\Omega| \approx nr \cdot \mathrm{poly}(\log n)$. In practice often $|\Omega|$ is very small. Hence introducing the *Low Rank* part does not add any further complexity in the multiplication by W and W'. So the dominant cost in calculating the truncated SVD in our algorithm is $O(|\Omega|)$. The SVT algorithm [CCS08] for exact matrix completion (4) involves calculating the SVD of a sparse matrix with cost $O(|\Omega|)$. This implies that the computational cost of our algorithm and that of [CCS08] is the same. This order computation does not include the number of iterations required for convergence. In our experimental studies we use warm-starts for efficiently computing the entire regularization path. We find that our algorithm converges in a few iterations. Since the true rank of the matrix $r \ll \min\{m,n\}$, the computational cost of evaluating the truncated SVD (with rank $\approx r$) is linear in matrix dimensions. This justifies the large-scale computational feasibility of our algorithm.

The PROPACK package does not allow one to request (and hence compute) only the singular values larger than a threshold λ — one has to specify the number in advance. So once all the computed singular values fall above the current threshold λ, our algorithm increases the number to be computed until the smallest is smaller than λ. In large scale problems, we put an absolute limit on the maximum number.

5. Generalized spectral regularization: from soft to hard-thresholding

In Section 1 we discussed the role of the nuclear norm as a convex surrogate for the rank of a matrix, and drew the analogy with LASSO regression versus best-subset selection. We argued that in many problems ℓ_1 regularization gives better prediction accuracy [ZY06]. However, if the underlying model is very sparse, then the LASSO with its uniform shrinkage can overestimate the number of non-zero coefficients [Fri08]. It can also produce highly shrunk and hence biased estimates of the coefficients.

Consider again the problem

$$\underset{\text{rank}(Z)=k}{\text{minimize}} \|P_\Omega(X) - P_\Omega(Z)\|_F^2, \tag{26}$$

a rephrasing of (1). This best rank-k solution also solves

$$\text{minimize } \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda \sum_j I(\gamma_j(Z) > 0), \tag{27}$$

where $\gamma_j(Z)$ is the jth singular value of Z, and for a suitable choice of λ that produces a solution with rank k.

The "fully observed" matrix version of the above problem is given by the ℓ_0 version of (9) as follows:

$$\min_Z \tfrac{1}{2}\|W - Z\|_F^2 + \lambda\|Z\|_0 \tag{28}$$

where $\|Z\|_0 = \text{rank}(Z)$. The solution of (28) is given by a reduced-rank SVD of W; for every λ there is a corresponding $q = q(\lambda)$ number of singular-values to be retained in the SVD decomposition. Problem 28 is non-convex in W but its global minimizer can be evaluated. As in (10) the thresholding operator resulting from (28) is

$$\mathbf{S}_\lambda^H(W) = UD_qV' \quad \text{where} \quad D_q = \text{diag}\,(d_1, \ldots, d_q, 0, \ldots, 0) \tag{29}$$

Similar to SOFT-IMPUTE (Algorithm 1), we present below HARD-IMPUTE (Algorithm 2) for the ℓ_0 penalty.

In penalized regression there have been recent developments directed towards "bridging" the gap between the ℓ_1 and ℓ_0 penalties [Fri08, FL01, Zha07]. This is done via using concave penalties that are a better surrogate (in the sense of approximating the penalty) to ℓ_0 over the ℓ_1. They also produce less biased estimates than those produced by the ℓ_1 penalized solutions. When the underlying model is very sparse they often perform very well [Fri08, FL01, Zha07], and often enjoy superior prediction accuracy when compared to softer penalties like ℓ_1. These methods still shrink, but are less aggressive than the best-subset selection.

By analogy, we propose using a more sophisticated version of spectral regularization. This goes beyond nuclear norm regularization by using slightly more aggressive penalties that bridge the gap between ℓ_1 (nuclear norm) and ℓ_0(rank constraint). We propose minimizing

$$f_{\mathbf{p},\lambda}(Z) \quad = \quad \tfrac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda \sum_j \mathbf{p}(\gamma_j(Z); \mu) \tag{30}$$

Algorithm 2 HARD-IMPUTE

1. Create a decreasing grid Λ of values $\lambda_1 > \ldots > \lambda_K$. Initialize $\tilde{Z}_{\lambda_k}$ $k = 1, \ldots, K$ (see Section 6).
2. For every fixed $\lambda = \lambda_1,\ \lambda_2, \ldots \in \Lambda$ iterate till convergence:
   (a) Initialize $Z^{\text{old}} \leftarrow \tilde{Z}_\lambda$.
   (b) Compute $Z^{\text{new}} \leftarrow \mathbf{S}^H_\lambda(P_\Omega(X) + P^\perp_\Omega(Z^{\text{old}}))$
   (c) If $\frac{\|Z^{\text{new}} - Z^{\text{old}}\|^2_F}{Z^{\text{old}}\|^2_F} < \epsilon$, go to step 2e.
   (d) Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$ and go to step 2a.
   (e) Assign $\hat{Z}_{H,\lambda} \leftarrow Z^{\text{new}}$.
3. Output the sequence of solutions $\hat{Z}_{H,\lambda_1}, \ldots, \hat{Z}_{H,\lambda_K}$.

where $\mathbf{p}(|t|; \mu)$ is concave in $|t|$. The parameter $\mu \in [\mu_{\text{inf}}, \mu_{\text{sup}}]$ controls the degree of concavity. We may think of $p(|t|; \mu_{\text{inf}}) = |t|$ (ℓ_1 penalty) on one end and $p(|t|; \mu_{\text{sup}}) = \|t\|_0$ (ℓ_0 penalty) on the other. In particular for the ℓ_0 penalty denote $f_{\mathbf{p},\lambda}(Z)$ by $f_{H,\lambda}(Z)$ for "hard" thresholding. See [Fri08, FL01, Zha07] for examples of such penalties.

In Remark 1 in Appendix A.1 we argue how the proof can be modified for general types of spectral regularization. Hence for minimizing the objective (30) we will look at the analogous version of (9, 28) which is

$$\min_Z \tfrac{1}{2}\|W - Z\|^2_F + \lambda \sum_j \mathbf{p}(\gamma_j(Z); \mu) \tag{31}$$

The solution is given by a thresholded SVD of W:

$$\mathbf{S}^{\mathbf{p}}_\lambda(W) = U D_{\mathbf{p},\lambda} V' \tag{32}$$

Where $D_{\mathbf{p},\lambda}$ is a entry-wise thresholding of the diagonal entries of the matrix D consisting of singular values of the matrix W. The exact form of the thresholding depends upon the form of the penalty function $\mathbf{p}(\cdot;\cdot)$, as discussed in Remark 1. Algorithm 1 and Algorithm 2 can be modified for the penalty $\mathbf{p}(\cdot;\mu)$ by using a more general thresholding function $\mathbf{S}^{\text{p}}_\lambda(\cdot)$ in Step 2b. The corresponding step becomes:

$$Z^{\text{new}} \leftarrow \mathbf{S}^p_\lambda(P_\Omega(X) + P^\perp_\Omega(Z^{\text{old}}))$$

However these types of spectral regularization make the criterion (30) non-convex and hence it becomes difficult to optimize globally.

6. Post-processing of "selectors" and initialization

Because the ℓ_1 norm regularizes by shrinking the singular values, the number of singular values retained (through cross-validation, say) may exceed the actual rank of the matrix. In

such cases it is reasonable to *undo* the shrinkage of the chosen models, which might permit a lower-rank solution.

If Z_λ is the solution to (11), then its *post-processed* version Z_λ^u obtained by "unshrinking" the eigen-values of the matrix Z_λ is obtained by

$$\begin{aligned} \boldsymbol{\alpha} &= \underset{\alpha_i \geq 0,\, i=1,\ldots,r_\lambda}{\arg\min} \quad \|P_\Omega(X) - \sum_{i=1}^{r_\lambda} \alpha_i P_\Omega(u_i v_i')\|^2 \qquad (33) \\ Z_\lambda^u &= UD_\alpha V', \end{aligned}$$

where $D_\alpha = \mathrm{diag}(\alpha_1, \ldots, \alpha_{r_\lambda})$. Here r_λ is the rank of Z_λ and $Z_\lambda = UD_\lambda V'$ is its SVD. The estimation in (33) can be done via ordinary least squares, which is feasible because of the sparsity of $P_\Omega(u_i v_i')$ and that r_λ is small.[3] If the least squares solutions $\boldsymbol{\alpha}$ do not meet the positivity constraints, then the negative sign can be absorbed into the corresponding singular vector.

Rather than estimating a diagonal matrix D_α as above, one can insert a matrix $M_{r_\lambda \times r_\lambda}$ between U and V above to obtain better training error for the same rank [KOM09]. Hence given U, V (each of rank r_λ) from the Soft-Impute algorithm, we solve

$$\begin{aligned} \hat{M} &= \underset{M}{\arg\min} \quad \|P_\Omega(X) - P_\Omega(UMV')\|^2 \qquad (34) \\ \hat{Z}_\lambda &= U\hat{M}V' \end{aligned}$$

The objective function in (34) is the Frobenius norm of an affine function of M and hence can be optimized very efficiently. Scalability issues pertaining to the optimization problem (34) can be handled fairly efficiently via conjugate gradients. Criterion (34) will definitely lead to a decrease in training error as that attained by $\hat{Z} = UD_\lambda V'$ for the same rank and is potentially an attractive proposal for the original problem (1). However this heuristic cannot be caste as a (jointly) convex problem in (U, M, V). In addition, this requires the estimation of up to r_λ^2 parameters, and has the potential for overfitting. In this paper we report experiments based on (33).

In many simulated examples we have observed that this post-processing step gives a good estimate of the underlying true rank of the matrix (based on prediction error). Since fixed points of Algorithm 2 correspond to local minima of the function (30), well-chosen warm starts $\tilde{Z}_\lambda$ are helpful. A reasonable prescription for warms-starts is the nuclear norm solution via (Soft-Impute), or the post processed version (33). The latter appears to significantly speed up convergence for Hard-Impute. This observation is based on our simulation studies.

7. Simulation Studies

In this section we study the training and test errors achieved by the estimated matrix by our proposed algorithms and those by [CCS08, KOM09]. The reconstruction algorithm (OptSpace) described in [KOM09] considers criterion (1) (in the presence of noise). It uses the representation $Z = USV'$ (which need not correspond to the SVD). For every fixed rank

3. Observe that the $P_\Omega(u_i v_i')$, $i = 1, \ldots, r_\lambda$ are not orthogonal, though the $u_i v_i'$ are.

r OptSpace uses a two-stage minimization procedure: firstly on S and then on U, V (in a Grassmann manifold) for computing a rank-r decomposition $\hat{Z} = \hat{U}\hat{S}\hat{V}'$ It uses a suitable starting point obtained by performing a sparse SVD on a *clean* version of the observed matrix $P_\Omega(X)$. This is similar to the formulation of maximum margin factorization (MMF) (6) as outlined in Section 1, without the Frobenius norm regularization on the components U, V.

To summarize, we look at the performance of the following methods:

- (a) Soft-Impute (algorithm 1); (b) Pp-SI Post-processing on the output of Algorithm 1, (c) Hard-Impute (Algorithm 2) starting with the output of (b).
- SVT algorithm by [CCS08]
- OptSpace reconstruction algorithm by [KOM09]

In all our simulation studies we took the underlying model as $Z_{m\times n} = U_{m\times r}V'_{r\times n} + \text{noise}$; where U and V are random matrices with standard normal Gaussian entries, and noise is i.i.d. Gaussian. Ω is uniformly random over the indices of the matrix with $p\%$ percent of missing entries. These are the models under which the coherence conditions hold true for the matrix completion problem to be meaningful as pointed out in [CT09, KOM09]. The signal to noise ratio for the model and the test-error (standardized) are defined as

$$\text{SNR} = \sqrt{\frac{\text{var}(UV')}{\text{var}(\text{noise})}}; \quad \text{testerror} = \frac{\|P_\Omega^\perp(UV' - \hat{Z})\|_F^2}{\|P_\Omega^\perp(UV')\|_F^2} \tag{35}$$

Training error (standardized) is defined as

$$\frac{|P_\Omega(Z - \hat{Z})\|_F^2}{\|P_\Omega(Z)\|_F^2}, \tag{36}$$

the fraction of the error explained on the observed entries by the estimate relative to a zero estimate.

Figures 1,2, and 3 show training and test error for all the algorithms mentioned above — both as a function of nuclear norm and rank — for the three problem instances. The results displayed in the figures are averaged over 50 simulations, and also show one-standard-error bands (hardly visible). Since OptSpace only uses rank, it is excluded from the top panels. In all examples $(m, n) = (100, 100)$. SNR, true rank and percentage of missing entries are indicated in the figures. There is a unique correspondence between λ and nuclear norm. The plots vs the rank indicate how effective the nuclear norm is as a rank approximation — that is whether it recovers the true rank while minimizing prediction error.

For SVT we use the MATLAB implementation of the algorithm downloaded from the second author's [CCS08] webpage. For OptSpace we use the MATLAB implementation of the algorithm as obtained from the third author's webpage [KOM09].

7.1 Observations

In Type a, the SNR= 1, fifty percent of entries are missing and the true underlying rank is ten. The performances of Pp-SI and Soft-Impute are clearly better than the rest. The

solution of SVT recovers a matrix with a rank much larger than the true rank. The SVT also has very poor prediction error, suggesting once again that exactly fitting the training data is far too rigid. SOFT-IMPUTE recovers an optimal rank (corresponding to the minima of the test error curve) which is larger than the true rank of the matrix, but the prediction error is very competitive. PP-SI estimates the right rank of the matrix based on the minima of the prediction error curve. This seems to be the only algorithm to do so in this example. Both HARD-IMPUTE and OPTSPACE perform very poorly in test error. This is a high noise situation, so the HARD-IMPUTE is too aggressive in selecting the singular vectors from the observed entries and hence ends up reaching a very sub-optimal subspace. The training errors of PP-SI and HARD-IMPUTE are smaller than that achieved by the SOFT-IMPUTE solution for a fixed rank along the regularization path. This is expected by the very method of construction. However this deteriorates the test error performance of the HARD-IMPUTE, at the same rank. The nuclear norm may not give very good training error at a certain rank (in the sense it has strong competitors), but this trade off is compensated in the better prediction error it achieves. Though the nuclear norm is often viewed as a surrogate for the rank of a matrix, we see in these examples that it can provide a superior mechanism for regularization. This is similar to the performance of LASSO in the context of regression. Although the LASSO penalty can be viewed as a convex surrogate for the ℓ_0 penalty in model selection, its ℓ_1 penalty provides a smoother and often better basis for regularization.

In Type b, the SNR= 1, fifty percent of entries are missing and the true underlying rank is six. OPTSPACE performs poorly in test error. HARD-IMPUTE performs worse than the PP-SI and SOFT-IMPUTE, but is pretty competitive near the true rank of the matrix. In this example however the PP-SI is the best in test error and nails the right rank of the matrix. Based on the above two example we observe that in high noise models HARD-IMPUTE and OPTSPACE behave very similarly.

In Type-c the SNR= 10, the noise is relatively small as compared to the other two cases. The true underlying rank is 5, but the proportion of missing entries is much higher around eighty percent. Test errors of both PP-SI and SOFT-IMPUTE are found to decrease till a large nuclear norm after which they become roughly the same, suggesting no further impact of regularization. OPTSPACE performs well in this example getting a sharp minima at the true rank of the matrix. This good behavior of the latter as compared to the previous two instances is because the SNR is very high. HARD-IMPUTE however shows the best performance in this example. The better performance of both OPTSPACE and HARD-IMPUTE over SOFT-IMPUTE is because the true underlying rank of the matrix is very small. This is reminiscent of better predictive performance of best-subset or concave penalized regression over LASSO in set-ups where the underlying model is very sparse [Fri08].

In addition we performed some large scale simulations in Table 1 for our algorithm in different problem sizes. The problem dimensions, SNR and time in seconds are reported. All computations are done in MATLAB and the MATLAB implementation of PROPACK is used.

8. Application to Netflix data

In this section we report briefly on the application of our proposed methods to the Netflix movie prediction contest. The training data consists of the ratings of 17,770 movies by

(m, n)	$\|\Omega\|$	true rank	SNR	effective rank	time(s)
$(3 \times 10^4, 10^4)$	10^4	15	1	$(13, 47, 80)$	$(41.9, 124.7, 305.8)$
$(10^5, 10^5)$	10^4	15	10	$(5, 14, 32, 62)$	$(37, 74.5, 199.8, 653)$
$(10^5, 10^5)$	10^5	15	10	$(18, 80)$	$(202, 1840)$
$(5 \times 10^5, 5 \times 10^5)$	10^4	15	10	11	628.14
$(5 \times 10^5, 5 \times 10^5)$	10^5	15	1	$(3, 11, 52)$	$(341.9, 823.4, 4810.75)$
$(10^6, 10^6)$	10^5	15	1	80	8906

Table 1: Performance of the SOFT-IMPUTE on different problem instances. Effective rank is the rank of the recovered matrix at value of λ for (11). Convergence criterion is taken as "fraction of improvement of objective value" less than 10^{-4}. All implementations are done in MATLAB including the MATLAB implementation of PROPACK on a Intel Xeon Linux 3GHz processor. Timings (in seconds) are to be interpreted keeping the MATLAB implementation in mind.

480,189 Netflix customers. The data matrix is extremely sparse, with 100,480,507 or 1% of the entries observed. The task is to predict the unseen ratings for a qualifying set and a test set of about 1.4 million ratings each, with the true ratings in these datasets held in secret by Netflix. A probe set of about 1.4 million ratings is distributed to participants, for calibration purposes. The movies and customers in the qualifying, test and probe sets are all subsets of those in the training set.

The ratings are integers from 1 (poor) to 5 (best). Netflix's own algorithm has an RMSE of 0.9525 and the contest goal is to improve this by 10%, or an RMSE of 0.8572. The contest has been going for almost 3 years, and the leaders have recently passed the 10% improvement threshold and may soon be awarded the grand prize. Many of the leading algorithms use the SVD as a starting point, refining it and combining it with other approaches. Computation of the SVD on such a large problem is prohibitive, and many researchers resort to approximations such as subsampling (see e.g. [RMH07]). Here we demonstrate that our spectral regularization algorithm can be applied to entire Netflix training set (the Probe dataset has been left outside the training set) with a reasonable computation time.

We removed the movie and customer means, and then applied HARD-IMPUTE with varying ranks. The results are shown in Table 2.

These results are not meant to be competitive with the best results obtained by the leading groups, but rather just demonstrate the feasibility of applying HARD-IMPUTE to such a large dataset. In addition, it may be mentioned here that the objective criterion as in Algorithm 1 or Algorithm 2 is known to have optimal generalization error or reconstruction error under the assumption that the structure of missing-ness is approximately

rank	time (hrs)	train error	RMSE
20	3.3	0.217	0.986
30	5.8	0.203	0.977
40	6.6	0.194	0.965
60	9.7	0.181	0.966

Table 2: Results of applying HARD-IMPUTE to the Netflix data. The computations were done on a Intel Xeon Linux 3GHz processor; timings are reported based on MATLAB implementations of PROPACK and our algorithm. RMSE is the root mean squared error over the probe set. "train error" is the proportion of error on the observed dataset achieved by our estimator relative to the zero estimator.

uniform [CT09, SAJ05, CR08, KOM09]. This assumption is definitely not true for the Netflix data due to the high imbalance in the degree of missingness. As we saw in the simulated examples, for small SNR HARD-IMPUTE performs pretty poorly in prediction error as compared to SOFT-IMPUTE; the Netflix data is likely to be very noisy. These provide some explanations for the RMSE values obtained in our results and suggest possible directions for modifications and improvements to achieve further improvements in prediction error.

ACKNOWLEDGEMENTS

We thank Emmanuel Candes, Andrea Montanari, Stephen Boyd and Nathan Srebro for helpful discussions. Trevor Hastie was partially supported by grant DMS-0505676 from the National Science Foundation, and grant 2R01 CA 72028-07 from the National Institutes of Health. Robert Tibshirani was partially supported from National Science Foundation Grant DMS-9971405 and National Institutes of Health Contract N01-HV-28183.

Appendix A. Appendix

A.1 Proof of Lemma 1

Proof. Let $Z = \tilde{U}_{m\times n}\tilde{D}_{n\times n}\tilde{V}'_{n\times n}$ be the SVD of Z. Assume WLOG $m \geq n$. We will explicitly evaluate the closed form solution of the problem (9).

$$\tfrac{1}{2}\|Z - W\|_F^2 + \lambda\|Z\|_* = \tfrac{1}{2}\left\{\|Z\|_F^2 - 2\sum_{i=1}^{n}\tilde{d}_i\tilde{u}_i'W\tilde{v}_i + \sum_{i=1}^{n}\tilde{d}_i^2\right\} + \lambda\sum_{i=1}^{n}\tilde{d}_i \tag{37}$$

where

$$\tilde{D} = \text{diag}\left[\tilde{d}_1, \ldots, \tilde{d}_n\right], \quad \tilde{U} = [\tilde{u}_1, \ldots, \tilde{u}_n], \quad \tilde{V} = [\tilde{v}_1, \ldots, \tilde{v}_n] \tag{38}$$

Minimizing (37) is equivalent to minimizing

$$-2\sum_{i=1}^{n}\tilde{d}_i\tilde{u}_i'W\tilde{v}_i + \sum_{i=1}^{n}\tilde{d}_i^2 + \sum_{i=1}^{n}2\lambda\tilde{d}_i; \quad \text{wrt } (\tilde{u}_i, \tilde{v}_i, \tilde{d}_i), i = 1, \ldots, n$$

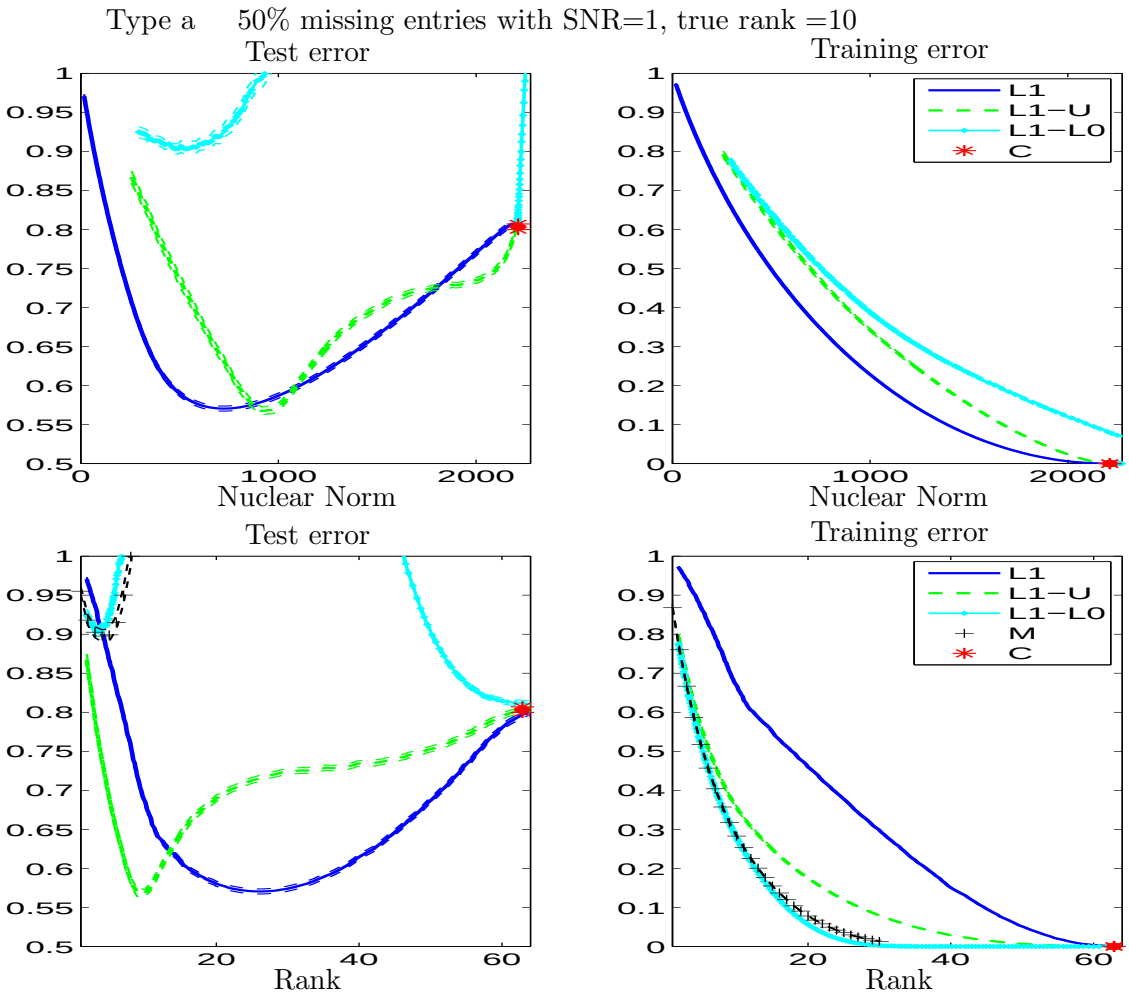

Figure 1: L1: solution for SOFT-IMPUTE; L1-U: Post processing after SOFT-IMPUTE; L1-L0 HARD-IMPUTE applied to L1-U; C : SVT algorithm; M: OPTSPACE algorithm. Both SOFT-IMPUTE and PP-SI perform well (prediction error) in the presence of noise. The latter estimates the actual rank of the matrix. Both the PP-SI and HARD-IMPUTE perform better than SOFT-IMPUTE in training error for the same rank or nuclear norm. HARD-IMPUTE and OPTSPACE perform poorly in prediction error. SVT algorithm does very poorly in prediction error, confirming our claim that (4) causes overfitting — it recovers a matrix with high nuclear norm and rank > 60 where the true rank is only 10. Values of test error larger than one are not shown in the figure. OPTSPACE is evaluated for a series of ranks ≤ 30.

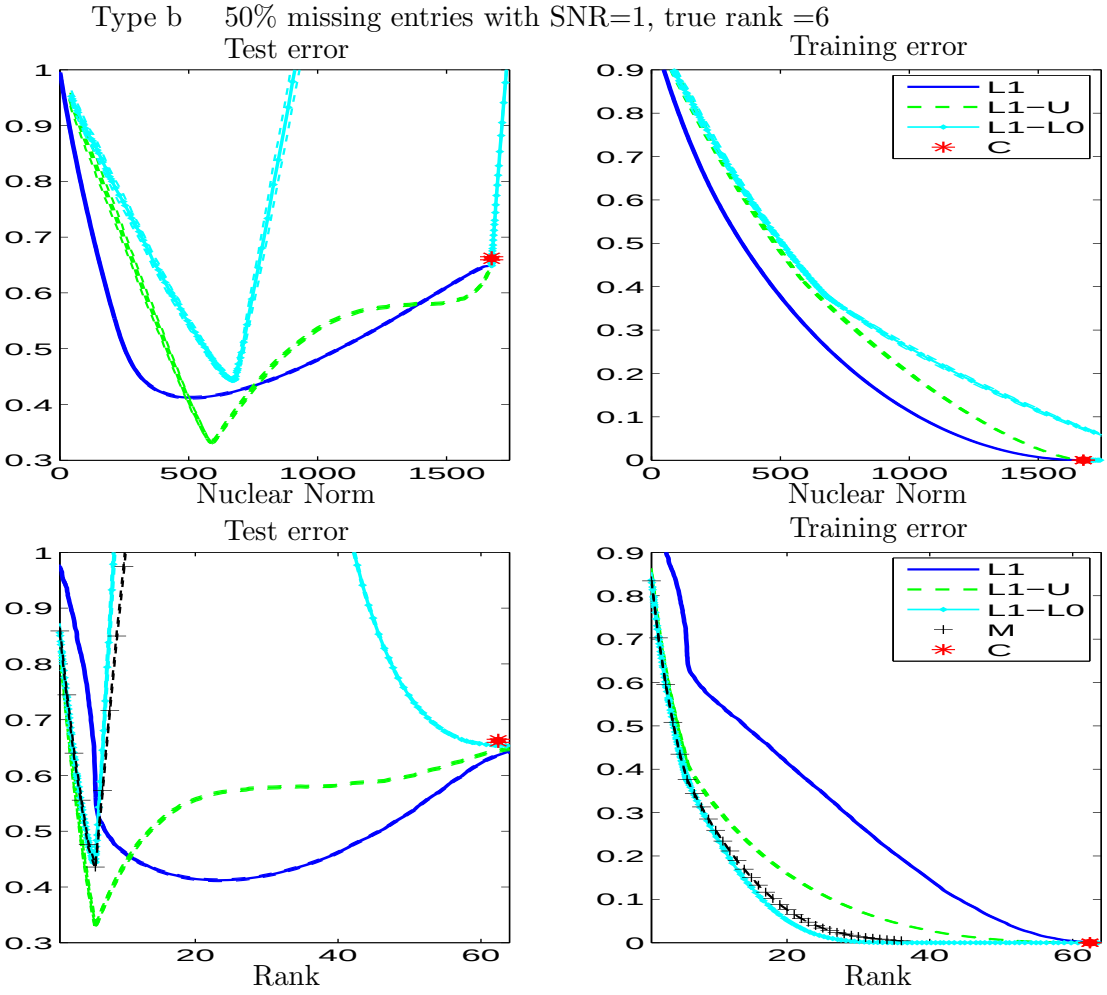

Figure 2: Pp-SI does the best in prediction error, closely followed by Soft-Impute. Both Hard-Impute, OptSpace have poor prediction error apart from near the true rank of the matrix ie 6 where they show reasonable performance. SVT algorithm does very poorly in prediction error — it recovers a matrix with high nuclear norm and rank > 60 where the true rank is only 6. OptSpace is evaluated for a series of ranks ≤ 35.

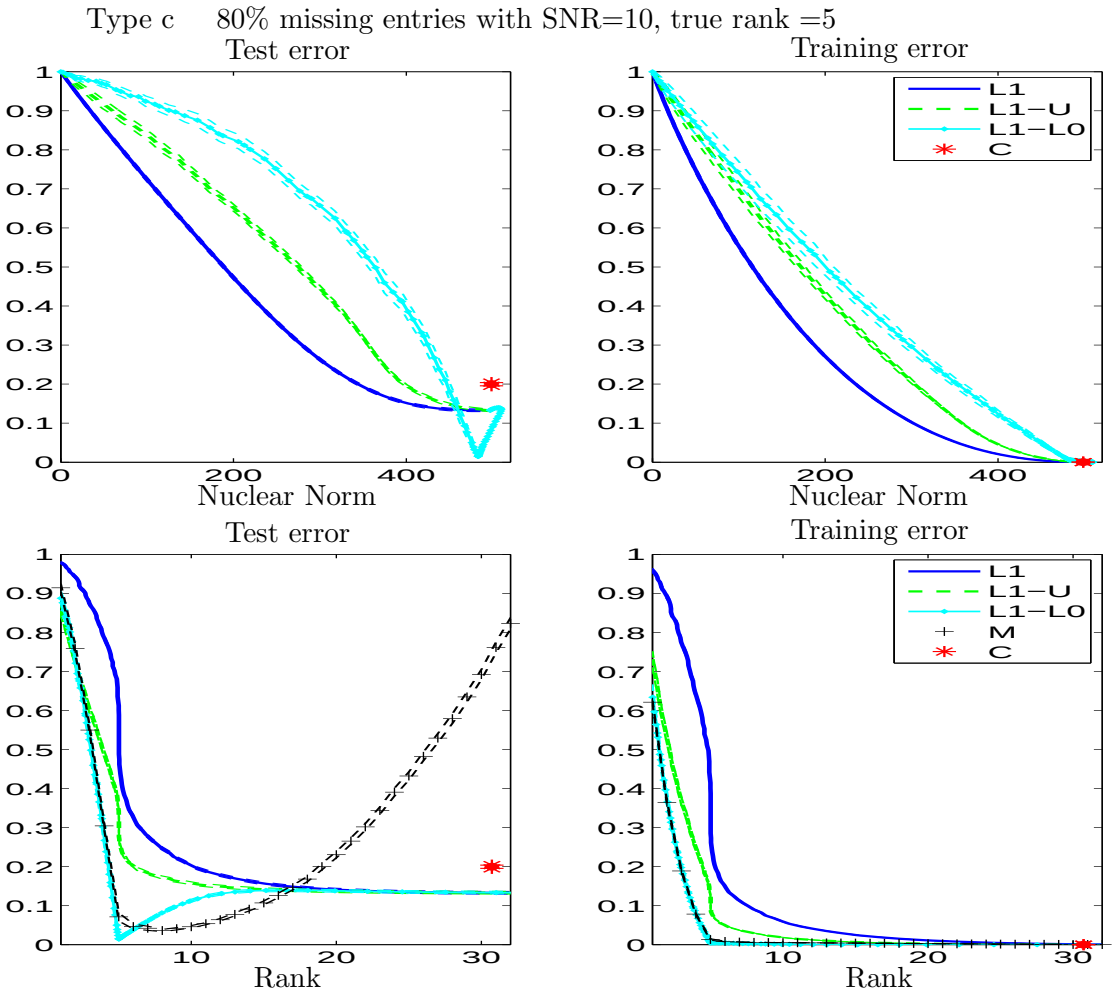

Figure 3: When the noise is low, HARD-IMPUTE can improve its performance. It gets the correct rank whereas OPTSPACE overestimates it. HARD-IMPUTE performs the best here wrt prediction error, followed by OPTSPACE. The latter does better than SOFT-IMPUTE. The noise here is low, still the SVT recovers a matrix with high rank approximately 30 and has poor prediction error as well. The test error of the SVT is found to be different from the limiting solution of SOFT-IMPUTE, though the former is allowed to run for 1000 iterations for convergence. This suggests that for small fluctuations of the objective criteria (11,2) around the minima the estimated "optimal solution" is not robust.

under the constraints $\tilde{U}'\tilde{U} = I_n$, $\tilde{V}'\tilde{V} = I_n$ and $\tilde{d}_i \geq 0 \, \forall i$.

Observe the above is equivalent to minimizing (wrt $\tilde{U}, \tilde{V}$) the function $Q(\tilde{U}, \tilde{V})$

$$Q(\tilde{U}, \tilde{V}) = \min_{\tilde{D} \geq 0} \quad \tfrac{1}{2} \left\{ -2 \sum_{i=1}^{n} \tilde{d}_i \tilde{u}_i' W \tilde{v}_i + \sum_{i=1}^{n} \tilde{d}_i^2 \right\} + \lambda \sum_{i=1}^{n} \tilde{d}_i \tag{39}$$

Since the objective to be minimized wrt $\tilde{D}$ (39) is separable in $\tilde{d}_i, i = 1, \ldots, n$ it suffices to minimize it wrt each $\tilde{d}_i$ separately.

The problem

$$\underset{\tilde{d}_i \geq 0}{\text{minimize}} \, \tfrac{1}{2} \left\{ -2 \tilde{d}_i \tilde{u}_i' W \tilde{v}_i + \tilde{d}_i^2 \right\} + \lambda \tilde{d}_i \tag{40}$$

can be solved looking at the stationary conditions of the function using its sub-gradient [Boy08]. The solution of the above problem is given by $S_\lambda(\tilde{u}_i' W \tilde{v}_i) = (\tilde{u}_i' W \tilde{v}_i - \lambda)_+$ the soft-thresholding of $\tilde{u}_i' W \tilde{v}_i$[4]. More generally the soft-thresholding operator [FHHT07, HTF09] is given by $S_\lambda(x) = \text{sgn}(x)(|x| - \lambda)_+$. See [FHHT07] for more elaborate discussions on how the soft-thresholding operator arises in univariate penalized least-squares problems with the ℓ_1 penalization.

Plugging the values of optimal $\tilde{d}_i, i = 1, \ldots, n$; obtained from (40) in (39) we get

$$Q(\tilde{U}, \tilde{V}) = \tfrac{1}{2} \left\{ \|Z\|_F^2 - 2 \sum_{i=1}^{n} (\tilde{u}_i' W \tilde{v}_i - \lambda)_+ (\tilde{u}_i' W \tilde{v}_i - \lambda) + (\tilde{u}_i' X \tilde{v}_i - \lambda)_+^2 \right\} \tag{41}$$

Minimizing $Q(\tilde{U}, \tilde{V})$ wrt $(\tilde{U}, \tilde{V})$ is equivalent to maximizing

$$\sum_{i=1}^{n} \left\{ 2(\tilde{u}_i' W \tilde{v}_i - \lambda)_+ (\tilde{u}_i' W \tilde{v}_i - \lambda) - (\tilde{u}_i' W \tilde{v}_i - \lambda)_+^2 \right\} = \sum_{\tilde{u}_i' W \tilde{v}_i > \lambda} (\tilde{u}_i' W \tilde{v}_i - \lambda)^2 \tag{42}$$

It is a standard fact that for every i the problem

$$\underset{\|u\|_2^2 \leq 1, \|v\|_2^2 \leq 1}{\text{maximize}} \; u' W v, \;\; \text{such that} \;\; u \perp \{\hat{u}_1, \ldots, \hat{u}_{i-1}\}; \; v \perp \{\hat{v}_1, \ldots, \hat{v}_{i-1}\} \tag{43}$$

is solved by $\hat{u}_i, \hat{v}_i$, the left and right singular vectors of the matrix W corresponding to its i^{th} largest singular value. The maximum value equals the singular value. It is easy to see that maximizing the expression to the right of (42) wrt $(u_i, v_i), i = 1, \ldots, n$ is equivalent to maximizing the individual terms $\tilde{u}_i' W \tilde{v}_i$. If $r(\lambda)$ denotes the number of singular values of W larger than λ then the $(\tilde{u}_i, \tilde{v}_i), i = 1, \ldots$ that maximize the expression (42) correspond to $\left[u_1, \ldots, u_{r(\lambda)}\right]$ and $\left[v_1, \ldots, v_{r(\lambda)}\right]$; the $r(\lambda)$ left and right singular vectors of W corresponding to the largest singular values. From (40) the optimal $\tilde{D} = \text{diag}\left[\tilde{d}_1, \ldots, \tilde{d}_n\right]$ is given by $D_\lambda = \text{diag}\left[(d_1 - \lambda)_+, \ldots, (d_n - \lambda)_+\right]$.

Since the rank of W is r, the minimizer $\hat{Z}$ of (9) is given by $U D_\lambda V'$ as in (10). □

Remark 1. *For a more general spectral regularization of the form $\lambda \sum_i \mathbf{p}(\gamma_i(Z))$ (as compared to $\sum_i \lambda \gamma_i(Z)$ used above) the optimization problem (40) will be modified accordingly.*

4. WLOG we can take $\tilde{u}_i' W \tilde{v}_i$ to be non-negative

The solution of the resultant univariate minimization problem will be given by $S_\lambda^{\mathbf{p}}(\tilde{u}_i' W \tilde{v}_i)$ *for some generalized "thresholding operator"* $S_\lambda^{\mathbf{p}}(\cdot)$*, where*

$$S_\lambda^{\mathbf{p}}(\tilde{u}_i' W \tilde{v}_i) = \underset{\tilde{d}_i \geq 0}{\arg\min} \tfrac{1}{2} \left\{ -2\tilde{d}_i \tilde{u}_i' W \tilde{v}_i + \tilde{d}_i^2 \right\} + \lambda \mathbf{p}(\tilde{d}_i) \tag{44}$$

The optimization problem analogous to (41) will be

$$\underset{\tilde{U},\tilde{V}}{\text{minimize}} \quad \tfrac{1}{2} \left\{ \|Z\|_F^2 - 2 \sum_{i=1}^n \hat{d}_i \tilde{u}_i' W \tilde{v}_i + \sum_{i=1}^n \hat{d}_i^2 \right\} + \lambda \sum_i \mathbf{p}(\hat{d}_i) \tag{45}$$

where $\hat{d}_i = S_\lambda^{\mathbf{p}}(\tilde{u}_i' W \tilde{v}_i)$*,* $\forall i$*. Any spectral function for which the above (45) is monotonically increasing in* $\tilde{u}_i' W \tilde{v}_i$ *for every* i *can be solved by a similar argument as given in the above proof. The solution will correspond to the first few largest left and right singular vectors of the matrix* W*. The optimal singular values will correspond to the relevant shrinkage/ threshold operator* $S_\lambda^{\mathbf{p}}(\cdot)$ *operated on the singular values of* W*. In particular for the indicator function* $\mathbf{p}(t) = \lambda \mathbf{1}(t \neq 0)$*, the top few singular values (un-shrunk) and the corresponding singular vectors is the solution.*

A.2 Proof of Lemma 3

This proof is based on sub-gradient characterizations ans is inspired by some techniques used in [CCS08].

Proof. From Lemma 1, we know that if $\hat{Z}$ solves the problem (9), then it satisfies the sub-gradient stationary conditions:

$$0 \in -(W - \hat{Z}) + \lambda \partial \|\hat{Z}\|_* \tag{46}$$

$\mathbf{S}_\lambda(W_1)$ and $\mathbf{S}_\lambda(W_2)$ solve the problem (9) with $W = W_1$ and $W = W_2$ respectively, hence (46) holds with $W = W_1$, $\hat{Z}_1 = \mathbf{S}_\lambda(W_1)$ and $W = W_2$, $\hat{Z}_2 = \mathbf{S}_\lambda(W_1)$.

The sub-gradients of the nuclear norm $\|Z\|_*$ are given by [CCS08, MGC09]

$$\partial \|Z\|_* = \{UV' + \omega : \omega_{m \times n},\ U'\omega = 0,\ \omega V = 0,\ \|\omega\|_2 \leq 1\} \tag{47}$$

where $Z = UDV'$ is the SVD of Z.

Let $p(\hat{Z}_i)$ denote an element in $\partial \|\hat{Z}_i\|_*$. Then

$$\hat{Z}_i - W_i + \lambda p(\hat{Z}_i) = 0,\ i = 1, 2. \tag{48}$$

The above gives

$$(\hat{Z}_1 - \hat{Z}_2) - (W_1 - W_2) + \lambda(p(\hat{Z}_1) - p(\hat{Z}_2)) = 0 \tag{49}$$

from which we obtain

$$\langle \hat{Z}_1 - \hat{Z}_2, \hat{Z}_1 - \hat{Z}_2 \rangle - \langle W_1 - W_2, \hat{Z}_1 - \hat{Z}_2 \rangle + \lambda \langle p(\hat{Z}_1) - p(\hat{Z}_2), \hat{Z}_1 - \hat{Z}_2 \rangle = 0 \tag{50}$$

where $\langle a, b \rangle = \text{trace}(a'b)$.

Now observe that

$$\langle p(\hat{Z}_1) - p(\hat{Z}_2), \hat{Z}_1 - \hat{Z}_2 \rangle = \langle p(\hat{Z}_1), \hat{Z}_1 \rangle - \langle p(\hat{Z}_1), \hat{Z}_2 \rangle - \langle p(\hat{Z}_2), \hat{Z}_1 \rangle + \langle p(\hat{Z}_2), \hat{Z}_2 \rangle \quad (51)$$

By the characterization of subgradients as in (47) and as also observed in [CCS08], we have

$$\langle p(\hat{Z}_i), \hat{Z}_i \rangle = \|\hat{Z}_i\|_* \text{ and } \|p(\hat{Z}_i)\|_2 \leq 1, \;\; i = 1, 2$$

which implies

$$|\langle p(\hat{Z}_i), \hat{Z}_j \rangle| \leq \|p(\hat{Z}_i)\|_2 \|\hat{Z}_j\|_* \leq \|\hat{Z}_j\|_* \text{ for } i \neq j \in \{1, 2\}$$

.

Using the above inequalities in (51) we obtain:

$$\begin{aligned} \langle p(\hat{Z}_1), \hat{Z}_1 \rangle + \langle p(\hat{Z}_2), \hat{Z}_2 \rangle &= \|\hat{Z}_1\|_* + \|\hat{Z}_2\|_* & (52) \\ -\langle p(\hat{Z}_1), \hat{Z}_2 \rangle - \langle p(\hat{Z}_2), \hat{Z}_1 \rangle &\geq -\|\hat{Z}_2\|_* - \|\hat{Z}_1\|_* & (53) \end{aligned}$$

Using (52,53) we see that the r.h.s. of (51) is non-negative. Hence

$$\langle p(\hat{Z}_1) - p(\hat{Z}_2), \hat{Z}_1 - \hat{Z}_2 \rangle \geq 0$$

Using the above in (49), we obtain:

$$\|\hat{Z}_1 - \hat{Z}_2\|_F^2 = \langle \hat{Z}_1 - \hat{Z}_2, \hat{Z}_1 - \hat{Z}_2 \rangle \leq \langle W_1 - W_2, \hat{Z}_1 - \hat{Z}_2 \rangle \quad (54)$$

Using the Cauchy-Schwarz Inequality $\|\hat{Z}_1 - \hat{Z}_2\|_2 \|W_1 - W_2\|_2 \geq \langle \hat{Z}_1 - \hat{Z}_2, W_1 - W_2 \rangle$ in (54) we get

$$\|\hat{Z}_1 - \hat{Z}_2\|_F^2 \leq \langle \hat{Z}_1 - \hat{Z}_2, W_1 - W_2 \rangle \leq \|\hat{Z}_1 - \hat{Z}_2\|_2 \|W_1 - W_2\|_2.$$

and in particular

$$\|\hat{Z}_1 - \hat{Z}_2\|_F^2 \leq \|\hat{Z}_1 - \hat{Z}_2\|_2 \|W_1 - W_2\|_2$$

which further simplifies to

$$\|W_1 - W_2\|_F^2 \geq \|\hat{Z}_1 - \hat{Z}_2\|_F^2 = \|\mathbf{S}_\lambda(W_1) - \mathbf{S}_\lambda(W_2)\|_F^2$$

□

A.3 Proof of Lemma 4

Proof. We will first show (19) by observing the following inequalities

$$\begin{aligned} \|Z_\lambda^{k+1} - Z_\lambda^k\|_F^2 &= \|\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)) - \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2 \\ \text{(by Lemma 3)} \quad &\leq \| \left(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)\right) - \left(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1})\right) \|_F^2 \\ &= \|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^{k-1})\|_F^2 & (55) \\ &\leq \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2 & (56) \end{aligned}$$

The above implies that the sequence $\{\|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2\}$ converges (since it is decreasing and bounded below). We still require to show that $\{\|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2\}$ converges to zero.

The convergence of $\{\|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2\}$ implies that:

$$\|Z_\lambda^{k+1} - Z_\lambda^k\|_F^2 - \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2 \to 0 \text{ as } k \to \infty$$

The above observation along with the inequality in (55,56) gives

$$\|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^{k-1})\|_F^2 - \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2 \to 0 \implies P_\Omega(Z_\lambda^k - Z_\lambda^{k-1}) \to 0 \tag{57}$$

as $k \to \infty$.

Lemma 2 shows that the non-negative sequence $f_\lambda(Z_\lambda^k)$ is decreasing in k. So as $k \to \infty$ the sequence $f_\lambda(Z_\lambda^k)$ converges. Furthermore from (16,17) we have

$$Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^k) - Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^{k+1}) \to 0 \text{ as } k \to \infty$$

which implies that

$$\|P_\Omega^\perp(Z_\lambda^k) - P_\Omega^\perp(Z_\lambda^{k+1})\|_F^2 \to 0 \text{ as } k \to \infty$$

The above along with (57) gives

$$Z_\lambda^k - Z_\lambda^{k-1} \to 0 \text{ as } k \to \infty$$

This completes the proof. □

A.4 Proof of Lemma 5

Proof. The sub-gradients of the nuclear norm $\|Z\|_*$ are given by

$$\partial\|Z\|_* = \{UV' + W : W_{m\times n},\ U'W = 0,\ WV = 0,\ \|W\|_2 \le 1\} \tag{58}$$

where $Z = UDV'$ is the SVD of Z. Since Z_λ^k minimizes $Q_\lambda(Z|Z_\lambda^{k-1})$, it satisfies:

$$0 \in -(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}) - Z_\lambda^k) + \partial\|Z_\lambda^k\|_* \ \ \forall k \tag{59}$$

Suppose Z^* is a limit point of the sequence Z_λ^k. Then there exists a subsequence $\{n_k\} \subset \{1, 2, \ldots\}$ such that $Z_\lambda^{n_k} \to Z^*$ as $k \to \infty$.

By Lemma 4 this subsequence $Z_\lambda^{n_k}$ satisfies

$$Z_\lambda^{n_k} - Z_\lambda^{n_k-1} \to 0$$

implying

$$P_\Omega^\perp(Z_\lambda^{n_k-1}) - Z_\lambda^{n_k} \longrightarrow P_\Omega^\perp(Z_\lambda^*) - Z_\lambda^* = -P_\Omega(Z^*)$$

Hence,

$$(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{n_k-1}) - Z_\lambda^{n_k}) \longrightarrow (P_\Omega(X) - P_\Omega(Z_\lambda^*)). \tag{60}$$

For every k, a sub-gradient $p(Z_\lambda^k) \in \partial\|Z_\lambda^k\|_*$ corresponds to a tuple (u_k, v_k, w_k) satisfying the properties of the set $\partial\|Z_\lambda^k\|_*$ (58).

Consider $p(Z_\lambda^{n_k})$ along the sub-sequence n_k. As $n_k \longrightarrow \infty$, $Z_\lambda^{n_k} \longrightarrow Z_\lambda^*$. Let

$$Z_\lambda^{n_k} = u_{n_k} D_{n_k} v'_{n_k} \ , Z^* = u_\infty D^* v'_\infty$$

denote the SVD's. The product of the singular vectors converge $u'_{n_k} v_{n_k} \rightarrow u'_\infty v_\infty$. Furthermore due to boundedness (passing on to a further subsequence if necessary) $w_{n_k} \rightarrow w_\infty$. The limit $u_\infty v'_\infty + w_\infty$ clearly satisfies the criterion of being a sub-gradient of Z^*. Hence this limit corresponds to $p(Z^*_\lambda) \in \partial \|Z^*_\lambda\|_*$.

Furthermore from (59, 60), passing on to the limits along the subsequence n_k we have

$$\mathbf{0} \in -(P_\Omega(X) - P_\Omega(Z^*_\lambda)) + \partial \|Z^*_\lambda\|_* \tag{61}$$

Hence the limit point Z^*_λ is a stationary point of $f_\lambda(Z)$.

We shall now prove (21). We know that for every n_k

$$Z^{n_k}_\lambda = \mathbf{S}_\lambda(P_\Omega(X) + P^\perp_\Omega(Z^{n_k-1}_\lambda)) \tag{62}$$

From Lemma 4 we know $Z^{n_k}_\lambda - Z^{n_k-1}_\lambda \rightarrow 0$. This observation along with the continuity of $\mathbf{S}_\lambda(\cdot)$ gives

$$\mathbf{S}_\lambda(P_\Omega(X) + P^\perp_\Omega(Z^{n_k-1}_\lambda)) \rightarrow \mathbf{S}_\lambda(P_\Omega(X) + P^\perp_\Omega(Z^*_\lambda))$$

Thus passing over to the limits on both sides of (62) we get

$$Z^*_\lambda = \mathbf{S}_\lambda(P_\Omega(X) + P^\perp_\Omega(Z^*_\lambda))$$

therefore completing the proof. □

References

[BM05] Samuel Burer and Renato D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–631, 2005.

[Boy08] Stephen Boyd. Ee 364b: Lecture notes, stanford university, 2008.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[CCS08] Jian-Feng Cai, Emmanuel J. Candes, and Zuowei Shen. A singular value thresholding algorithm for matrix completion, 2008.

[CR08] Emmanuel Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2008.

[CT09] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion, 2009.

[CW05] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4(4):1168–1200, 2005.

[DJKP95] D. Donoho, I. Johnstone, G. Kerkyachairan, and D. Picard. Wavelet shrinkage; asymptopia? (with discussion). *J. Royal. Statist. Soc.*, 57:201–337, 1995.

[Faz02] M. Fazel. *Matrix Rank Minimization with Applications*. PhD thesis, Stanford University, 2002.

[FHHT07] Jerome Friedman, Trevor Hastie, Holger Hoefling, and Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 2(1):302–332, 2007.

[FL01] Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360(13), 2001.

[Fri08] Jerome Friedman. Fast sparse regression and classification. Technical report, Department of Statistics, Stanford University, 2008.

[HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction (Springer Series in Statistics)*. Springer New York, 2 edition, 2009.

[HTS+99] Trevor Hastie, Robert Tibshirani, Gavin Sherlock, Michael Eisen, Patrick Brown, and David Botstein. Imputing missing data for gene expression arrays. Technical report, Division of Biostatistics, Stanford University, 1999.

[KOM09] Raghunandan H. Keshavan, Sewoong Oh, and Andrea Montanari. Matrix completion from a few entries. *CoRR*, abs/0901.3150, 2009.

[Lar] R.M. Larsen. Propack-software for large and sparse svd calculations.

[Lar98] R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. Technical Report DAIMI PB-357, Department of Computer Science, Aarhus University, 1998.

[LV08] Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identfication. submitted to Mathematical Programming, 2008.

[MGC09] S. Ma, D. Goldfarb, and L. Chen. Fixed Point and Bregman Iterative Methods for Matrix Rank Minimization. *ArXiv e-prints*, May 2009.

[RFP07] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, 2007.

[RMH07] Salakhutdinov R. R., A. Mnih, and G. E Hinton. Restricted boltzmann machines for collaborative filtering. In *International Conference on Machine Learning, Corvallis, Oregon.*, 2007.

[RS05] Jason Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. *22nd International Conference on Machine Learning*, 2005.

[SAJ05] Nathan Srebro, Noga Alon, and Tommi Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. *Advances in Neural Information Processing Systems*, 2005.

[SJ03] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *In 20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

[SN07] ACM SIGKDD and Netflix. Soft modelling by latent variables: the nonlinear iterative partial least squares (NIPALS) approach. In *Proceedings of KDD Cup and Workshop*, 2007.

[SRJ05] Nathan Srebro, Jason Rennie, and Tommi Jaakkola. Maximum margin matrix factorization. *Advances in Neural Information Processing Systems*, 17, 2005.

[TCS+01] Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B. Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1996.

[TPNT09] Gabor Takacs, Istvan Pilaszy, Bottyan Nemeth, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

[Zha07] Cun Hui Zhang. Penalized linear unbiased selection. Technical report, Departments of Statistics and Biostatistics, Rutgers University, 2007.

[ZY06] P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.