Regularization methods for learning incomplete matrices

Rahul Mazumder * Trevor Hastie † Robert Tibshirani ‡

June 11, 2009

Abstract

We use convex relaxation techniques to provide a sequence of solutions to the matrix completion problem. Using the nuclear norm as a regularizer, we provide simple and very efficient algorithms for minimizing the reconstruction error subject to a bound on the nuclear norm. Our algorithm iteratively replaces the missing elements with those obtained from a thresholded SVD. With warm starts this allows us to efficiently compute an entire regularization path of solutions.

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]. The "Netflix" competition is a primary 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 viewers would give for the 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 low 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. $Z \approx V_{mk}G_{kn}$, 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. Very 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. Typically we view the observed entries in X as the corresponding entries from Z contaminated with noise.

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:

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

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 [NJ03]. For a fully-observed X, on the other

*Statistics Department,Stanford University rahul.mazumder@gmail.com

†Statistics Department and Department of Health, Research and Policy, Stanford University, hastie@stanford.edu

‡Department of Health, Research and Policy and Statistics Department, Stanford University tibs@stanford.edu

hand, the solution is given by the singular value decomposition (SVD) of X. The following seemingly small modification to (1)

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

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 as explored in [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], the problems become prohibitively expensive if the dimensions of the matrix exceeds a hundred [CCS08]. In this paper we propose an algorithm that scales to large problems with $m, n \approx 10^4$–10^5 or even larger. 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 in MATLAB. For the same sized matrix with $|\Omega| = 10^5$ we obtain a rank-52 solution in under 80 minutes.

[CT09, CCS08, CR08] consider the criterion

$$\begin{aligned} \text{minimize} \quad & \|Z\|_* \\ \text{subject to} \quad & Z_{ij} = X_{ij},\ \forall (i,j) \in \Omega \end{aligned} \tag{3}$$

When $\delta = 0$, criterion (1) is equivalent to (3), in that it requires the training error to be zero. [CT09, CR08] further develop theoretical properties establishing the equivalence of the rank minimization and the nuclear norm minimization problems (1,3). Cai et. al. [CCS08] in their paper propose a first-order singular-value-thresholding algorithm scalable to large matrices for the problem (2) with $\delta = 0$. They comment on the problem (2), with $\delta > 0$, and suggest that it becomes prohibitive for large scale problems. Hence they consider the $\delta > 0$ case to be unsuitable for matrix completion.

We believe that (3) will almost always be too rigid, as it will force the procedure to overfit. If minimization of prediction error is our main goal, then the solution Z^* will typically lie somewhere in the interior of the path (Figure 1), indexed by δ.

In this paper we provide an algorithm for computing solutions of (2), on a grid of δ values, based on warm restarts. The algorithm is inspired by Hastie et al.'s SVD- impute [HTS+99, TCS+01] and is very different the proximal forward-backward splitting method of [CCS08, CW05, SMC08], which requires the choice of a step size. In [SMC08], the SVD step becomes prohibitive, so some randomized algorithms are used for the computation. Our algorithm is very different, and by exploiting matrix structure can solve problems much larger than those in [SMC08].

Our algorithm requires the computation of a low-rank SVD of a matrix (which is not sparse) at every iteration. Here we crucially exploit the problem matrix structure:

$$Y = Y_{SP} \ \text{(Sparse)} \quad + \quad Y_{LR} \ \text{(Low Rank)} \tag{4}$$

In (4) 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 few singular vectors/values of Y. Due to the specific structure of (4), multiplication by Y and Y' can both be done in a cost-efficient way..

2 Algorithm and Convergence analysis

2.1 Notation

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

$$P_\Omega(Y)\ (i,j) = \begin{cases} Y_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega, \end{cases} \tag{5}$$

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 (5) we can rewrite $\sum_{(i,j)\in\Omega}(Z_{ij} - X_{ij})^2$ as $\|P_\Omega(Z) - P_\Omega(X)\|_F^2$.

2.2 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 convex optimization problem*

$$\underset{Z}{\text{minimize}} \quad \tfrac{1}{2}\|Z - W\|_F^2 + \lambda\|Z\|_* \tag{6}$$

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

$$\mathbf{S}_\lambda(W) \equiv UD_\lambda V' \quad \textit{with} \quad D_\lambda = \text{diag}\left[(d_1 - \lambda)_+, \ldots, (d_r - \lambda)_+\right], \tag{7}$$

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

The notation $\mathbf{S}_\lambda(W)$ refers to *soft-thresholding* [DJKP95]. The proof follows by looking at the subgradient of the function to be minimized, and is given in [CCS08].

2.3 Algorithm

Problem (2) can be written in its equivalent Lagrangian form

$$\underset{Z}{\text{minimize}} \quad \tfrac{1}{2}\|P_\Omega(Z) - P_\Omega(X)\|_F^2 + \lambda\|Z\|_* \tag{8}$$

Here $\lambda \geq 0$ is a regularization parameter controlling the nuclear norm of the minimizer $\hat{Z}_\lambda$ of (8) (with a 1-1 mapping to $\delta > 0$ in (2)). We now present an algorithm for computing a series of solutions to (8) using warm starts. Define $f_\lambda(Z) = \frac{1}{2}\|P_\Omega(Z) - P_\Omega(X)\|_F^2 + \lambda\|Z\|_*$.

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 $\frac{\|f_\lambda(Z^{\text{new}}) - f_\lambda(Z^{\text{old}})\|_F^2}{\|f_\lambda(Z^{\text{old}})\|_F^2} < \epsilon$, go to step 2e.
   (c) Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$ and go to step 2b.
   (d) Assign $\hat{Z}_\lambda \leftarrow Z^{\text{new}}$ and $Z^{\text{old}} \leftarrow Z^{\text{new}}$
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 (8). Figure 1 shows some examples of solutions using Algorithm 1 (blue curves). We see test and training error in the left two columns as a function of the nuclear norm, obtained from a grid of values Λ. These error curves show a smooth and very competitive performance.

2.4 Convergence analysis

In this section we prove that Algorithm 1 converges to the solution to (2).

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

$$Q_\lambda(Z|\tilde{Z}) = \tfrac{1}{2}\|P_\Omega(X) + P_\Omega^\perp(\tilde{Z}) - Z\|_F^2 + \lambda\|Z\|_*, \tag{9}$$

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*

$$Z_\lambda^{k+1} \quad = \quad \arg\min_Z Q_\lambda(Z|Z_\lambda^k), \tag{10}$$

with $Z_\lambda^0 = 0$. The sequence Z_λ^k satisfies

$$f_\lambda(Z_\lambda^{k+1}) \leq Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^k) \leq f_\lambda(Z_\lambda^k) \tag{11}$$

Proof.

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

□

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

$$\|\mathbf{S}_\lambda(W_1) - \mathbf{S}_\lambda(W_2)\|_F^2 \leq \|W_1 - W_2\|_F^2 \tag{12}$$

Proof. We omit the proof here for the sake of brevity. The details work out by expanding the operator $\mathbf{S}_\lambda(\cdot)$ in terms of the singular value decomposition of W_1 and W_2. Then we use trace inequalities for the product of two matrices [Las95] where one is real symmetric, the other arbitrary. A proof of this Lemma also appears in [SMC08], though the method is different from ours. □

Lemma 4. *Suppose the sequence Z_λ^k obtained from (10) converges to Z_λ^∞. Then Z_λ^∞ is a stationary point of $f_\lambda(Z)$.*

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

$$\partial\|Z\|_* = \{UV' + W : W_{m\times n},\ U'W = 0,\ WV = 0,\ \|W\|_2 \leq 1\} \tag{13}$$

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

$$0 \in -(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}) - Z_\lambda^k) + \partial\|Z_\lambda^k\|_* \ \ \forall k \tag{14}$$

Since $Z_\lambda^k \to Z_\lambda^\infty$,

$$(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}) - Z_\lambda^k) \longrightarrow (P_\Omega(X) - P_\Omega(Z_\lambda^\infty)). \tag{15}$$

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). Then (passing on to a subsequence if necessary), $(u_k, v_k, w_k) \to (u_\infty, v_\infty, w_\infty)$ and this limit corresponds to $p(Z_\lambda^\infty) \in \partial\|Z_\lambda^\infty\|_*$.

Hence, from (14, 15), passing on to the limits

$$\mathbf{0} \in (P_\Omega(X) - P_\Omega(Z_\lambda^\infty)) + \partial\|Z_\lambda^\infty\|_* \tag{16}$$

This proves the stationarity of the limit Z_λ^∞. □

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

$$\min_Z \tfrac{1}{2}\|P_\Omega(Z) - P_\Omega(X)\|_F^2 + \lambda\|Z\|_* \tag{17}$$

Proof. Firstly observe that the sequence Z_λ^k is bounded; for it to converge it must have a unique accumulation point.

Observe that

$$\begin{aligned}
\|Z_\lambda^{k+1} - Z_\lambda^k\|_F^2 &= \|\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)) - \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2 \\
\text{(by Lemma 3)} \quad &\leq \| \left(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)\right) - \left(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1})\right) \|_F^2 \\
&= \|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^{k-1})\|_F^2 \\
&\leq \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2
\end{aligned} \tag{18}$$

Due to boundedness, every infinite subsequence of Z_λ^k has a further subsequence that converges. If the sequence Z_λ^k has two distinct limit points then for infinitely many $k' \geq 0$, $\|Z_\lambda^{k'} - Z_\lambda^{k'-1}\|_F \geq \epsilon$, for some $\epsilon > 0$. Using (18) this contradicts the convergence of any subsequence of Z_λ^k. Hence the sequence Z_λ^k converges. Using Lemma 4, the limit Z_λ^∞ is a stationary point of $f_\lambda(Z)$ and hence its minimizer. □

3 From soft to hard-thresholding

The nuclear norm behaves like a ℓ_1 norm, and can be viewed as a soft approximation of the ℓ_0 norm or rank of a matrix. In penalized linear regression for example, the ℓ_1 norm or LASSO [Tib96] is widely used as a convex surrogate for the ℓ_0 penalty or best-subset selection. The LASSO performs very well on a wide variety of situations in producing a parsimonious model with good prediction error. However, if the underlying model is very sparse, then the LASSO with its uniform shrinkage can overestimate the number of non-zero coefficients. In such situations concave penalized regressions are gaining popularity as a surrogate to ℓ_0. By analogy for matrices, it makes sense to go beyond the nuclear norm minimization problem to more aggressive penalties bridging the gap between ℓ_1 and ℓ_0. We propose minimizing

$$f_{p,\lambda}(Z) \quad = \quad \tfrac{1}{2}\|P_\Omega(Z) - P_\Omega(X)\|_F^2 + \lambda \sum_j p(\lambda_j(Z); \gamma) \tag{19}$$

where $p(|t|; \gamma)$ is concave in $|t|$. The parameter $\gamma \in [\gamma_{\inf}, \gamma_{\sup}]$ controls the degree of concavity, with $p(|t|; \gamma_{\inf}) = |t|$ (ℓ_1 penalty), on one end and $p(|t|; \gamma_{\sup}) = |t|^0$ (ℓ_0 penalty) on the other. In particular for the ℓ_0 penalty denote $f_{p,\lambda}(Z)$ by $f_{H,\lambda}(Z)$ for "hard" thresholding. See [Fri08, FL01, Zha07] for examples of such penalties.

Criterion (19) is no longer convex and hence becomes more difficult. It can be shown that Algorithm 1 can be modified in a suitable fashion for the penalty $p(\cdot; \gamma)$. This algorithm also has guaranteed convergence properties. The details of these arguments and statistical properties will be studied in a longer version of this paper. As a concrete example, we present here some features of the ℓ_0 norm regularization on singular values.

The version of (6) for the ℓ_0 norm is

$$\min_Z \tfrac{1}{2}\|Z - W\|_F^2 + \lambda\|Z\|_0. \tag{20}$$

The solution 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. As in (7), the thresholding operator resulting from (20) is

$$\mathbf{S}_\lambda^H(W) = UD_qV' \quad \text{where} \quad D_q = \text{diag}\,(d_1, \ldots, d_q, 0, \ldots, 0) \tag{21}$$

Similar to **Soft-Impute** (Algorithm 1), the algorithm **Hard-Impute** for the ℓ_0 penalty is given by Algorithm 2.

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 3.1).
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 $Z^{\text{new}} \leftarrow \mathbf{S}_\lambda^H(P_\Omega(X) + P_\Omega^\perp(Z^{\text{old}}))$
   (c) If $\frac{\|f_\lambda(Z^{\text{new}}) - f_\lambda(Z^{\text{old}})\|_F^2}{\|f_\lambda(Z^{\text{old}})\|_F^2} < \epsilon$, go to step 2e.
   (d) Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$ and go to step 2b.
   (e) Assign $\hat{Z}_{H,\lambda} \leftarrow Z^{\text{new}}$.
3. Output the sequence of solutions $\hat{Z}_{H,\lambda_1}, \ldots, \hat{Z}_{\lambda_K}$.

3.1 Post-processing 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 (8), then its *post-processed* version Z_λ^u obtained by "unshrinking" the eigen-values of the matrix Z_λ is obtained by

$$\begin{aligned} \alpha &= \underset{\alpha_i \geq 0,\ i=1,\ldots,r_\lambda}{\arg\min} \quad \|P_\Omega(X) - \sum_{i=1}^{r_\lambda} \alpha_i P_\Omega(u_i v_i')\|^2 \\ Z_\lambda^u &= UD_\alpha V', \end{aligned} \tag{22}$$

where $D_\alpha = \text{diag}(\alpha_1, \ldots, \alpha_{r_\lambda})$. Here r_λ is the rank of Z_λ and $Z_\lambda = UD_\lambda V'$ is its SVD. The estimation in (22) 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.[1] 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.

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 (19), 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 (22). The latter appears to significantly speed up convergence for **Hard-Impute**.

3.2 Computation

The computationally demanding part of Algorithms 1 and 2 is in $\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k))$ or $\mathbf{S}_\lambda^H(P_\Omega(X) + P_\Omega^\perp(Z_{H,\lambda}^k))$. These require calculating a low- rank SVD of the matrices of interest, since the underlying

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

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 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

$$P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) = \left\{P_\Omega(X) - P_\Omega(Z_\lambda^k)\right\} \text{ (Sparse)} \quad + \quad Z_\lambda^k \text{ (LowRank)} \tag{23}$$

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 the 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.

Our algorithms **Soft-Impute** and **Hard-Impute** both require repeated computation of a truncated SVD for a matrix W with structure as in (23). Note that in (23) the term $P_\Omega(Z_\lambda^k)$ can be computed in $O(|\Omega|r)$ flops using only the required outer products.

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 multiplication. cost.

For the reconstruction problem to be theoretically meaningful in the sense of [CT09], we require that $|\Omega| \approx nr\text{poly}(\log n)$. Hence introducing the LowRank 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 (3) 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. 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.

4 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 (**Rcon**) described in [KOM09] considers criterion (1) (in presence of noise). For every fixed rank r it uses a bi-convex algorithm on a Grassmanian Manifold for computing a rank-r approximation USV' (not the SVD). It uses a suitable starting point obtained by performing a sparse SVD on a *clean* version of the observed matrix $P_\Omega(X)$. To summarize, we look at the performance of the following methods:

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

(m,n)	$\|\Omega\|$	true rank (r)	SNR	effective rank $(\hat{r})$	# Iters	time(s)
$(3\times 10^4, 10^4)$	10^4	15	1	$(13, 47, 80)$	$(3,3,3)$	$(41.9, 124.7, 305.8)$
$(5\times 10^4, 5\times 10^4)$	10^4	15	1	8	80	237
$(10^5, 10^5)$	10^4	15	10	$(5, 14, 32, 62)$	$(3,3,3,3)$	$(37, 74.5, 199.8, 653)$
$(10^5, 10^5)$	10^5	15	10	$(18, 80)$	$(3,3)$	$(202, 1840)$
$(5\times 10^5, 5\times 10^5)$	10^4	15	10	11	3	628.14
$(5\times 10^5, 5\times 10^5)$	10^5	15	1	$(3, 11, 52)$	$(3,3,3)$	$(341.9, 823.4, 4810.75)$

Table 1: Performance of the **Soft-Impute** on different problem instances.

In all our simulation studies we took the underlying model as $Z_{m\times n} = U_{m\times r}V'_{r\times n}$ + noise; where U and V are random matrices with standard normal Gaussian entries, and noise is iid 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

$$\text{SNR} = \sqrt{\frac{\text{var}(UV')}{\text{var}(noise)}}; \quad \text{testerror} = \frac{\|P_\Omega^\perp(UV' - \hat{Z})\|_F^2}{\|P_\Omega^\perp(UV')\|_F^2} \tag{24}$$

In Figure 1, results corresponding to the training and test errors are shown for all algorithms mentioned above — nuclear norm (left two panels) and rank (right two panels)— in three problem instances. Since **Rcon** only uses rank, it is excluded from the left 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. We summarize our findings in the caption of the figure.

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

Acknowledgements

We thank Emmanuel Candes, Andrea Montanari and Steven Boyd 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.

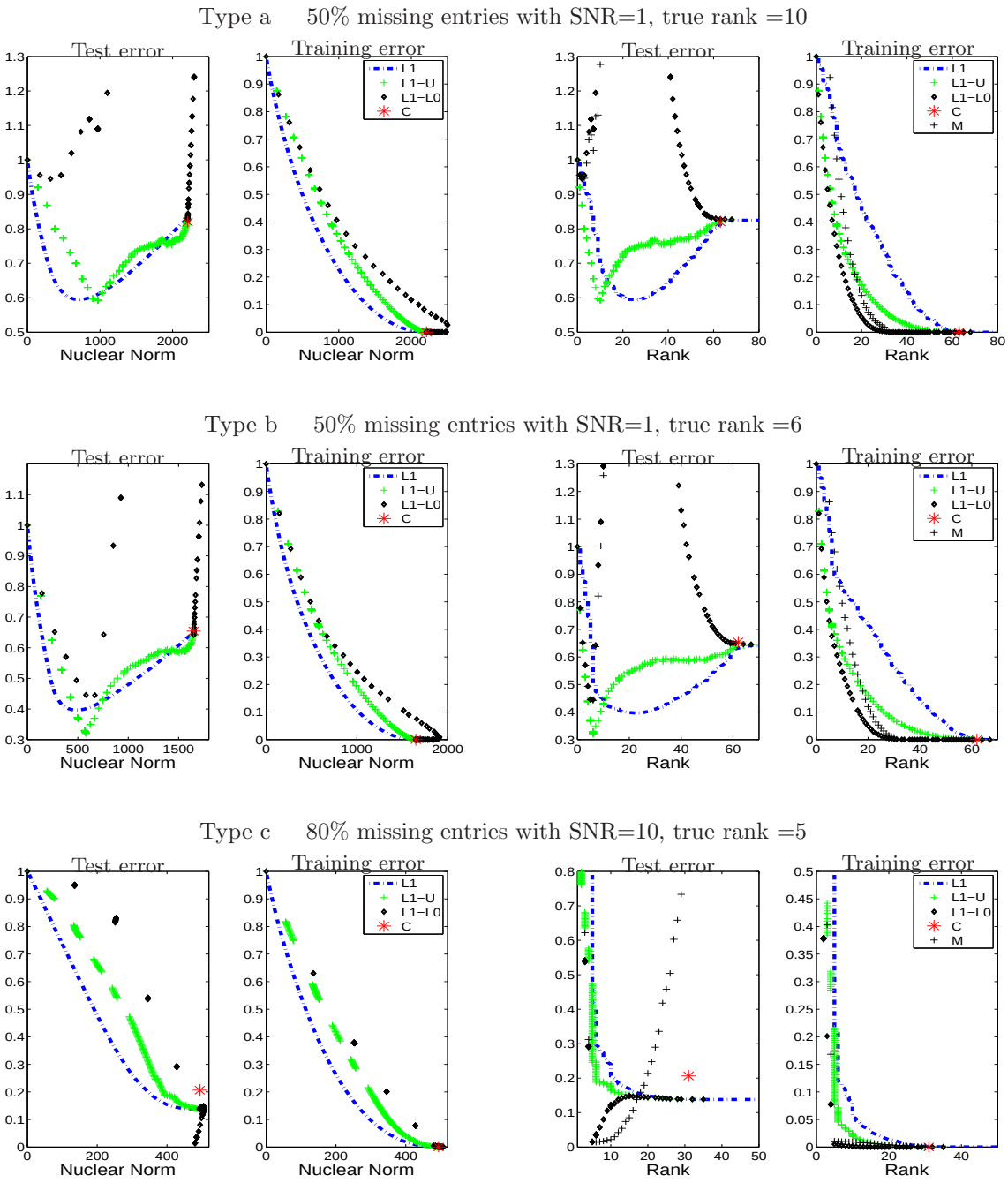

Figure 1: L1: solution for **Soft-Impute**; L1-U: Post processing after **Soft-Impute**; L1-L0 **Hard-Impute** applied to L1-U; C : **SVT** algorithm; M: **Recon** algorithm. **Soft-Impute** performs well in the presence of noise (top and middle panel). When the noise is low, **Hard-Impute** can improve its performance.The post-processed version tends to get the correct rank in many situations as in Types b,c. In Type b, the post-processed version does better than the rest in prediction error. In all the situations **SVT** algorithm does very poorly in prediction error, confirming our claim that (3) causes overfitting. **Recon** predicts poorly as well apart from Type-c, where it gets better error than **Soft-Impute**. However **Hard-Impute** and **Recon** have the same performance there.

References

[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.

[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.

[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.

[Las95] Jean B. Lasserre. A trace inequality for matrix product. *IEEE Transactions on AUtomatic Control*, 40, 1995.

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

[NJ03] Nathan Srebro Nati and Tommi Jaakkola. Weighted low-rank approximations. In *In 20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

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

[SMC08] D. Goldfarb S. Ma and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. 2008.

[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.

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