Journal of Machine Learning Research 11 (2010) 2287-2322 Submitted 7/09; Revised 4/10; Published 8/10

Spectral Regularization Algorithms for Learning Large Incomplete Matrices

Rahul Mazumder RAHULM@STANFORD.EDU
Trevor Hastie* HASTIE@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305

Robert Tibshirani† TIBS@STANFORD.EDU
Department of Health, Research and Policy
Stanford University
Stanford, CA 94305

Editor: Tommi Jaakkola

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 of order linear in the matrix dimensions. Our semidefinite-programming algorithm is readily scalable to large matrices; for example SOFT-IMPUTE takes a few hours to compute low-rank approximations of a $10^6 \times 10^6$ incomplete matrix with 10^7 observed entries, and fits a rank-95 approximation to the full Netflix training set in 3.3 hours. Our methods achieve good training and test errors and exhibit superior timings when compared to other competitive state-of-the-art techniques.

Keywords: collaborative filtering, nuclear norm, spectral regularization, netflix prize, large scale convex optimization

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 (Candès and Recht, 2008; Candès and Tao, 2009; Rennie and Srebro, 2005). The "Netflix" competition (for example, SIGKDD and Netflix, 2007) 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 about 480K viewers and 18K movies, and hence 8.6 billion (8.6×10^9) potential entries. However, on average each viewer rates about 200

∗. Also in the Department of Health, Research and Policy.
†. Also in the Department of Statistics.

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, that is, $Z \approx V_{m\times k}G_{k\times n}$, 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.

Srebro et al. (2005a) studied generalization error bounds for learning low-rank matrices. Recently Candès and Recht (2008), Candès and Tao (2009), and Keshavan et al. (2009) 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.

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:

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

where $\delta \ge 0$ is a regularization parameter controlling the tolerance in training error. The rank constraint in (1) makes the problem for general Ω combinatorially hard (Srebro and Jaakkola, 2003). 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),

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

makes the problem convex (Fazel, 2002). 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 (Fazel, 2002; Candès and Recht, 2008; Candès and Tao, 2009; Recht et al., 2007). Optimization of (2) is a semi-definite programming problem (Boyd and Vandenberghe, 2004) and can be solved efficiently for small problems, using modern convex optimization software like SeDuMi and SDPT3 (Grant and Boyd., 2009). However, since these algorithms are based on second order methods (Liu and Vandenberghe, 2009), they can become prohibitively expensive if the dimensions of the matrix get large (Cai et al., 2008). Equivalently we can reformulate (2) in *Lagrange* form

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

Here $\lambda \ge 0$ is a regularization parameter controlling the nuclear norm of the minimizer $\hat{Z}_\lambda$ of (3); there is a 1-1 mapping between $\delta \ge 0$ and $\lambda \ge 0$ 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 10^6–10^8 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-40 solution to (2) for a problem of size $10^5 \times 10^5$ and $|\Omega| = 5 \times 10^6$ observed entries in less than 18 minutes.
- For the same sized matrix with $|\Omega| = 10^7$ we obtain a rank-5 solution in less than 21 minutes.
- For a $10^6 \times 10^5$ sized matrix with $|\Omega| = 10^8$ a rank-5 solution is obtained in approximately 4.3 hours.
- We fit a rank-66 solution for the Netflix data in 2.2 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 95 solution takes 3.27 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 in Section 4. The computational aspects of the algorithm are described in Section 5, and Section 6 discusses how nuclear norm regularization can be generalized to more aggressive and general types of spectral regularization. Section 7 describes post-processing of "selectors" and initialization. We discuss comparisons with related work, simulations and experimental studies in Section 9 and application to the Netflix data in Section 10.

2. Context and Related Work

Candès and Tao (2009), Cai et al. (2008), and Candès and Recht (2008) consider the criterion

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

With $\delta = 0$, the criterion (1) is equivalent to (4), in that it requires the training error to be zero. Cai et al. (2008) 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 over-fitting. If minimization of prediction error is an important goal, then the optimal solution $\hat{Z}$ will typically lie somewhere in the interior of the path indexed by δ (Figures 2, 3 and 4).

In this paper we provide an algorithm SOFT-IMPUTE for computing solutions of (3) on a grid of λ values, based on warm restarts. The algorithm is inspired by SVD-IMPUTE (Troyanskaya et al.,

1. For large problems data transfer, access and reading take quite a lot of time and is dependent upon the platform and machine. Over here we report the times taken for the computational bottle-neck, that is, the SVD computations over all iterations. All times are reported based on computations done in a Intel Xeon Linux 3GHz processor using MATLAB, with no C or Fortran interlacing.

2001)—an EM-type (Dempster et al., 1977) iterative algorithm that alternates between imputing the missing values from a current SVD, and updating the SVD using the "complete" data matrix. In its very motivation, SOFT-IMPUTE is different from generic first order algorithms (Cai et al., 2008; Ma et al.; Ji and Ye, 2009). The latter 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.

The iterative algorithms proposed in Ma et al. and Ji and Ye (2009) require the computation of a SVD of a dense matrix (with dimensions equal to the size of the matrix X) at every iteration, as the bottleneck. This makes the algorithms prohibitive for large scale computations. Ma et al. use randomized algorithms for the SVD computation. Our algorithm SOFT-IMPUTE also requires an SVD computation at every iteration, but by exploiting the *problem structure*, can easily handle matrices of very large dimensions. 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 rank $\tilde{r} \ll m,n$, where $\tilde{r}$ is very close to $r \ll m,n$ the rank of the estimated matrix Z (upon convergence of the algorithm). For large scale problems, we use iterative methods based on Lanczos bidiagonalization with partial re-orthogonalization (as in the PROPACK algorithm, Larsen, 1998), for computing the first $\tilde{r}$ 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. In decomposition (5), the computationally burdensome work in computing a low-rank SVD is of an order that depends linearly on the matrix dimensions. More precisely, evaluating each singular vector requires computation of the order of $O((m+n)\tilde{r}) + O(|\Omega|)$ flops and evaluating r' of them requires $O((m+n)\tilde{r}r') + O(|\Omega|r')$ flops. Exploiting warm-starts, we observe that $\tilde{r} \approx r$—hence every SVD step of our algorithm computes r singular vectors, with complexity of the order $O((m+n)r^2) + O(|\Omega|r)$ flops. This computation is performed for the number of iterations SOFT-IMPUTE requires to run till convergence or a certain tolerance.

In this paper we show asymptotic convergence of SOFT-IMPUTE and further derive its non-asymptotic rate of convergence which scales as $O(1/k)$ (k denotes the iteration number). However, in our experimental studies on low-rank matrix completion, we have observed that our algorithm is faster (based on timing comparisons) than the accelerated version of Nesterov (Ji and Ye, 2009; Nesterov, 2007), having a provable (worst case) convergence rate of $O(\frac{1}{k^2})$. With warm-starts SOFT-IMPUTE computes the entire regularization path 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 (1). This is supported by our experimental studies. 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, Tibshirani, 1996; Hastie et al., 2009). 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 (Friedman, 2008; Hastie et al., 2009; Mazumder et al., 2009). 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 matrix factorization* methods—in short MMMF—and use

a factor model for the matrix Z (Srebro et al., 2005b). Let $Z = UV'$ where $U_{m\times r'}$ and $V_{n\times r'}$, and consider the following problem

$$\underset{U,V}{\text{minimize}} \frac{1}{2}\sum_{(i,j)\in\Omega}(X_{ij}-(UV')_{ij})^2+\frac{\lambda}{2}(\|U\|_F^2+\|V\|_F^2). \tag{6}$$

It turns out that (6) is intimately related to (3), since (see Lemma 6)

$$||Z||_* = \min_{U,V:\, Z=UV'} \frac{1}{2}\left(\|U\|_F^2+\|V\|_F^2\right).$$

For example, if $r' = \min(m,n)$, the solution to (6) coincides with the solution to (3).[2] However, (6) is not convex in its arguments, while (3) is. We compare these two criteria in detail in Section 8, and the relative performance of their respective algorithms in Section 9.2.

3. SOFT-IMPUTE–an Algorithm for Nuclear Norm Regularization

We first introduce some notation that will be used for the rest of this article.

3.1 Notation

We adopt the notation of Cai et al. (2008). Define a matrix $P_\Omega(Y)$ (with dimension $m\times n$)

$$P_\Omega(Y)\,(i,j) = \begin{cases} Y_{ij} & \text{if } (i,j)\in\Omega \\ 0 & \text{if } (i,j)\notin\Omega, \end{cases} \tag{7}$$

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 (7) 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, 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*

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}\|W-Z\|_F^2+\lambda\|Z\|_* \tag{8}$$

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

$$\mathbf{S}_\lambda(W) \equiv UD_\lambda V' \quad \textit{with} \quad D_\lambda = \text{diag}\left[(d_1-\lambda)_+,\ldots,(d_r-\lambda)_+\right], \tag{9}$$

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* (Donoho et al., 1995). Lemma 1 appears in Cai et al. (2008) and Ma et al. where the proof uses 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 6. Our proof is followed by a remark that covers these more general cases.

2. We note here that the original MMMF formulation uses $r' = \min\{m,n\}$. In this paper we will consider it for a family of r' values.

3.3 Algorithm

Using the notation in 3.1, we rewrite (3) as:

$$\underset{Z}{\text{minimize}} \quad f_\lambda(Z) := \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_* \,. \tag{10}$$

We now present Algorithm 1—SOFT-IMPUTE—for computing a series of solutions to (10) for different values of λ using warm starts.

Algorithm 1 SOFT-IMPUTE

1. Initialize $Z^{\text{old}} = 0$.
2. Do for $\lambda_1 > \lambda_2 > \ldots > \lambda_K$:
   (a) Repeat:
      i. Compute $Z^{\text{new}} \leftarrow \mathbf{S}_{\lambda_k}(P_\Omega(X) + P_\Omega^\perp(Z^{\text{old}}))$.
      ii. If $\frac{\|Z^{\text{new}} - Z^{\text{old}}\|_F^2}{\|Z^{\text{old}}\|_F^2} < \varepsilon$ exit.
      iii. Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$.
   (b) Assign $\hat{Z}_{\lambda_k} \leftarrow Z^{\text{new}}$.
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). Figures 2, 3 and 4 show some examples of solutions using SOFT-IMPUTE (blue continuous 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.

4. Convergence Analysis

In this section we study the convergence properties of Algorithm 1. Unlike generic first-order methods (Nesterov, 2003) including competitive first-order methods for nuclear norm regularized problems (Cai et al., 2008; Ma et al.), SOFT-IMPUTE does not involve the choice of any additional step-size. Most importantly our algorithm is readily scalable for solving large scale semidefinite programming problems (2) and (10) as will be explained later in Section 5.

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

$$Q_\lambda(Z|\tilde{Z}) = \frac{1}{2}\|P_\Omega(X) + P_\Omega^\perp(\tilde{Z}) - Z\|_F^2 + \lambda\|Z\|_* \tag{11}$$

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

In Section 4.1, we show that the sequence Z_λ^k generated via SOFT-IMPUTE *converges* asymptotically, that is, as $k \to \infty$ to a minimizer of the objective function $f_\lambda(Z)$. SOFT-IMPUTE produces a sequence of solutions for which the criterion decreases to the optimal solution with every iteration and the successive iterates get closer to the optimal set of solutions of the problem 10. Section 4.2

derives the non-asymptotic convergence rate of the algorithm. The latter analysis concentrates on the objective values $f_\lambda(Z_\lambda^k)$. Due to computational resources if one wishes to stop the algorithm after K iterations, then Theorem 2 provides a certificate of how *far* Z_λ^k is from the solution. Though Section 4.1 alone establishes the convergence of $f_\lambda(Z_\lambda^k)$ to the minimum of $f_\lambda(Z)$, this does not, in general, settle the *convergence* of Z_λ^k unless further conditions (like strong convexity) are imposed on $f_\lambda(\cdot)$.

4.1 Asymptotic Convergence

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

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

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

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

Proof Note that

$$Z_\lambda^{k+1} = \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)). \tag{12}$$

By Lemma 1 and the definition (11) of $Q_\lambda(Z|Z_\lambda^k)$, we have:

$$\begin{aligned}
f_\lambda(Z_\lambda^k) &= Q_\lambda(Z_\lambda^k|Z_\lambda^k) \\
&= \frac{1}{2}\|P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) - Z_\lambda^k\|_F^2 + \lambda\|Z_\lambda^k\|_* \\
&\geq \min_Z \frac{1}{2}\left\{\|P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) - Z\|_F^2\right\} + \lambda\|Z\|_* \\
&= Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^k) \\
&= \frac{1}{2}\|\left\{P_\Omega(X) - P_\Omega(Z_\lambda^{k+1})\right\} + \left\{P_\Omega^\perp(Z_\lambda^k) - P_\Omega^\perp(Z_\lambda^{k+1})\right\}\|_F^2 + \lambda\|Z_\lambda^{k+1}\|_* \\
&= \frac{1}{2}\left\{\|P_\Omega(X) - P_\Omega(Z_\lambda^{k+1})\|_F^2 + \|P_\Omega^\perp(Z_\lambda^k) - P_\Omega^\perp(Z_\lambda^{k+1})\|_F^2\right\} + \lambda\|Z_\lambda^{k+1}\|_* \qquad (13) \\
&\geq \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z_\lambda^{k+1})\|_F^2 + \lambda\|Z_\lambda^{k+1}\|_* \qquad (14) \\
&= Q_\lambda(Z_\lambda^{k+1}|Z_\lambda^{k+1}) \\
&= f(Z_\lambda^{k+1}).
\end{aligned}$$

■

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

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

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

Lemma 3 is proved in Ma et al.; their proof is complex and based on trace inequalities. We give a concise proof based on elementary convex analysis 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:*

$$\|Z_\lambda^{k+1} - Z_\lambda^k\|_F^2 \leq \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2 \quad \forall k. \tag{15}$$

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*

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

Hence it is a solution to the fixed point equation

$$Z = \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z)). \tag{16}$$

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*

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}\|P_\Omega(X) - P_\Omega(Z)\|_F^2 + \lambda\|Z\|_*. \tag{17}$$

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 (17) and satisfies the fixed point equation (16).

Hence

$$\begin{aligned}
\|\hat{Z}_\lambda - Z_\lambda^k\|_F^2 &= \|\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(\hat{Z}_\lambda)) - \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2 && (18)\\
&\leq \|(P_\Omega(X) + P_\Omega^\perp(\hat{Z}_\lambda)) - (P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2\\
&= \|P_\Omega^\perp(\hat{Z}_\lambda - Z_\lambda^{k-1})\|_F^2\\
&\leq \|\hat{Z}_\lambda - Z_\lambda^{k-1}\|_F^2. && (19)
\end{aligned}$$

In (18) two substitutions were made; the left one using (16) in Lemma 5, the right one using (12). Inequality (19) 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 (19) implies that at every iteration Z_λ^k gets closer to an optimal solution for the problem (17).[3] This property holds in addition to the decrease of the objective function (Lemma 2) at every iteration.

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

4.2 Convergence Rate

In this section we derive the worst case convergence rate of SOFT-IMPUTE.

Theorem 2 *For every fixed $\lambda \geq 0$, the sequence $Z_\lambda^k; k \geq 0$ defined in Lemma 2 has the following non-asymptotic (worst) rate of convergence:*

$$f_\lambda(Z_\lambda^k) - f_\lambda(Z_\lambda^\infty) \leq \frac{2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2}{k+1}. \tag{20}$$

The proof of this theorem is in Appendix A.6.

In light of Theorem 2, a $\delta > 0$ accurate solution of $f_\lambda(Z)$ is obtained after a maximum of $\frac{2}{\delta}\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2$ iterations. Using warm-starts, SOFT-IMPUTE traces out the path of solutions on a grid of λ values $\lambda_1 > \lambda_2 > \ldots > \lambda_K$ with a total of [4]

$$\sum_{i=1}^{K} \frac{2}{\delta}\|\hat{Z}_{\lambda_{i-1}} - Z_{\lambda_i}^\infty\|_F^2 \tag{21}$$

iterations. Here $\hat{Z}_{\lambda_0} = 0$ and $\hat{Z}_{\lambda_i}$ denotes the output of SOFT-IMPUTE (upon convergence) for $\lambda = \lambda_i$ ($i \in \{1, \ldots, K-1\}$). The solutions $Z_{\lambda_i}^\infty$ and $Z_{\lambda_{i-1}}^\infty$ are likely to be *close* to each other, especially on a dense grid of λ_i's. Hence every summand of (21) and the total number of iterations is expected to be significantly smaller than that obtained via arbitrary cold-starts.

5. Computational Complexity

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 have explicit[5] 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

$$\begin{array}{rcccc} P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k) & = & \{P_\Omega(X) - P_\Omega(Z_\lambda^k)\} & + & Z_\lambda^k \\ & = & \textit{Sparse} & + & \textit{Low Rank}. \end{array} \tag{22}$$

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 (Larsen, 2004, 1998) 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.

4. We assume the solution $\hat{Z}_\lambda$ at every $\lambda \in \{\lambda_1, \ldots, \lambda_K\}$ is computed to an accuracy of $\delta > 0$.

5. Though we cannot prove theoretically that every iterate of the sequence Z_λ^k will be of low-rank; this observation is rather practical based on the manner in which we trace out the entire path of solutions based on warm-starts. Our simulation results support this observation as well.

Algorithm SOFT-IMPUTE requires repeated computation of a truncated SVD for a matrix W with structure as in (22). Assume that at the current iterate, the matrix Z_λ^k has rank $\tilde{r}$. Note that in (22) the term $P_\Omega(Z_\lambda^k)$ can be computed in $O(|\Omega|\tilde{r})$ 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 $O(|\Omega|\tilde{r})$ to create the matrix $P_\Omega(Z_\lambda^k)$, this is used for each of the $\tilde{r}$ such multiplications (which also cost $O(|\Omega|\tilde{r})$), so we need not include that cost here. The Low Rank part costs $O((m+n)\tilde{r})$ for the multiplication by b_1. Hence the cost is $O(|\Omega|)+O((m+n)\tilde{r})$ per vector multiplication. Supposing we want a $\tilde{r}$ rank SVD of the matrix (22), the cost will be of the order of $O(|\Omega|\tilde{r})+O((m+n)(\tilde{r})^2)$ (for that iteration, that is, to obtain Z_λ^{k+1} from Z_λ^k). Suppose the rank of the solution Z_λ^k is r, then in light of our above observations $\tilde{r} \approx r \ll \min\{m,n\}$ and the order is $O(|\Omega|r)+O((m+n)r^2)$.

For the reconstruction problem to be theoretically meaningful in the sense of Candès and Tao (2009) we require that $|\Omega| \approx nr \cdot \text{poly}(\log n)$. 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 (Cai et al., 2008) for exact matrix completion (4) involves calculating the SVD of a sparse matrix with cost $O(|\Omega|)$. This implies that the computational order of SOFT-IMPUTE and that of SVT 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. On small scale examples, based on comparisons with the accelerated gradient method of Nesterov (see Section 9.3; Ji and Ye, 2009; Nesterov, 2007) we find that our algorithm converges faster than the latter in terms of run-time and number of SVD computations/ iterations. This supports the computational effectiveness of SOFT-IMPUTE. In addition, 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 above discussions focus on the computational complexity for obtaining a low-rank SVD, which is to be performed at every iteration of SOFT-IMPUTE. Similar to the total iteration complexity bound of SOFT-IMPUTE (21), the total cost to compute the regularization path on a grid of λ values is given by:

$$\sum_{i=1}^{K} O\left((|\Omega|\bar{r}_{\lambda_i} + (m+n)\bar{r}_{\lambda_i}^2)\frac{2}{\delta}\|\hat{Z}_{\lambda_{i-1}} - Z_{\lambda_i}^{\infty}\|_F^2 \right).$$

Here $\bar{r}_\lambda$ denotes the rank[6] (on an average) of the iterates Z_λ^k generated by SOFT-IMPUTE for fixed λ.

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.

6. We assume, above that the grid of values $\lambda_1 > \ldots \lambda_K$ is such that *all* the solutions $Z_\lambda, \lambda \in \{\lambda_1, \ldots, \lambda_K\}$ are of *small* rank, as they appear in Section 5.

6. 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. However, if the underlying model is very sparse, then the LASSO with its uniform shrinkage can both overestimate the number of non-zero coefficients (Friedman, 2008) in the model, and overly shrink (bias) those included toward zero. In this section we propose a natural generalization of SOFT-IMPUTE to overcome these problems.

Consider again the problem

$$\underset{\text{rank}(Z)\le k}{\text{minimize}} \ \frac{1}{2}\|P_\Omega(X)-P_\Omega(Z)\|_F^2,$$

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

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

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 (8) as follows:

$$\underset{Z}{\text{minimize}} \ \frac{1}{2}\|W-Z\|_F^2+\lambda\|Z\|_0, \tag{23}$$

where $\|Z\|_0=\text{rank}(Z)$. The solution of (23) 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 (23) is non-convex in W but its global minimizer can be evaluated. As in (9) the thresholding operator resulting from (23) is

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

Similar to SOFT-IMPUTE (Algorithm 1), we present below HARD-IMPUTE (Algorithm 2) for the ℓ_0 penalty. The continuous parameterization via λ does not appear to offer obvious advantages over rank-truncation methods. We note that it does allow for a continuum of warm starts, and is a natural post-processor for the output of SOFT-IMPUTE (next section). But it also allows for further generalizations that bridge the gap between hard and soft regularization methods.

In penalized regression there have been recent developments directed towards "bridging" the gap between the ℓ_1 and ℓ_0 penalties (Friedman, 2008; Zhang, 2010; Mazumder et al., 2009). This is done via using non-convex 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, and 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

$$f_{\mathbf{p},\lambda}(Z) \ = \ \frac{1}{2}\|P_\Omega(X)-P_\Omega(Z)\|_F^2+\lambda\sum_j \mathbf{p}(\gamma_j(Z);\mu), \tag{24}$$

Algorithm 2 HARD-IMPUTE

1. Initialize $\tilde{Z}_{\lambda_k}$ $k = 1, \ldots, K$ (for example, using SOFT-IMPUTE; see Section 7).
2. Do for $\lambda_1 > \lambda_2 > \ldots > \lambda_K$:
   (a) Repeat:
      i. Compute $Z^{\text{new}} \leftarrow \mathbf{S}^H_{\lambda_k}(P_\Omega(X) + P^\perp_\Omega(Z^{\text{old}}))$.
      ii. If $\frac{\|Z^{\text{new}} - Z^{\text{old}}\|_F^2}{\|Z^{\text{old}}\|_F^2} < \varepsilon$ exit.
      iii. Assign $Z^{\text{old}} \leftarrow Z^{\text{new}}$.
   (b) Assign $\hat{Z}_{H,\lambda_k} \leftarrow Z^{\text{new}}$.
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 Friedman (2008), Mazumder et al. (2009) and Zhang (2010) 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 (24) we will look at the analogous version of (8, 23) which is

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}\|W - Z\|_F^2 + \lambda \sum_j \mathbf{p}(\gamma_j(Z);\mu).$$

The solution is given by a thresholded SVD of W,

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

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 2(a)i. The corresponding step becomes:

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

However these types of spectral regularization make the criterion (24) non-convex and hence it becomes difficult to optimize globally. Recht et al. (2007) and Bach (2008) also consider the rank estimation problem from a theoretical standpoint.

7. 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 (10), then its *post-processed* version Z_λ^u obtained by "unshrinking" the eigen-values of the matrix Z_λ is obtained by

$$\begin{aligned} \alpha &= \underset{\alpha_i \geq 0,\, i=1,\ldots,r_\lambda}{\arg\min} \quad \|P_\Omega(X) - \sum_{i=1}^{r_\lambda} \alpha_i P_\Omega(u_i v_i')\|^2 \\ Z_\lambda^u &= UD_\alpha V', \end{aligned} \tag{25}$$

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 (25) 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.[7] If the least squares solutions α 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. Hence given U, V (each of rank r_λ) from the SOFT-IMPUTE algorithm, we solve

$$\begin{aligned} \hat{M} &= \underset{M}{\arg\min} \quad \|P_\Omega(X) - P_\Omega(UMV')\|^2, \\ \text{where, } \hat{Z}_\lambda &= U\hat{M}V'. \end{aligned} \tag{26}$$

The objective function in (26) is the Frobenius norm of an affine function of M and hence can be optimized very efficiently. Scalability issues pertaining to the optimization problem (26) can be handled fairly efficiently via conjugate gradients. Criterion (26) 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 over-fitting. In this paper we report experiments based on (25).

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 (24), 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 (25). The latter appears to significantly speed up convergence for HARD-IMPUTE. This observation is based on our simulation studies.

8. Soft-Impute and Maximum-Margin Matrix Factorization

In this section we compare in detail the MMMF criterion (6) with the SOFT-IMPUTE criterion (3). For ease of comparison here, we put down these criteria again using our P_Ω notation.

MMMF solves

$$\underset{U,V}{\text{minimize}} \quad \frac{1}{2}||P_\Omega(X - UV)||_F^2 + \frac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2), \tag{27}$$

where $U_{m \times r'}$ and $V_{n \times r'}$ are arbitrary (non-orthogonal) matrices. This problem formulation and related optimization methods have been explored by Srebro et al. (2005b) and Rennie and Srebro (2005).

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

SOFT-IMPUTE solves

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2}||P_\Omega(X-Z)||_F^2 + \lambda\|Z\|_*. \tag{28}$$

For each given maximum rank, MMMF produces an estimate by doing further shrinkage with its quadratic regularization. SOFT-IMPUTE performs rank reduction and shrinkage at the same time, in one smooth convex operation. The following theorem shows that this one-dimensional SOFT-IMPUTE family lies exactly in the two-dimensional MMMF family.

Theorem 3 *Let X be $m \times n$ with observed entries indexed by Ω.*

1. *Let $r' = \min(m,n)$. Then the solutions to (27) and (28) coincide for all $\lambda \geq 0$.*
2. *Suppose $\hat{Z}^*$ is a solution to (28) for $\lambda^* > 0$, and let r^* be its rank. Then for any solution $\hat{U}, \hat{V}$ to (27) with $r' = r^*$ and $\lambda = \lambda^*$, $\hat{U}\hat{V}^T$ is a solution to (28). The SVD factorization of $\hat{Z}^*$ provides one such solution to (27). This implies that the solution space of (28) is contained in that of (27).*

Remarks:

1. Part 1 of this theorem appears in a slightly different form in Srebro et al. (2005b).
2. In part 1, we could use $r' > \min(m,n)$ and get the same equivalence. While this might seem unnecessary, there may be computational advantages; searching over a bigger space might protect against local minima. Likewise in part 2, we could use $r' > r^*$ and achieve the same equivalence. In either case, no matter what r' we use, the solution matrices $\hat{U}$ and $\hat{V}$ have the same rank as $\hat{Z}$.
3. Let $\hat{Z}(\lambda)$ be a solution to (28) at λ. We conjecture that rank $[\hat{Z}(\lambda)]$ is monotone non-increasing in λ. If this is the case, then Theorem 3, part 2 can be further strengthened to say that for all $\lambda \geq \lambda^*$ and $r' = r^*$ the solutions of (27) coincide with that of (28).

The MMMF criterion (27) defines a two-dimensional family of models indexed by (r', λ), while the SOFT-IMPUTE criterion (28) defines a one-dimensional family. In light of Theorem 3, this family is a special path in the two-dimensional grid of solutions $[\hat{U}_{(r',\lambda)}, \hat{V}_{(r',\lambda)}]$. Figure 1 depicts the situation. Any MMMF model at parameter combinations above the red squares are redundant, since their fit is the same at the red square. However, in practice the red squares are not known to MMMF, nor is the actual rank of the solution. Further orthogonalization of $\hat{U}$ and $\hat{V}$ would be required to reveal the rank, which would only be approximate (depending on the convergence criterion of the MMMF algorithm).

Despite the equivalence of (27) and (28) when $r' = \min(m,n)$, the criteria are quite different. While (28) is a convex optimization problem in Z, (27) is a non-convex problem in the variables U, V and has possibly several local minima; see also Abernethy et al. (2009). It has been observed empirically and theoretically (Burer and Monteiro, 2005; Rennie and Srebro, 2005) that bi-convex methods used in the optimization of (27) can get stuck in sub-optimal local minima for a small value of r' or a poorly chosen starting point. For a large number of factors r' and large dimensions m, n the computational cost may be quite high (See also experimental studies in Section 9.2).

Criterion (28) is convex in Z for every value of λ, and it outputs the solution $\hat{Z}$ in the form of its soft-thresholded SVD, implying that the "factors" U, V are already orthogonal and the rank is known.

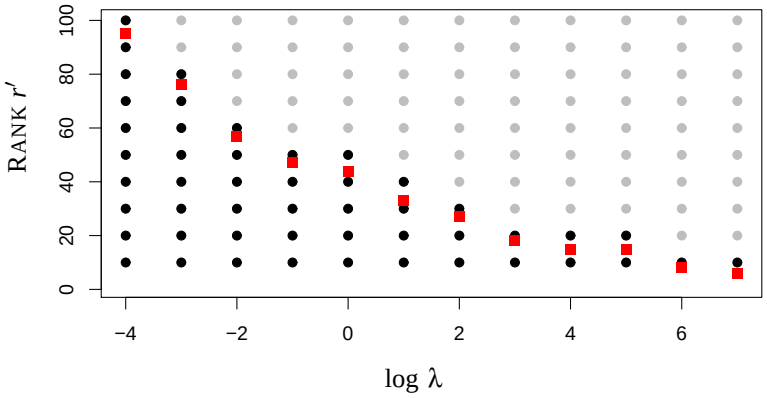

Figure 1: Comparison of the parameter space for MMMF (grey and black points), and SOFT-IMPUTE (red squares) for a simple example. Since all MMMF solutions with parameters above the red squares are identical to the SOFT-IMPUTE solutions at the red squares, all the grey points are redundant.

MMMF has two different tuning parameters r' and λ, both of which are related to the rank or spectral properties of the matrices U, V. SOFT-IMPUTE 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 (see Section 9 for illustrations).
- In practice when neither the optimal values of r' and λ are known, a two-dimensional search (for example, by cross validation) is required to select suitable values.

Further discussions and connections between the tuning parameters and spectral properties of the matrices can be found in Burer and Monteiro (2005) and Abernethy et al. (2009).

The proof of Theorem 3 requires a lemma.

Lemma 6 *For any matrix Z, the following holds:*

$$||Z||_* = \min_{U,V:\, Z=UV^T} \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right). \tag{29}$$

If $\text{rank}(Z) = k \leq \min\{m,n\}$*, then the minimum above is attained at a factor decomposition* $Z = U_{m\times k}V_{n\times k}^T$.

Note that in the decomposition $Z = UV^T$ in (29) there is no constraint on the number of columns r of the factor matrices $U_{m\times r}$ and $V_{n\times r}$. Lemma 6 is stronger than similar results appearing in Rennie and Srebro (2005) and Abernethy et al. (2009) which establish (29) for $r = \min\{m,n\}$—we give a tighter estimate of the rank k of the underlying matrices. The proof is given in Appendix A.5.

8.1 Proof of Theorem 3

Part 1. For $r = \min(m,n)$, any matrix $Z_{m\times n}$ can be written in the form of $Z = UV^T$. The criterion (27) can be written as

$$\min_{U,V} \quad \tfrac{1}{2}||P_\Omega(X - UV^T)||_F^2 + \tfrac{\lambda}{2}(\|U\|_F^2 + \|V\|_F^2) \tag{30}$$

$$= \min_{U,V} \quad \tfrac{1}{2}||P_\Omega(X - UV^T)||_F^2 + \lambda\|UV^T\|_* \qquad \text{(by Lemma 6)}$$

$$= \min_{Z} \quad \tfrac{1}{2}||P_\Omega(X - Z)||_F^2 + \lambda\|Z\|_*. \tag{31}$$

The equivalence of the criteria in (30) and (31) completes the proof of part 1.
Part 2. Note that if we know that the solution $\hat{Z}^*$ to (28) with $\lambda = \lambda^*$ has rank r^*, then $\hat{Z}^*$ also solves

$$\min_{Z,\, \mathrm{rank}(Z)=r^*} \quad \tfrac{1}{2}||P_\Omega(X - Z)||_F^2 \quad + \lambda\|Z\|_*.$$

We now repeat the steps (30)—(31), restricting the rank r' of U and V to be $r' = r^*$, and the result follows. ■

9. Numerical Experiments and Comparisons

In this section we study the performance of SOFT-IMPUTE, its post-processed variants, and HARD-IMPUTE for noisy matrix completion problems. The examples assert our claim that the matrix reconstruction criterion (4) (Cai et al., 2008) is too rigid if one seeks good predictive models. We include the related procedures of Rennie and Srebro (2005) and Keshavan et al. (2009) in our comparisons.

The reconstruction algorithm OPTSPACE, described in Keshavan et al. (2009) considers criterion (1) (in the presence of noise). It uses the representation $Z = USV'$ (which need not correspond to the SVD). OPTSPACE alternates between estimating S and U,V (in a Grassmann manifold) for computing a rank-r decomposition $\hat{Z} = \hat{U}\hat{S}\hat{V}'$. It starts with a sparse SVD on a *clean* version of the observed matrix $P_\Omega(X)$. This is similar to the formulation of MMMF (27) as detailed in Section 8, without the squared Frobenius norm regularization on the components U,V.

To summarize, we study the following methods:

1. SOFT-IMPUTE–Algorithm 1;
2. SOFT-IMPUTE+–post-processing on the output of SOFT-IMPUTE, as in Section 7;
3. HARD-IMPUTE–Algorithm 2, starting with the output of SOFT-IMPUTE+;
4. SVT–algorithm by Cai et al. (2008);
5. OPTSPACE–reconstruction algorithm by Keshavan et al. (2009);
6. MMMF–algorithm for (6) as in Rennie and Srebro (2005).

In all our simulation studies we use the underlying model $Z_{m\times n} = U_{m\times r}V'_{r\times n} + \varepsilon$, where U and V are random matrices with standard normal Gaussian entries, and ε 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 (Candès and Tao, 2009; Keshavan et al., 2009). The signal to noise ratio for the model and the test-error (standardized) are defined as

$$\text{SNR} = \sqrt{\frac{\text{var}(UV')}{\text{var}(\varepsilon)}}; \quad \text{Test Error} = \frac{\|P_\Omega^\perp(UV' - \hat{Z})\|_F^2}{\|P_\Omega^\perp(UV')\|_F^2}.$$

Training error (standardized) is defined as

$$\text{Training Error} = \frac{\|P_\Omega(Z - \hat{Z})\|_F^2}{\|P_\Omega(Z)\|_F^2},$$

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

Figures 2, 3 and 4 show training and test error for all of 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). In all examples $(m,n) = (100,100)$. For MMMF we use $r' = \min(m,n) = 100$, the number of columns in U and V. The performance of MMMF is displayed only in the plots with the nuclear norm along the horizontal axis, since the algorithm does not deliver a precise rank. 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 versus 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 routines not our own we use the MATLAB code as supplied on webpages by the authors. For SVT second author of Cai et al. (2008), for OPTSPACE third author of Keshavan et al. (2009), and for MMMF first author of Rennie and Srebro (2005).

9.1 Observations

The captions of each of Figures 2–4 detail the results, which we summarize here. For the first two figures, the noise is quite high with SNR= 1, and 50% of the entries are missing. In Figure 2 the true rank is 10, while in Figure 3 it is 6. SOFT-IMPUTE, MMMF and SOFT-IMPUTE+ have the best prediction performance, while SOFT-IMPUTE+ is better at estimating the correct rank. The other procedures perform poorly here, although OPTSPACE improves somewhat in Figure 3. SVT has very poor prediction error, suggesting once again that exactly fitting the training data is far too rigid. SOFT-IMPUTE+ has the best performance in Figure 3 (smaller rank—more aggressive fitting), and HARD-IMPUTE starts recovering here. In both figures the training error for SOFT-IMPUTE (and hence MMMF) wins as a function of nuclear norm (as it must, by construction), but the more aggressive fitters SOFT-IMPUTE+ and HARD-IMPUTE have better training error as a function of rank.

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.

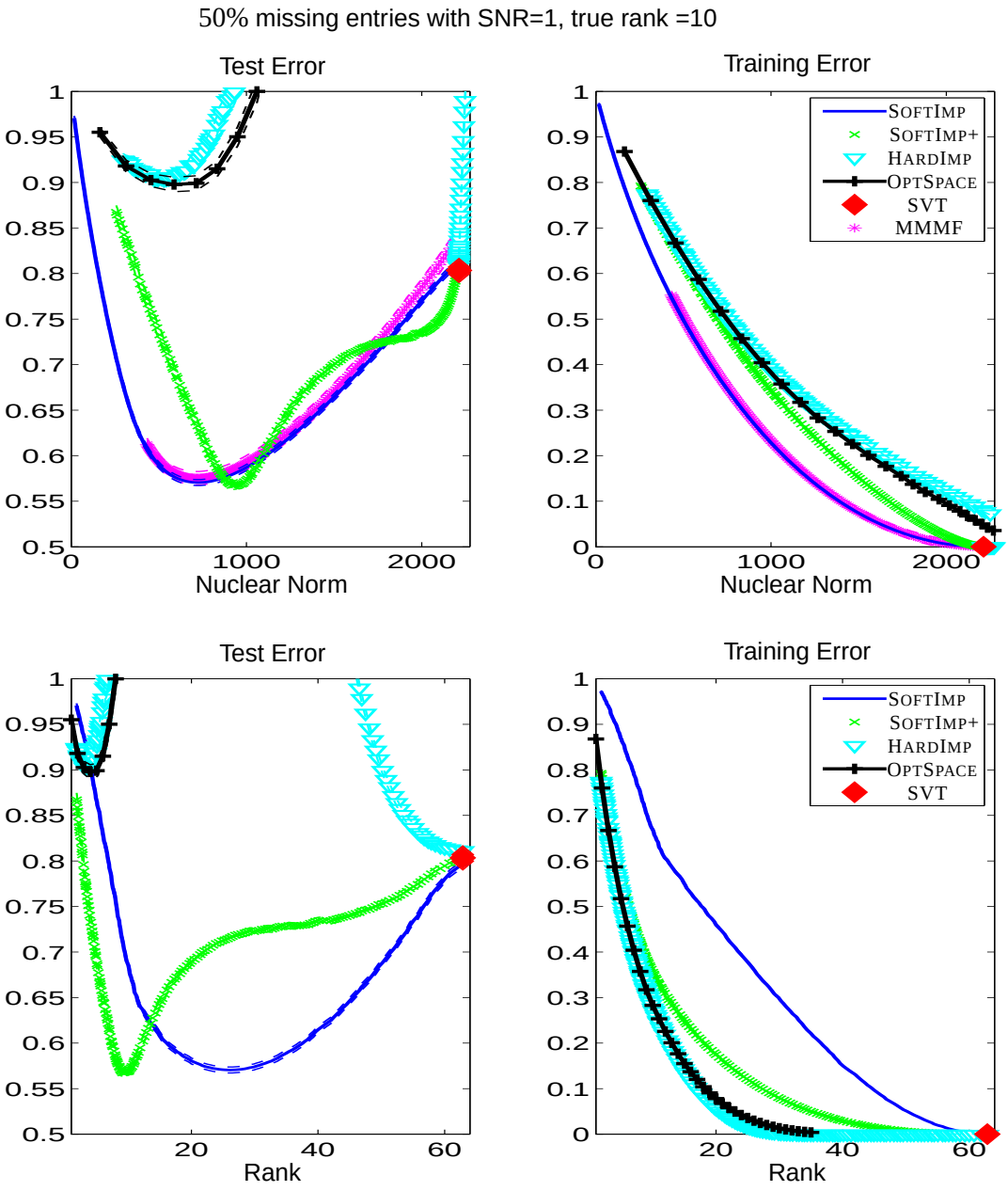

Figure 2: SOFTIMP+ refers to the post-processing after SOFT-IMPUTE; HARD-IMPUTE uses SOFT-IMP+ as starting values. Both SOFT-IMPUTE and SOFT-IMPUTE+ perform well (prediction error) in the presence of noise; the latter estimates the actual rank of the matrix. MMMF (with full rank 100 factor matrices) has performance similar to SOFT-IMPUTE. HARD-IMPUTE and OPTSPACE show poor prediction error. SVT also has poor prediction error, confirming our claim in this example that criterion (4) can result in 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.

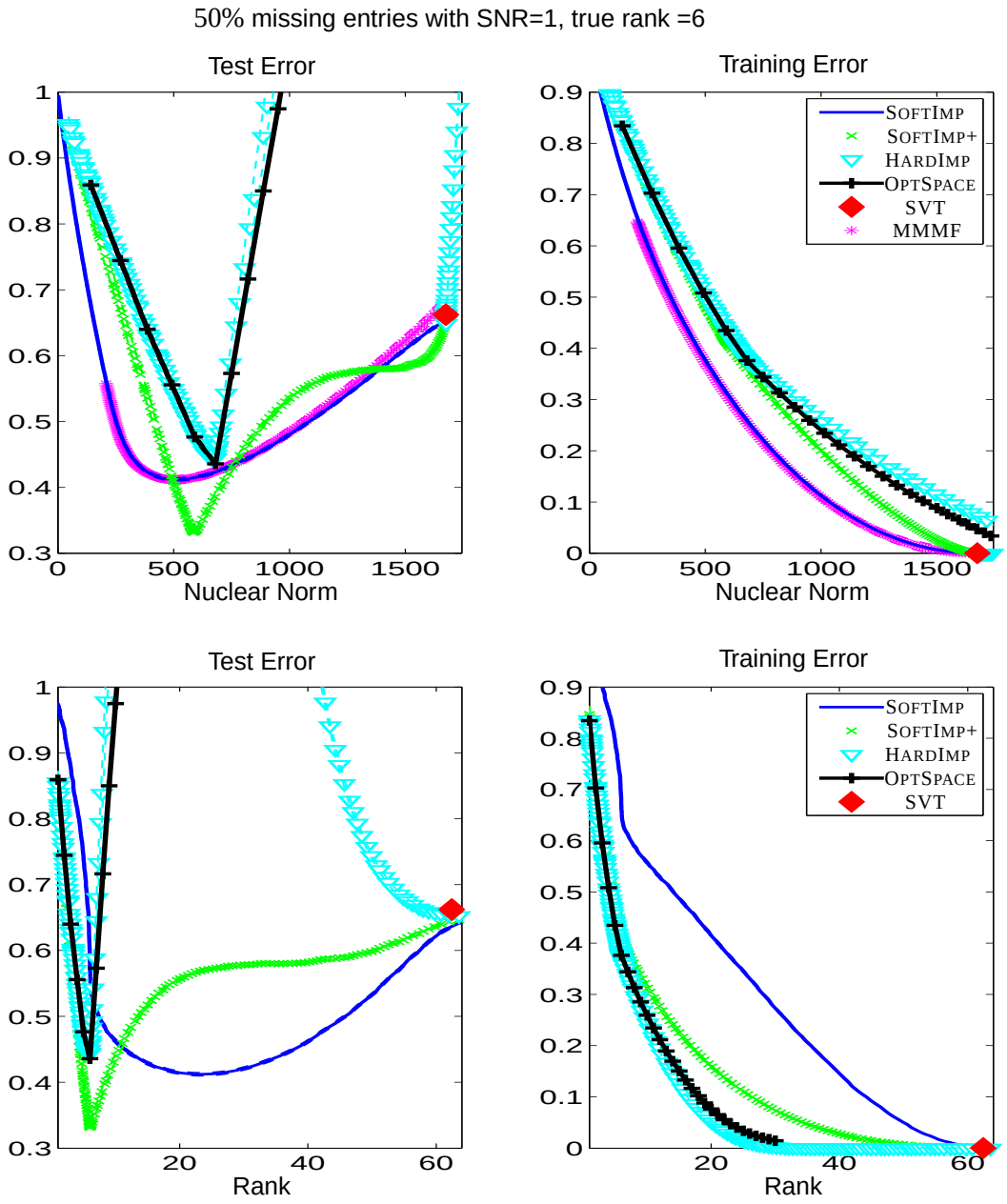

Figure 3: SOFT-IMPUTE+ has the best prediction error, closely followed by SOFT-IMPUTE and MMMF. Both HARD-IMPUTE and OPTSPACE have poor prediction error apart from near the true rank 6 of the matrix, where they show reasonable performance. SVT has very poor prediction error; it recovers a matrix with high nuclear norm and rank > 60, where the true rank is only 6. OPTSPACE is evaluated for a series of ranks ≤ 35.

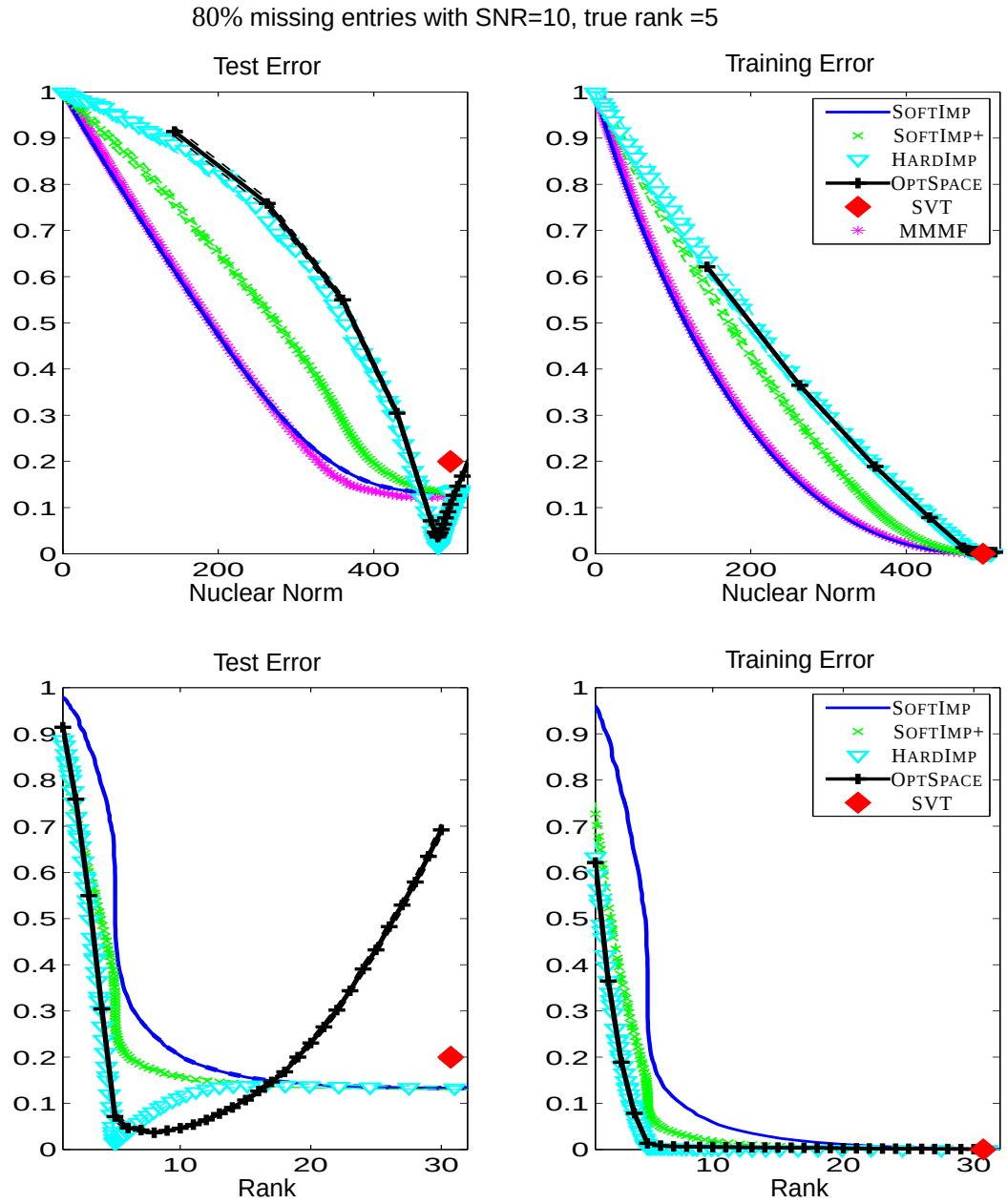

Figure 4: With low noise the performance of HARD-IMPUTE improves. It gets the correct rank whereas OPTSPACE slightly overestimates the rank. HARD-IMPUTE has the best prediction error, followed by OPTSPACE. Here MMMF has slightly better prediction error than SOFT-IMPUTE. Although the noise is low here, SVT recovers a matrix with high rank (approximately 30) and has poor prediction error as well. The test error of SVT is found to be different from the limiting solution of SOFT-IMPUTE; although in theory the limiting solution of (10) should coincide with that of SVT, in practice we never go to the limit.

In Figure 4 with SNR= 10 the noise is relatively small compared to the other two cases. The true underlying rank is 5, but the proportion of missing entries is much higher at eighty percent. Test errors of both SOFT-IMPUTE+ 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. MMMF has slightly better test error than SOFT-IMPUTE around a nuclear norm of 350, while in theory they should be identical. Notice, however, that the training error is slightly worse (everywhere), suggesting that MMMF is sometimes trapped in local minima. The fact that this slightly underfit solution does better in test error is a quirk of this particular example. OPTSPACE performs well in this high-SNR example, achieving a sharp minima at the true rank of the matrix. HARD-IMPUTE performs the best in this example. The better performance of both OPTSPACE and HARD-IMPUTE over SOFT-IMPUTE can be attributed both to the low-rank truth and the high SNR. This is reminiscent of the better predictive performance of best-subset or concave penalized regression often seen over LASSO in setups where the underlying model is very sparse (Friedman, 2008).

9.2 Comparison with Fast MMMF (Rennie and Srebro, 2005)

In this section we compare SOFT-IMPUTE with MMMF in terms of computational efficiency. We also examine the consequences of two regularization parameters (r', λ) for MMMF over one for SOFT-IMPUTE.

Rennie and Srebro (2005) describes a fast algorithm based on conjugate-gradient descent for minimization of the MMMF criterion (6). With (6) being non-convex, it is hard to provide theoretical optimality guarantees for the algorithm for arbitrary r', λ—that is, what type of solution it converges to or how far it is from the global minimizer.

In Table 1 we summarize the performance results of the two algorithms. For both SOFT-IMPUTE and MMMF we consider a equi-spaced grid of 150 $\lambda \in [\lambda_{\min}, \lambda_{\max}]$, with $\lambda_{\min}$ corresponding to a full-rank solution of SOFT-IMPUTE and $\lambda_{\max}$ the zero solution. For MMMF, three different values of r' were used, and for each $(\hat{U}, \hat{V})$ were solved for over the grid of λ values. A separate held-out validation set with twenty percent of the missing entries sampled from $\Omega^{\perp}$ were used to train the tuning parameter λ (for each value of r') for MMMF and SOFT-IMPUTE. Finally we evaluate the standardized prediction errors on a test set consisting of the remaining eighty percent of the missing entries in $\Omega^{\perp}$. In all cases we report the training errors and test errors on the optimally tuned λ. SOFT-IMPUTE was run till a tolerance of 10^{-4} was achieved (fraction of decrease of objective value). Likewise for MMMF we set the tolerance of the conjugate gradient method to 10^{-4}.

In Table 1, for every algorithm total time indicates the time required for evaluating solutions over the entire grid of λ values. In these examples, we used direct SVD factorization based methods for the SVD computation, since the size of the problems were quite small. In all these examples we observe that SOFT-IMPUTE performs very favorably in terms of total times. For MMMF the time to train the models increase with increasing rank r'; and in case the underlying matrix has rank which is larger than r', the computational cost will be large in order to get competitive predictive accuracy. This point is supported in the examples of Table 1. It is important to note that, the prediction error of SOFT-IMPUTE as obtained on the validation set is actually within standard error of the best prediction error produced by all the MMMF models. In addition we also performed some medium-scale examples increasing the dimensions of the matrices. To make comparisons fair, SOFT-IMPUTE made use of *direct* SVD computations (in MATLAB) instead of iterative algorithms

Data	Method (rank)	Test error	Training error	Time (secs)
$(m,n) = (10^2, 10^2)$	SOFT-IMPUTE (39)	0.7238 (0.0027)	0.0720	4.1745
$\|\Omega\| = 5 \times 10^3 (50\%)$	MMMF (20)	0.7318 (0.0031)	0.2875	48.1090
SNR= 3	MMMF (60)	0.7236 (0.0027)	0.1730	62.0230
rank (R)= 30	MMMF (100)	0.7237 (0.0027)	0.1784	96.2750
$(m,n) = (10^2, 10^2)$	SOFT-IMPUTE (37)	0.5877 (0.0047)	0.0017	4.0976
$\|\Omega\| = 2 \times 10^3 (20\%)$	MMMF (20)	0.5807 (0.0049)	0.0186	53.7533
SNR= 10	MMMF (60)	0.5823 (0.0049)	0.0197	62.0230
rank(R)= 10	MMMF (100)	0.5823 (0.0049)	0.0197	84.0375
$(m,n) = (10^2, 10^2)$	SOFT-IMPUTE (59)	0.6008 (0.0028)	0.0086	3.8447
$\|\Omega\| = 8 \times 10^3 (80\%)$	MMMF (20)	0.6880 (0.0029)	0.4037	33.8685
SNR= 10	MMMF (60)	0.5999 (0.0028)	0.0275	57.3488
rank(R)= 45	MMMF (100)	0.5999 (0.0028)	0.0275	89.4525

Table 1: Performances of SOFT-IMPUTE and MMMF for different problem instances, in terms of test error (with standard errors in parentheses), training error and times for learning the models. SOFT-IMPUTE,"rank" denotes the rank of the recovered matrix, at the optimally chosen value of λ. For the MMMF, "rank" indicates the value of r' in $U_{m\times r'}, V_{n\times r'}$. Results are averaged over 50 simulations.

exploiting the specialized *Sparse+Low-Rank* structure (22). We report our findings on one such simulation example:

- For $(m,n) = (2000, 1000)$, $|\Omega|/(m \cdot n) = 0.2$, rank $= 500$ and SNR=10; SOFT-IMPUTE takes 1.29 hours to compute solutions on a grid of 100 λ values. The test error on the validation set and training error are 0.9630 and 0.4375 with the recovered solution having a rank of 225.

  For the same problem, MMMF with $r' = 200$ takes 6.67 hours returning a solution with test-error 0.9678 and training error 0.6624. With $r' = 400$ it takes 12.89 hrs with test and training errors 0.9659 and 0.6564 respectively.

We will like to note that DeCoste (2006) proposed an efficient implementation of MMMF via an ensemble based approach, which is quite different in spirit from the batch optimization algorithms we are studying in this paper. Hence we do not compare it with SOFT-IMPUTE.

9.3 Comparison with Nesterov's Accelerated Gradient Method

Ji and Ye (2009) proposed a first-order algorithm based on Nesterov's acceleration scheme (Nesterov, 2007), for nuclear norm minimization for a generic multi-task learning problem (Argyriou et al., 2008, 2007). Their algorithm (Liu et al., 2009; Ji and Ye, 2009) can be adapted to the SOFT-IMPUTE problem (10); hereafter we refer to it as NESTEROV. It requires one to compute the SVD of a dense matrix having the dimensions of X, which makes it prohibitive for large-scale problems. We instead would make use of the structure (22) for the SVD computation, a special characteristic of matrix completion which is not present in a generic multi-task learning problem. Here we compare the performances of SOFT-IMPUTE and NESTEROV on small scale examples, where direct SVDs can be computed easily.

Since both algorithms solve the same criterion, the quality of the solutions—objective values, training and test errors—will be the same (within tolerance). We hence compare their performances based on the times taken by the algorithms to converge to the optimal solution of (10) on a grid of values of λ. Both algorithms compute a path of solutions using warm starts. Results are shown in Figure 5, for four different scenarios described in Table 2.

Example	(m,n)	$\|\Omega\|/(m\cdot n)$	Rank	Test Error
i	$(100,100)$	0.5	5	0.4757
ii	$(100,100)$	0.2	5	0.0668
iii	$(100,100)$	0.8	5	0.0022
iv	$(1000,500)$	0.5	30	0.0028

Table 2: Four different examples used for timing comparisons of SOFT-IMPUTE and NESTEROV (accelerated Nesterov algorithm of Ji and Ye 2009). In all cases the SNR= 10.

Figure 5 shows the time to convergence for the two algorithms. Their respective number of iterations are not comparable. This is because NESTEROV uses a line-search to compute an adaptive step-size (approximate the Lipschitz constant) at every iteration, whereas SOFT-IMPUTE does not.

SOFT-IMPUTE has a rate of convergence given by Theorem 2, which for large k is worse than the accelerated version NESTEROV with rate $O(1/k^2)$. However, timing comparisons in Figure 5 show that SOFT-IMPUTE performs very favorably. We do not know the exact reason behind this, but mention some possibilities. Firstly the rates are *worst case* convergence rates. On particular problem instances of the form (10), the rates of convergence in *practice* of SOFT-IMPUTE and NESTEROV may be quite similar. Since Ji and Ye (2009) uses an adaptive step-size strategy, the choice of a step-size may be time consuming. SOFT-IMPUTE on the other hand, uses a constant step size.

Additionally, it appears that the use of the *momentum* term in NESTEROV affects the *Sparse+Low-rank* decomposition (22). This may prevent the algorithm to be adapted for solving large problems, due to costly SVD computations.

9.4 Large Scale Simulations for SOFT-IMPUTE

Table 3 reports the performance of SOFT-IMPUTE on some large-scale problems. All computations are performed in MATLAB and the MATLAB implementation of PROPACK is used. Data input, access and transfer in MATLAB take a sizable portion of the total computational time, given the size of these problems. However, the main computational bottle neck in our algorithm is the structured SVD computation. In order to focus more on the essential computational task, Table 3 displays the total time required to perform the SVD computations over all iterations of the algorithm. Note that for all the examples considered in Table 3, the implementations of algorithms NESTEROV (Liu et al., 2009; Ji and Ye, 2009) and MMMF (Rennie and Srebro, 2005) are prohibitively expensive both in terms of computational time and memory requirements, and hence could not be run. We used the value $\lambda = ||P_\Omega(X)||_2/K$ with SOFT-IMPUTE, with $K = 1.5$ for all examples but the last, where $K = 2$. $\lambda_0 = ||P_\Omega(X)||_2$ is the largest singular value of the input matrix X (padded with zeros); this is the smallest value of λ for which $\mathbf{S}_{\lambda_0}(P_\Omega(X)) = 0$ in the first iteration of SOFT-IMPUTE (Section 3).

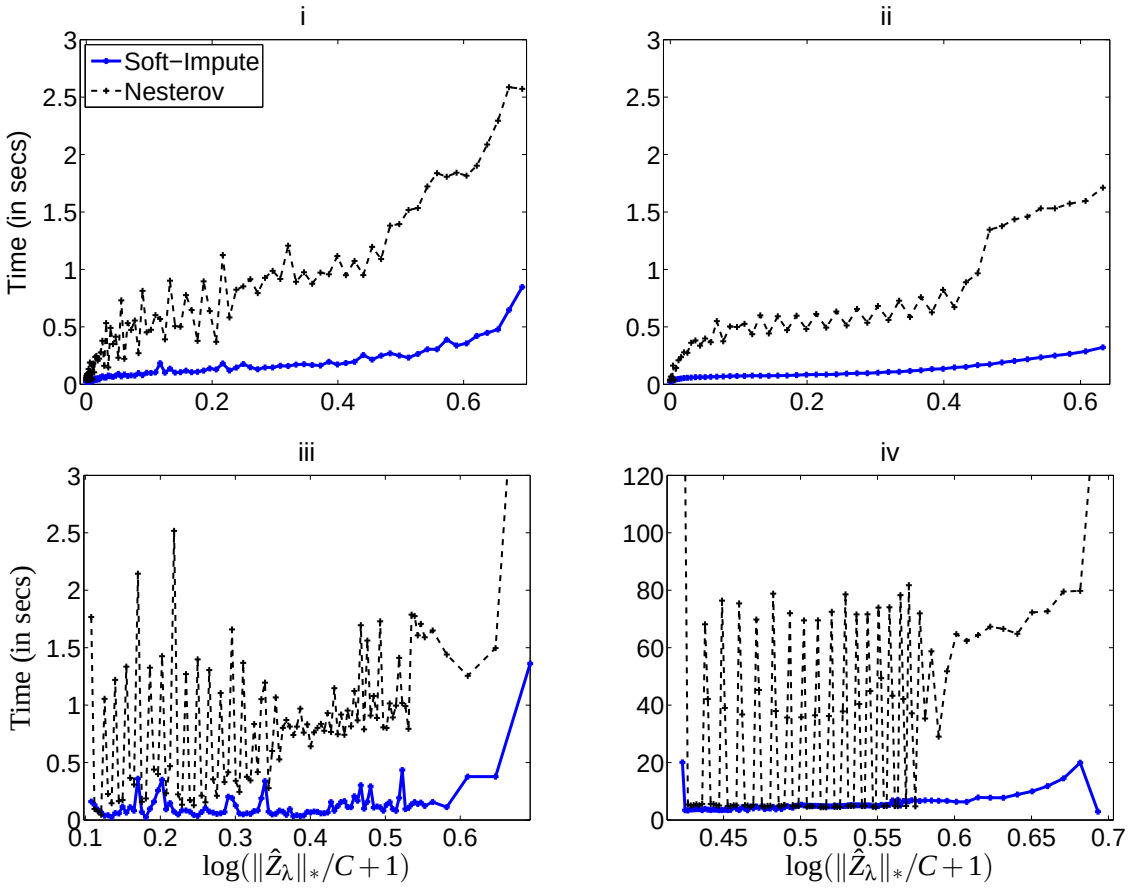

Figure 5: Timing comparisons of SOFT-IMPUTE and NESTEROV (accelerated Nesterov algorithm of Ji and Ye 2009). The horizontal axis corresponds to the standardized nuclear norm, with $C = \max_\lambda \|\hat{Z}_\lambda\|_*$. Shown are the times till convergence for the two algorithms over an entire grid of λ values for examples i–iv (in the last the matrix dimensions are much larger). The overall time differences between Examples i–iii and Example iv is due to the increased cost of the SVD computations. Results are averaged over 10 simulations. The times for NESTEROV change far more erratically with λ than they do for SOFT-IMPUTE.

The prediction performance is awful for all but one of the models, because in most cases the fraction of observed data is very small. These simulations were mainly to show the computational capabilities of SOFT-IMPUTE on very large problems.

10. Application to the Netflix Data Set

The Netflix training data consists of the ratings of 17,770 movies by 480,189 Netflix customers. The resulting data matrix is extremely sparse, with 100,480,507 or 1% of the entries observed. The task was 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 data sets held in secret by Netflix. A probe set of about 1.4 million

(m,n)	$\|\Omega\|$	$\|\Omega\|/(m\cdot n)$ $\times 100\%$	Recovered rank	Time (mins)	Test error	Training error
$(10^4,10^4)$	10^5	0.1	40^*	0.2754	0.9946	0.6160
$(10^4,10^4)$	10^5	0.1	40^*	0.3770	0.9959	0.6217
$(10^4,10^4)$	10^5	0.1	50^*	0.4292	0.9962	0.3862
$(10^4,10^4)$	10^6	1.0	5	1.6664	0.6930	0.6600
$(10^5,10^5)$	5×10^6	0.05	40^*	17.2518	0.9887	0.8156
$(10^5,10^5)$	10^7	0.1	5	20.3142	0.9803	0.9761
$(10^6,10^5)$	10^8	0.1	5	259.9620	0.9913	0.9901
$(10^6,10^6)$	10^7	0.001	20^*	99.6780	0.9998	0.5834

Table 3: Performance of SOFT-IMPUTE on different problem instances. All models are generated with SNR=10 and underlying rank=5. Recovered rank is the rank of the solution matrix $\hat{Z}$ at the value of λ used in (10). Those with stars reached the "maximum rank" threshold, and option in our algorithm. Convergence criterion is taken as "fraction of improvement of objective value" less than 10^{-4} or a maximum of 15 iterations for the last four examples. All implementations are done in MATLAB including the MATLAB implementation of PROPACK on a Intel Xeon Linux 3GHz processor.

ratings was 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 was to improve this by 10%, or an RMSE of 0.8572 or better. The contest ran for about 3 years, and the winning team was "Bellkor's Pragmatic Chaos", a merger of three earlier teams (see `http://www.netflixprize.com/` for details). They claimed the grand prize of $1M on September 21, 2009.

Many of the competitive algorithms build on a regularized low-rank factor model similar to (6) using randomization schemes like mini-batch, stochastic gradient descent or sub-sampling to reduce the computational cost over making several passes over the entire data-set (see Salakhutdinov et al., 2007; Bell and Koren., 2007; Takacs et al., 2009, for example). In this paper, our focus is not on using randomized or sub-sampling schemes. Here we demonstrate that our nuclear-norm regularization algorithm can be applied in batch mode on the entire Netflix training set with a reasonable computation time. We note however that the conditions under which the nuclear-norm regularization is theoretically meaningful (Candès and Tao, 2009; Srebro et al., 2005a) are not met on the Netflix data set.

We applied SOFT-IMPUTE to the training data matrix and then performed a least-squares unshrinking on the singular values with the singular vectors and the training data row and column means as the bases. The latter was performed on a data-set of size 10^5 randomly drawn from the probe set. The prediction error (RMSE) is obtained on a left out portion of the probe set. Table 4 reports the performance of the procedure for different choices of the tuning parameter λ (and the corresponding rank); times indicate the total time taken for the SVD computations over all iterations. A maximum of 10 iterations were performed for each of the examples. Again, these results are not competitive with those of the competition leaders, but rather demonstrate the feasibility of applying SOFT-IMPUTE to such a large data set.

λ	Rank	Time (hrs)	RMSE
$\lambda_0/250$	42	1.36	0.9622
$\lambda_0/300$	66	2.21	0.9572
$\lambda_0/500$	81	2.83	0.9543
$\lambda_0/600$	95	3.27	0.9497

Table 4: Results of applying SOFT-IMPUTE to the Netflix data. $\lambda_0 = ||P_\Omega(X)||_2$; see Section 9.4. 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 root-mean squared error, as defined in the text.

Acknowledgments

We thank the reviewers for their suggestions that lead to improvements in this paper. We thank Stephen Boyd, Emmanuel Candes, Andrea Montanari, 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. Proofs

We begin with the proof of Lemma 1.

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 without loss of generality, $m \geq n$. We will explicitly evaluate the closed form solution of the problem (8). Note that

$$\frac{1}{2}\|Z-W\|_F^2 + \lambda\|Z\|_* = \frac{1}{2}\left\{\|Z\|_F^2 - 2\sum_{i=1}^n \tilde{d}_i\tilde{u}_i' W\tilde{v}_i + \sum_{i=1}^n \tilde{d}_i^2\right\} + \lambda\sum_{i=1}^n \tilde{d}_i \tag{32}$$

where

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

Minimizing (32) is equivalent to minimizing

$$-2\sum_{i=1}^n \tilde{d}_i\tilde{u}_i' W\tilde{v}_i + \sum_{i=1}^n \tilde{d}_i^2 + \sum_{i=1}^n 2\lambda\tilde{d}_i;\ \text{ w.r.t. } (\tilde{u}_i,\tilde{v}_i,\tilde{d}_i),\ i=1,\ldots,n,$$

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 (w.r.t. $\tilde{U},\tilde{V}$) the function $Q(\tilde{U},\tilde{V})$:

$$Q(\tilde{U},\tilde{V}) = \min_{\tilde{D}\geq 0}\ \ \frac{1}{2}\left\{-2\sum_{i=1}^n \tilde{d}_i\tilde{u}_i' W\tilde{v}_i + \sum_{i=1}^n \tilde{d}_i^2\right\} + \lambda\sum_{i=1}^n \tilde{d}_i. \tag{33}$$

Since the objective (33) to be minimized w.r.t. $\tilde{D}$, is separable in $\tilde{d}_i, i = 1,\ldots,n$; it suffices to minimize it w.r.t. each $\tilde{d}_i$ separately.

The problem

$$\underset{\tilde{d}_i \geq 0}{\text{minimize}} \frac{1}{2}\left\{-2\tilde{d}_i\tilde{u}_i' W\tilde{v}_i + \tilde{d}_i^2\right\} + \lambda\tilde{d}_i \tag{34}$$

can be solved looking at the stationary conditions of the function using its sub-gradient (Nesterov, 2003). 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 $\tilde{u}_i' W\tilde{v}_i$ (without loss of generality, we can take $\tilde{u}_i' W\tilde{v}_i$ to be non-negative). More generally the soft-thresholding operator (Friedman et al., 2007; Hastie et al., 2009) is given by $S_\lambda(x) = \text{sgn}(x)(|x| - \lambda)_+$. See Friedman et al. (2007) 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 (34) in (33) we get

$$Q(\tilde{U},\tilde{V}) = \frac{1}{2}\left\{\|Z\|_F^2 - 2\sum_{i=1}^{n}(\tilde{u}_i' W\tilde{v}_i - \lambda)_+(\tilde{u}_i' W\tilde{v}_i - \lambda) + (\tilde{u}_i' X\tilde{v}_i - \lambda)_+^2\right\}. \tag{35}$$

Minimizing $Q(\tilde{U},\tilde{V})$ w.r.t. $(\tilde{U},\tilde{V})$ is equivalent to maximizing

$$\sum_{i=1}^{n}\left\{2(\tilde{u}_i' W\tilde{v}_i - \lambda)_+(\tilde{u}_i' W\tilde{v}_i - \lambda) - (\tilde{u}_i' W\tilde{v}_i - \lambda)_+^2\right\} = \sum_{\tilde{u}_i' W\tilde{v}_i > \lambda}(\tilde{u}_i' W\tilde{v}_i - \lambda)^2. \tag{36}$$

It is a standard fact that for every i the problem

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

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 (36) 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 (36) 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 (34) 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 (8) is given by $UD_\lambda V'$ as in (9). ■

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 (34) will be modified accordingly. 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*

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

The optimization problem analogous to (35) will be

$$\underset{\tilde{U},\tilde{V}}{\text{minimize}} \;\; \frac{1}{2}\left\{\|Z\|_F^2 - 2\sum_{i=1}^{n}\hat{d}_i\tilde{u}_i' W\tilde{v}_i + \sum_{i=1}^{n}\hat{d}_i^2\right\} + \lambda\sum_i \mathbf{p}(\hat{d}_i), \tag{37}$$

where $\hat{d_i} = S^{\mathbf{p}}_{\lambda}(\tilde{u}_i' W \tilde{v}_i), \forall i$. *Any spectral function for which the above (37) 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* $S^{\mathbf{p}}_{\lambda}(\cdot)$ *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 and is inspired by some techniques used in Cai et al. (2008).

Proof From Lemma 1, we know that if $\hat{Z}$ solves the problem (8), then it satisfies the sub-gradient stationary conditions:

$$0 \in -(W - \hat{Z}) + \lambda \partial \|\hat{Z}\|_*. \tag{38}$$

$\mathbf{S}_\lambda(W_1)$ and $\mathbf{S}_\lambda(W_2)$ solve the problem (8) with $W = W_1$ and $W = W_2$ respectively, hence (38) 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

$$\partial \|Z\|_* = \{UV' + \omega : \omega_{m \times n},\ U'\omega = 0,\ \omega V = 0,\ \|\omega\|_2 \leq 1\}, \tag{39}$$

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

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

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

The above gives

$$(\hat{Z}_1 - \hat{Z}_2) - (W_1 - W_2) + \lambda(p(\hat{Z}_1) - p(\hat{Z}_2)) = 0, \tag{40}$$

from which we obtain

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

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

Now observe that

$$\langle p(\hat{Z}_1) - p(\hat{Z}_2), \hat{Z}_1 - \hat{Z}_2 \rangle = \langle p(\hat{Z}_1), \hat{Z}_1 \rangle - \langle p(\hat{Z}_1), \hat{Z}_2 \rangle - \langle p(\hat{Z}_2), \hat{Z}_1 \rangle + \langle p(\hat{Z}_2), \hat{Z}_2 \rangle. \tag{41}$$

By the characterization of subgradients as in (39), we have

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

This 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 (41) we obtain:

$$\begin{aligned} \langle p(\hat{Z}_1), \hat{Z}_1 \rangle + \langle p(\hat{Z}_2), \hat{Z}_2 \rangle &= \|\hat{Z}_1\|_* + \|\hat{Z}_2\|_* & (42) \\ -\langle p(\hat{Z}_1), \hat{Z}_2 \rangle - \langle p(\hat{Z}_2), \hat{Z}_1 \rangle &\geq -\|\hat{Z}_2\|_* - \|\hat{Z}_1\|_*. & (43) \end{aligned}$$

Using (42,43) we see that the r.h.s. of (41) 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 (40), we obtain:

$$\|\hat{Z}_1 - \hat{Z}_2\|_F^2 = \langle \hat{Z}_1 - \hat{Z}_2, \hat{Z}_1 - \hat{Z}_2 \rangle \leq \langle W_1 - W_2, \hat{Z}_1 - \hat{Z}_2 \rangle. \tag{44}$$

Using the Cauchy-Schwarz Inequality, $\|\hat{Z}_1 - \hat{Z}_2\|_F \|W_1 - W_2\|_F \geq \langle \hat{Z}_1 - \hat{Z}_2, W_1 - W_2 \rangle$ in (44) we get

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

and in particular

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

The above 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 (15) by observing the following inequalities:

$$\begin{aligned}
\|Z_\lambda^{k+1} - Z_\lambda^k\|_F^2 &= \|\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)) - \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}))\|_F^2 \\
\text{(by Lemma 3)} \quad &\leq \| \left(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^k)\right) - \left(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1})\right) \|_F^2 \\
&= \|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^{k-1})\|_F^2 \qquad (45) \\
&\leq \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2. \qquad (46)
\end{aligned}$$

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 (45,46) gives

$$\|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^{k-1})\|_F^2 - \|Z_\lambda^k - Z_\lambda^{k-1}\|_F^2 \to 0 \implies P_\Omega(Z_\lambda^k - Z_\lambda^{k-1}) \to 0, \tag{47}$$

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 (13,14) 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 (47) 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

$$\partial\|Z\|_* = \{UV' + W : W_{m\times n},\ U'W = 0,\ WV = 0,\ \|W\|_2 \le 1\}, \tag{48}$$

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

$$0 \in -(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{k-1}) - Z_\lambda^k) + \partial\|Z_\lambda^k\|_* \ \ \forall k. \tag{49}$$

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,

$$(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{n_k-1}) - Z_\lambda^{n_k}) \longrightarrow (P_\Omega(X) - P_\Omega(Z_\lambda^*)). \tag{50}$$

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 $\partial\|Z_\lambda^k\|_*$ (48).

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 $u_{n_k}' v_{n_k} \to u_\infty' v_\infty$. Furthermore due to boundedness (passing on to a further subsequence if necessary) $w_{n_k} \to w_\infty$. 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 (49, 50), passing on to the limits along the subsequence n_k, we have

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

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

We shall now prove (16). We know that for every n_k

$$Z_\lambda^{n_k} = \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{n_k-1})). \tag{51}$$

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

$$\mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^{n_k-1})) \to \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^*)).$$

Thus passing over to the limits on both sides of (51) we get

$$Z_\lambda^* = \mathbf{S}_\lambda(P_\Omega(X) + P_\Omega^\perp(Z_\lambda^*)),$$

therefore completing the proof. ■

A.5 Proof of Lemma 6

The proof is motivated by the principle of embedding an arbitrary matrix into a positive semidefinite matrix (Fazel, 2002). We require the following proposition, which we prove using techniques used in the same reference.

Proposition 1 *Suppose matrices $W_{m\times m}, \tilde{W}_{n\times n}, Z_{m\times n}$ satisfy the following:*

$$\begin{pmatrix} W & Z \\ Z^T & \tilde{W} \end{pmatrix} \succeq 0.$$

Then $\text{trace}(W) + \text{trace}(\tilde{W}) \geq 2\|Z\|_*$.

Proof Let $Z_{m\times n} = L_{m\times r}\Sigma_{r\times r}R_{n\times r}^T$ denote the SVD of Z, where r is the rank of the matrix Z. Observe that the trace of the product of two positive semidefinite matrices is always non-negative. Hence we have the following inequality:

$$\text{trace}\begin{pmatrix} LL^T & -LR^T \\ -RL^T & RR^T \end{pmatrix}\begin{pmatrix} W & Z \\ Z^T & \tilde{W} \end{pmatrix} \succeq 0.$$

Simplifying the above expression we get:

$$\text{trace}(LL^TW) - \text{trace}(LR^TZ^T) - \text{trace}(RL^TZ) + \text{trace}(RR^T\tilde{W}) \geq 0. \tag{52}$$

Due to the orthogonality of the columns of L, R we have the following inequalities:

$$\text{trace}(LL^TW) \leq \text{trace}(W) \text{ and } \text{trace}(RR^T\tilde{W}) \leq \text{trace}(\tilde{W}).$$

Furthermore, using the SVD of Z:

$$\text{trace}(LR^TZ^T) = \text{trace}(\Sigma) = \text{trace}(LR^TZ^T).$$

Using the above in (52), we have:

$$\text{trace}(W) + \text{trace}(\tilde{W}) \geq 2\,\text{trace}(\Sigma) = 2\,\|Z\|_*.$$

■

Proof [Proof of Lemma 6.] For the matrix Z, consider any decomposition of the form $Z = \tilde{U}_{m\times r}\tilde{V}^T_{n\times r}$ and construct the following matrix

$$\begin{pmatrix} \tilde{U}\tilde{U}^T & Z \\ Z^T & \tilde{V}\tilde{V}^T \end{pmatrix} = \begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix} (\tilde{U}^T\tilde{V}^T), \tag{53}$$

which is positive semidefinite. Applying Proposition 1 to the left hand matrix in (53), we have:

$$\text{trace}(\tilde{U}\tilde{U}^T) + \text{trace}(\tilde{V}\tilde{V}^T) \geq 2\|Z\|_*.$$

Minimizing both sides above w.r.t. the decompositions $Z = \tilde{U}_{m\times r}\tilde{V}^T_{n\times r}$; we have

$$\min_{\tilde{U},\tilde{V};\, Z=\tilde{U}\tilde{V}^T} \left\{\text{trace}(\tilde{U}\tilde{U}^T) + \text{trace}(\tilde{V}\tilde{V}^T)\right\} \geq 2\|Z\|_*. \tag{54}$$

Through the SVD of Z we now show that equality is attained in (54). Suppose Z is of rank $k \leq \min(m,n)$, and denote its SVD by $Z_{m\times n} = L_{m\times k}\Sigma_{k\times k}R^T_{n\times k}$. Then for $\tilde{U} = L_{m\times k}\Sigma^{\frac{1}{2}}_{k\times k}$ and $\tilde{V} = R_{n\times k}\Sigma^{\frac{1}{2}}_{k\times k}$ the equality in (54) is attained.

Hence, we have:

$$\begin{aligned} \|Z\|_* &= \min_{\tilde{U},\tilde{V};Z=\tilde{U}\tilde{V}^T} \left\{\text{trace}(\tilde{U}\tilde{U}^T) + \text{trace}(\tilde{V}\tilde{V}^T)\right\} \\ &= \min_{\tilde{U},\tilde{V};Z=\tilde{U}_{m\times k}\tilde{V}^T_{n\times k}} \left\{\text{trace}(\tilde{U}\tilde{U}^T) + \text{trace}(\tilde{V}\tilde{V}^T)\right\}. \end{aligned}$$

Note that the minimum can also be attained for matrices with $r \geq k$ or even $r \geq \min(m,n)$; however, it suffices to consider matrices with $r = k$. Also it is easily seen that the minimum cannot be attained for any $r < k$; hence the minimal rank r for which (29) holds true is $r = k$. ■

A.6 Proof of Theorem 2

There is a close resemblance between SOFT-IMPUTE and Nesterov's gradient method (Nesterov, 2007, Section 3). However, as mentioned earlier the original motivation of our algorithm is very different.

The techniques used in this proof are adapted from Nesterov (2007).

Proof Plugging $Z^k_\lambda = \tilde{Z}$ in (11), we have

$$\begin{aligned} Q_\lambda(Z|Z^k_\lambda) &= f_\lambda(Z^k_\lambda) + \frac{1}{2}\|P^\perp_\Omega(Z^k_\lambda - Z)\|^2_F \\ &\geq f_\lambda(Z^k_\lambda). \end{aligned} \tag{55}$$

Let $Z^k_\lambda(\theta)$ denote a convex combination of the optimal solution (Z^∞_λ) and the k^{th} iterate (Z^k_λ):

$$Z^k_\lambda(\theta) = \theta Z^\infty_\lambda + (1-\theta)Z^k_\lambda. \tag{56}$$

Using the convexity of $f_\lambda(\cdot)$ we get:

$$f_\lambda(Z_\lambda^k(\theta)) \leq (1-\theta) f_\lambda(Z_\lambda^k) + \theta f_\lambda(Z_\lambda^\infty). \tag{57}$$

Expanding $Z_\lambda^k(\theta)$ using (56), and simplifying $P_\Omega^\perp(Z_\lambda^k - Z_\lambda^k(\theta))$ we have:

$$\begin{aligned} \|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^k(\theta))\|_F^2 &= \theta^2 \|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^\infty)\|_F^2 \\ &\leq \theta^2 \|Z_\lambda^k - Z_\lambda^\infty\|_F^2 \qquad (58) \\ &\leq \theta^2 \|Z_\lambda^0 - Z_\lambda^\infty\|_F^2. \qquad (59) \end{aligned}$$

Line 59 follows from (58) by observing that $\|Z_\lambda^m - Z_\lambda^\infty\|_F^2 \leq \|Z_\lambda^{m-1} - Z_\lambda^\infty\|_F^2$, $\forall m$—a consequence of the inequalities (19) and (18), established in Theorem 1.

Using (55), the value of $f_\lambda(Z)$ at the $(k+1)^{\text{th}}$ iterate satisfies the following chain of inequalities:

$$\begin{aligned} f_\lambda(Z_\lambda^{k+1}) &\leq \min_Z \left\{ f_\lambda(Z) + \frac{1}{2}\|P_\Omega^\perp(Z_\lambda^k - Z)\|_F^2 \right\} \\ &\leq \min_{\theta\in[0,1]} \left\{ f_\lambda(Z_\lambda^k(\theta)) + \frac{1}{2}\|P_\Omega^\perp(Z_\lambda^k - Z_\lambda^k(\theta))\|_F^2 \right\} \qquad (60) \\ &\leq \min_{\theta\in[0,1]} \left\{ f_\lambda(Z_\lambda^k) + \theta(f_\lambda(Z_\lambda^\infty) - f_\lambda(Z_\lambda^k)) + \frac{1}{2}\theta^2 \|Z_\lambda^0 - Z_\lambda^\infty\|_F^2 \right\}. \qquad (61) \end{aligned}$$

Line 61 follows from Line 60, by using (57) and (59).

The r.h.s. expression in (61), is minimized at $\widehat{\theta}(k+1)$ given by

$$\begin{aligned} \widehat{\theta}(k+1) &= \min\{1, \theta_k\} \in [0,1], \text{ where,} \\ \theta_k &= \frac{f_\lambda(Z_\lambda^k) - f_\lambda(Z_\lambda^\infty)}{\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2}. \end{aligned}$$

If $\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2 = 0$, then we take $\theta_k = \infty$.

Note that θ_k is a decreasing sequence. This implies that if $\theta_{k_0} \leq 1$ then $\theta_m \leq 1$ for all $m \geq k_0$.

Suppose, $\theta_0 > 1$. Then $\widehat{\theta}(1) = 1$. Hence using (61) we have:

$$f_\lambda(Z_\lambda^1) - f_\lambda(Z_\lambda^\infty) \leq \frac{1}{2}\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2 \implies \theta_1 \leq \frac{1}{2}.$$

Thus we get back to the former case.

Hence $\theta_k \leq 1$ for all $k \geq 1$.

In addition, observe the previous deductions show that, if $\theta_0 > 1$ then (20) holds true for $k = 1$.

Combining the above observations, plugging in the value of $\widehat{\theta}$ in (61) and simplifying, we get:

$$f_\lambda(Z_\lambda^{k+1}) - f_\lambda(Z_\lambda^k) \leq -\frac{(f_\lambda(Z_\lambda^k) - f_\lambda(Z_\lambda^\infty))^2}{2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2}. \tag{62}$$

For the sake of notational convenience, we define the sequence $\alpha_k = f_\lambda(Z_\lambda^k) - f_\lambda(Z_\lambda^\infty)$. It is easily seen that α_k is a non-negative decreasing sequence.

Using this notation in (62) we get:

$$\begin{aligned} \alpha_k &\geq \frac{\alpha_k^2}{2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2} + \alpha_{k+1} \\ (\text{Since } \alpha_k \downarrow) &\geq \frac{\alpha_k \alpha_{k+1}}{2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2} + \alpha_{k+1}. \end{aligned} \tag{63}$$

Dividing both sides of the inequality in (63), by $\alpha_k \alpha_{k+1}$ we have:

$$\alpha_{k+1}^{-1} \geq \frac{1}{2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2} + \alpha_k^{-1}. \tag{64}$$

Summing both sides of (64) over $1 \leq k \leq (k-1)$ we get:

$$\alpha_k^{-1} \geq \frac{k-1}{2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2} + \alpha_1^{-1}. \tag{65}$$

Since $\theta_1 \leq 1$, we observe $\alpha_1/(2\|Z_\lambda^0 - Z_\lambda^\infty\|_F^2) \leq 1/2$—using this in (65) and rearranging terms we get, the desired inequality (20)—completing the proof of the Theorem. ■

References

J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert. A new approach to collaborative filtering: operator estimation with spectral regularization. *Journal of Machine Learning Research*, 10:803–826, 2009.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19*. MIT Press, 2007.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

F. Bach. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9: 1019–1048, 2008.

R. M. Bell and Y. Koren. Lessons from the Netflix prize challenge. Technical report, AT&T Bell Laboratories, 2007.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

J. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion, 2008. Available at `http://www.citebase.org/abstract?id=oai:arXiv.org:0810.3286`.

E. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9:717–772, 2008.

E. J. Candès and T. Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2009.

D. DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 249–256. ACM, 2006.

A. Dempster, N.. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society Series B*, 39:1–38, 1977.

D. Donoho, I. Johnstone, G. Kerkyachairan, and D. Picard. Wavelet shrinkage; asymptopia? (with discussion). *Journal of the Royal Statistical Society: Series B*, 57:201–337, 1995.

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

J. Friedman. Fast sparse regression and classification. Technical report, Department of Statistics, Stanford University, 2008.

J. Friedman, T. Hastie, H. Hoefling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 2(1):302–332, 2007.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, 2009. Web page and software available at `http://stanford.edu/~boyd/cvx`.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Prediction, Inference and Data Mining (Second Edition)*. Springer Verlag, New York, 2009.

S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *Proceedings of the 26th International Conference on Machine Learning*, pages 457–464, 2009.

R. H. Keshavan, S. Oh, and A. Montanari. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2009.

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

R.M. Larsen. Propack-software for large and sparse svd calculations, 2004. Available at `http://sun.stanford.edu/~rmunk/PROPACK`.

J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009. Available at `http://www.public.asu.edu/~jye02/Software/SLEP`.

Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identfication. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

S. Ma, D. Goldfarb, and L. Chen. Fixed point and Bregman iterative methods for matrix rank minimization. *Mathematical Programming Series A*, `forthcoming`.

R. Mazumder, J. Friedman, and T. Hastie. Sparsenet: coordinate descent with non-convex penalties. Technical report, Stanford University, 2009.

Y. Nesterov. *Introductory Lectures on Convex Optimization: Basic course*. Kluwer, Boston, 2003.

Y. Nesterov. Gradient methods for minimizing composite objective function. Technical Report 76, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, 2007. Available at http://www.citebase.org/abstract?id=oai:arXiv.org:0706.4138.

J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 713–719. ACM, 2005.

R. Salakhutdinov, A. Mnih, and G. E. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, pages 791–798. AAAI Press, 2007.

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. Available at http://www.cs.uic.edu/∼liub/KDD-cup-2007/proceedings.html.

N. Srebro and T. Jaakkola. Weighted low-rank approximations. In *Proceedings of the 20th International Conference on Machine Learning*, pages 720–727. AAAI Press, 2003.

N. Srebro, N. Alon, and T. Jaakkola. Generalization error bounds for collaborative prediction with low-rank matrices. In *Advances in Neural Information Processing Systems 17*, pages 5–27. MIT Press, 2005a.

N. Srebro, J. Rennie, and T. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336. MIT Press, 2005b.

G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk. Scalable collaborative filtering approaches for large recommender systems. *Journal of Machine Learning Research*, 10:623–656, 2009.

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

O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman. Missing value estimation methods for DNA microarrays. *Bioinformatics*, 17(6):520–525, 2001.

C. H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010.