Nuclear Norm Minimization via Active Subspace Selection

Cho-Jui Hsieh CJHSIEH@CS.UTEXAS.EDU
Department of Computer Science, The University of Texas, Austin, TX 78721, USA

Peder A. Olsen PEDERAO@US.IBM.COM
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract

We describe a novel approach to optimizing matrix problems involving nuclear norm regularization and apply it to the matrix completion problem. We combine methods from non-smooth and smooth optimization. At each step we use the proximal gradient to select an active subspace. We then find a smooth, convex relaxation of the smaller subspace problems and solve these using second order methods. We apply our methods to matrix completion problems including `Netflix` dataset, and show that they are more than 6 times faster than state-of-the-art nuclear norm solvers. Also, this is the first paper to scale nuclear norm solvers to the `Yahoo-Music` dataset, and the first time in the literature that the efficiency of nuclear norm solvers can be compared and even compete with non-convex solvers like Alternating Least Squares (ALS).

1. Introduction

We solve the nuclear norm optimization problem:

$$X = \underset{X \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} F(X) = \underset{X \in \mathbb{R}^{m \times n}}{\operatorname{argmin}} f(X) + \lambda \|X\|_*, \quad (1)$$

where $f(X)$ is a twice differentiable convex function, $\lambda > 0$ is the regularization parameter, and $\|X\|_* = \sum_{i=1}^{m} \sigma_i(X) = \operatorname{trace}(\sqrt{X^\top X})$ is the nuclear norm (also known as the trace norm). The nuclear norm regularization promotes a low rank solution, which is a key idea that can be applied to many applications such as recommender systems (Candès & Recht, 2009), dimension reduction in multivariate regression (Yuan et al., 2007), multi-task learning (Argryiou et al., 2008), and multi-label learning (Cabral et al., 2011). The nuclear norm regularization is an ℓ_1 regularization of the singular values of X, and it therefore promotes a low rank solution. It is proved that the underlying low rank solution can be recovered by solving (1) under certain conditions (Candés & Tao, 2009; Recht et al., 2010).

There has been much work on developing efficient nuclear norm minimization solvers (Ji & Ye, 2009; Mazumder et al., 2010; Jaggi & Sulovsky, 2010; Avron et al., 2012), but most of them still fail to solve large-scale problems. (Avron et al., 2012) reports that on the Netflix dataset, one of the state-of-the-art Stochastic Sub-Gradient Descent (SSGD) algorithms cannot achieve 0.95 test Root Mean Square Error (RMSE) in one day, while other non-convex methods (ALS) methods can achieve 0.93 test RMSE in a couple of hours. Similar scalability problems also arise in multi-task learning (Dudik et al., 2012) and multivariate regression (Giraud, 2011). This scalability deficiency makes nuclear norm regularization less applicable for large-scale real world problems, despite its strong theoretical guarantees.

In contrast, recently ℓ_1-regularized solvers have been well-developed and scaled to ultra-large-scale problems with a trillion parameters (Hsieh et al., 2013). The key technique used in the fastest ℓ_1-minimization solvers is to detect a small subset of active variables and focus on optimizing these (Olsen et al., 2012; Hsieh et al., 2011; Yuan et al., 2012). Since the nuclear norm is equivalent to the ℓ_1 norm on singular values, it is natural to ask the following question: can we identify a small *active subspace* and efficiently minimize the reduced-sized nuclear norm minimization problem?

In this paper, we propose two new methods to solve large-scale nuclear norm regularized problems using *active subspace selection*. Our methods alternate between two phases: firstly we identify the *active row and column subspaces*; secondly, within the subspace, the problem can be reduced to a smaller $k \times k$ nuclear norm minimization problem. We then describe two efficient solvers to solve the reduced problem, *alter-*

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

nating minimization method and *cone projection Newton descent method*, to minimize the sub-problem. We show that the active subspace will never change in a neighborhood of the global minimum, and in practice the subspace converges in very few iterations (10 in our experiments), thus our methods are extremely fast compared to other state-of-the-art methods.

Applications. The nuclear norm minimization can be applied to many applications where a low rank solution is preferred, and each application uses a different loss function $f(X)$ in (1). For example:

- **Matrix Completion**: Given a partially observed low rank $m \times n$ matrix A with observed entries in Ω, we can recover A by solving (1) with

$$f(X) = \frac{1}{2}\|\Pi_\Omega(X) - \Pi_\Omega(A)\|_F^2, \tag{2}$$

  where $(\Pi_\Omega(X))_{ij} = X_{ij}$ for all $(i,j) \in \Omega$ and 0 otherwise.
- **Multivariate Regression**: Given a data matrix $A \in \mathbb{R}^{l\times m}$ where each row of A is a data point, and a label matrix $B \in \mathbb{R}^{l\times n}$ where each row is n labels to a input data, we compute the model X by solving (1) with

$$f(X) = \frac{1}{2}\|AX - B\|_F^2. \tag{3}$$

  A low rank solution of X is preferred based on the discussion of (Yuan et al., 2007) for multivariate regression and (Amit et al., 2007) for multi-class learning.

In the experiments we will show the effectiveness of our algorithm on the two problems described above, and our method can also be extended to solve other nuclear norm regularized problems, including multi-task learning and clustering with missing labels.

2. Background Material

In this section we give some interesting background information related to nuclear norm minimization.

If X has the singular value decomposition $X = U\Sigma V^\top$, then we define the shrinkage operator by

$$S_\lambda(X) = U(\Sigma - \lambda I)_+ V^\top, \tag{4}$$

where the operation $a_+ = \max\{0, a\}$ is applied elementwise to the matrix. Also, we denote the pseudo-inverse of X by $X^\dagger$.

It is worthwhile noting that the regression problem (3) with a nuclear norm regularizer can be solved analytically when $AA^\top$ and $BB^\top$ commute.

Theorem 1 (Exact Solvability)**.** *Let $A \in \mathbb{R}^{m\times m}$, $B, X \in \mathbb{R}^{m\times n}$ then if $AA^\top$ and $BB^\top$ commutes the minimum of $F(X)$ is achieved for $X^* = (A^TA)^\dagger S_\lambda(A^\top B)$.*

The theorem can be proved by verifying that $F(X)$ is convex and that 0 is a subgradient at X^*. A similar result is shown in Theorem 4 of (Yu & Schuurmans, 2011). A particularly important case of Theorem 1 is $A = I$ when the solution simplifies to $X^* = S_\lambda(B)$, and this special case has been proved and used in many previous papers (Cai et al., 2010; Mazumder et al., 2010).

We now describe two first order methods that converge to the global minimum of the matrix completion and regression problems respectively.

Theorem 2. *The iteration $X_{k+1} = S_\lambda(\Pi_\Omega(A) + \Pi_\Omega^\perp(X_k))$, where $\Pi_\Omega^\perp(X_k) = X_k - \Pi_\Omega(X_k)$ converge to the global minimum of (2).*

See the Appendix 6.1 in the supplement for a sketch of the proof.

Theorem 3. *The iteration $X_{k+1} = S_{\lambda/c}(X_k - \frac{1}{c}\nabla f(X_k))$ converge to the global minimum of (3) if $cI \succ A^\top A = f''(X)$.*

See the Appendix 6.2 in the supplement for a sketch of the proof. It should be noted that the iteration in Theorem 2 is a special case of Theorem 3 with $c = 1$. The vector $S_{\lambda/c}(X - \frac{1}{c}f'(X))$ is the *proximal gradient* (Toh & Yun, 2010; Ji & Ye, 2009) and we shall make use of it later when we explore more efficient methods.

3. Our Proposed Method

Our proposed method iterates between two phases. At each iteration, we identify the *active row and column subspaces* U_A, V_A using the power method. Within the active subspace, the original problem can be reduced to a $k \times k$ nuclear norm minimization problem with $k \ll \min(m, n)$. We then propose efficient ways to minimize the sub-problem. Our framework can be summarized in Algorithm 1.

Algorithm 1: Our proposed framework

Input : regularization parameter λ, initial $X = U\Sigma V^\top$
Output: The optimal solution $X^* \in \mathbb{R}^{m\times n}$

for *iter* $= 1, 2, \ldots$ **do**
  $[\bar{U}, \bar{\Sigma}, \bar{V}] \leftarrow \texttt{ApproxSVD}(X - \nabla f(X))$;
  $U_G \leftarrow \{\bar{\boldsymbol{u}}_i \mid \bar{\Sigma}_{ii} > \lambda\}, V_G \leftarrow \{\bar{\boldsymbol{v}}_i \mid \bar{\Sigma}_{ii} > \lambda\}$;
  $U_A \leftarrow \texttt{QR}([U_G, U]), V_A \leftarrow \texttt{QR}([V_G, V])$;
  $S \leftarrow \operatorname{argmin}_S f(U_A S V_A^\top) + \lambda\|S\|_*$;
  $[U_S, \Sigma, V_S] \leftarrow \texttt{ThinSVD}(S)$;
  $U \leftarrow U_A U_S, \;\; V \leftarrow V_A V_S, \;\; X \leftarrow U\Sigma V^\top$;
end

3.1. Active subspace selection

In this section we introduce our *active subspace selection* strategy. Note that *any* matrix $X \in \mathbb{R}^{m\times n}$ can be represented as a sum of rank one matrices:

$$X = \sum_{ij} \sigma_{ij}\mathbf{u}_i\mathbf{v}_j^\top, \tag{5}$$

when $\text{span}\{\mathbf{u}_i\}_i = \mathbb{R}^m$ and $\text{span}\{\mathbf{v}_j\}_j = \mathbb{R}^n$. Since the solution is low rank, σ will be sparse. Therefore, the goal of the active set selection phase is to *eliminate rank-one subspaces* which are likely to have zero weights in the final solution, and then focus on the remaining rank one subspaces, which we call the *active subspace*.

If the SVD of X is $U\Sigma V$, it is shown in (Watson, 1992) that the sub-differential of $\|X\|_*$ is

$$\partial\|X\|_* = \{UV^\top + W : U^\top W = 0, WV = 0, \|W\|_2 \le 1\},$$

Assume $U^\perp, V^\perp$ are orthogonal subspace complements to U, V, then the sub-differential with respect to σ_{ij} can be written

$$\partial_{\sigma_{ij}} F(X) \in [\mathbf{u}_i^\top \nabla f(X)\mathbf{v}_j - \lambda, \mathbf{u}_i^\top \nabla f(X)\mathbf{v}_j + \lambda] \tag{6}$$

if $\mathbf{u}_i \in U^\perp$ or $\mathbf{v}_j \in V^\perp$ (equivalently, $\mathbf{u}_i^\top X\mathbf{v}_j = 0$). We can easily see that $|\mathbf{u}_i^\top \nabla f(X)\mathbf{v}_j| \le \lambda$ if and only if 0 is in the sub-differential. If we want to update X by a rank one factor $\sigma\mathbf{u}\mathbf{v}^\top$, the optimal step size is

$$\sigma^* = \operatorname*{argmin}_\sigma f(X + \sigma\mathbf{u}\mathbf{v}^\top) + \lambda\|X + \sigma\mathbf{u}\mathbf{v}^\top\|_*,$$

and it will have the solution $\sigma^* = 0$ whenever 0 is in the sub-differential set. We therefore define the *fixed rank one subspace* as

$$\mathcal{F} = \{\mathbf{u}\mathbf{v}^T \mid \mathbf{u}^\top X\mathbf{v} = 0 \text{ and } |\mathbf{u}^\top \nabla f(X)\mathbf{v}| \le \lambda\}.$$

Therefore, all the rank one subspaces in $\mathcal{F}$ have zero weight in the current solution X, and is not likely to change from zero to nonzero. However, fixed subspace elements can still be activated in the next iteration of our algorithm, when we compute a new proximal gradient. We define the *active rank one subspace* as

$$\mathcal{A} \equiv \{\mathbf{u}\mathbf{v}^\top \mid \mathbf{u}^\top X\mathbf{v} \ne 0 \text{ or } |\mathbf{u}^\top \nabla f(X)\mathbf{v}| > \lambda\},$$

which is the complementary set of $\mathcal{F}$. We focus on updating X on *active rank one subspaces* and fix the weights for all the fixed subspaces to be zero. In the following we propose a simple way to eliminate the fixed subspaces:

Theorem 4. *Assume $X = U\Sigma V^\top$ is the reduced SVD of X ($U \in \mathbb{R}^{m\times k}$, $V \in \mathbb{R}^{n\times k}$ and Σ has positive diagonal values), and $S_\lambda(X - \nabla f(X)) = U_G\Sigma_G V_G^\top$ (also a reduced SVD). Let U_A be an orthonormal basis of span(U, U_G), V_A be an orthonormal basis of span(V, V_G), and $U_A^\perp, V_A^\perp$ are orthonormal complements to U_A, V_A, then*

$$\{\mathbf{u}\mathbf{v}^\top \mid \mathbf{u} \in {U_A}^\perp \text{ or } \mathbf{v} \in {V_A}^\perp\} \subset \mathcal{F}.$$

The proof is in Appendix 6.3. Given U_A, V_A, we can form the column basis $\hat{U} = [U_A, U_A^\perp]$ and row basis $\hat{V} = [V_A, V_A^\perp]$, and then re-parameterize X by $X = \hat{U}Y\hat{V}^\top$ where $Y \in \mathbb{R}^{m\times n}$. Now solving the original problem is equivalent to solving $\min_Y F(UYV^\top)$. Assume both U_A and V_A have k columns, then by Theorem 4 only the top $k \times k$ submatrix of Y belong to the *active set* and all other elements of Y belong to the *fixed set*. So after eliminating the *fixed set*, the problem can be reduced to the following sub-problem:

$$\operatorname*{argmin}_{S\in\mathbb{R}^{k\times k}} F(U_ASV_A^\top) = f(U_ASV_A^\top) + \lambda\|U_ASV_A^\top\|_*, \tag{7}$$

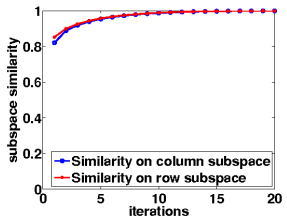
where $S = Y_{1:k,1:k} \in \mathbb{R}^{k\times k}$. Moreover, since U_A, V_A are orthogonal matrices $\|U_ASV_A^\top\|_* = \|S\|_*$, (7) is equivalent to

$$\operatorname*{argmin}_{S\in\mathbb{R}^{k\times k}} \hat{f}(S) + \lambda\|S\|_* \equiv \hat{F}(S), \tag{8}$$

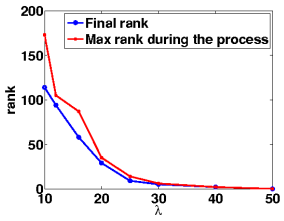
where $\hat{f}(S) = f(U_ASV_A^\top)$, $k \ll \min(n, m)$. Since the variable in (8) is a small $k \times k$ matrix, solving (8) is often much cheaper than solving the original problem. We will discuss how to solve (8) in Section 3.3. In addition, we will show in Theorem 7 that U_A, V_A are *exactly* equal to the row and column subspace of the optimal solution X^* in a neighborhood of X^*, and in this case solving (8) one time will achieve the global optimum.

Empirically we also observe that the active subsapce U_A, V_A converges to the final space quickly. In Figure 1a we compute the similarity between $(U_A)_t$ (U_A at the t-th iteration) and U^*, which is measured by the smallest singular value of $(U^*)^\top(U_A)_t$. If $U^* \subset \text{span}((U_A)_t)$, this value will be 1. We can see the value converges to 1 in ten steps. Moreover, the rank of our solutions $X_1, X_2, \ldots$ does not blow up when the final solution has a small rank, as shown in Figure 1b.

Relationship to other greedy methods. The family of "greedy algorithms", including GECO (Shalev-Shwartz et al., 2011), Lifted-CD (Dudik et al., 2012), and GCG (Zhang et al., 2012), have been proposed and achieved state-of-the-art performance on solving nuclear norm regularized problems. The greedy algorithms also consider the coordinate decomposition (5), but they construct the active subspace in a greedy manner: at each iteration, they add the top singular vector pair $(\mathbf{u}, \mathbf{v})$ to the active subspace, and then solve the problem within this subspace. The drawback of these greedy algorithms is the difficulty in removing unimportant basis elements. In contrast, our algorithm selects the rank-k subspace anew at each



(a) Subspace similarity between active subspace and the final solution



(b) Maximum rank during the process

Figure 1: Experiments on the ml100k dataset demonstrates the power of active subspace selection. Figure 1a shows that the active subspace converges very quickly, and Figure 1b demonstrates that the rank of X_t never greatly exceeds the final rank for a wide range of λ.

iteration, and thus rapidly removes irrelevant basis elements and converges faster as shown in the experiments.

3.2. Step 1: Computing active subspace

In this section, we discuss an efficient way to compute the active subspace U_A, V_A. Note again that U_A is the orthonormal basis of $\text{span}(U, U_G)$. U is the column basis of the current solution X, and this is always maintained during the process since we always maintain the low rank factor form $USV^\top$ of X.

To compute U_G and V_G, we have to compute $S_\lambda(X - \nabla f(X))$, which requires the top k singular vectors of $X - \nabla f(X)$ (assuming we know the rank k). This is the bottleneck in other proximal gradient based methods (Mazumder et al., 2010; Ji & Ye, 2009; Toh & Yun, 2010). As discussed in (Mazumder et al., 2010), since X is a low rank matrix and $\nabla f(X)$ usually have a special form, the matrix vector product $(X - \nabla f(X))\mathbf{v}$ can often be computed efficiently. For example, $\nabla f(X) = (X - A)_\Omega$ in matrix completion problems with square loss, and $\nabla f(X) = A^\top AX - A^\top B$ in multivariate regression problems. Therefore, (Mazumder et al., 2010) apply a Lanczos algorithm to compute the top k eigenvectors in Soft-Impute.

In our solver, we compute $S_\lambda(X - \nabla f(X))$ faster than Soft-Impute by use of the following innovations:

1. We observe that $S_\lambda(X - \nabla f(X))$ will not change a lot in two consecutive iterations. Therefore, we apply the power method (Halko et al., 2011) and use the eigenvectors in the previous iteration to initialize the power method. Using this *warm start* technique, the power method usually converges in 3 iterations.
2. In our solver, this step is only used to identify the subspace, and we will solve the sub-problem more accurately as discussed in Section 3.3. Therefore, we do not need to compute $S_\lambda(X - \nabla f(X))$ very accurately. We will show in Theorem 9 that our algorithm converges linearly even with only one step of the power method, and this is not true for other gradient descent algorithms.

Also, it is easy to increase the target rank k in the power method. Assume we already compute the top k_1 eigenvectors of $S_\lambda(X - \nabla f(X))$ and we find the k_1-th singular value is larger than λ, then we have to increase k_1. Suppose we increase k_1 to k_2; we can keep the top k_1 singular vectors, and then run the power method to compute the $k_1 + 1, \ldots, k_2$-th singular vectors. During the process, we just need to make sure those $k_1 + 1, \ldots, k_2$-th vectors are orthogonal to the top k_1 singular vectors. Therefore, the time complexity of each step of the power method is $O(|\Omega|k_2)$ to computing $A\mathbf{u}$ for each new vector $\mathbf{u} \in \mathbb{R}^{k_2}$ that is added, and $O((m+n)k_2^2)$ for orthogonalization. The algorithm is summarized in Algorithm 2.

Algorithm 2: Power method (ApproxSVD in Algorithm 1)

Input : Input matrix A, rank k, initial $R \in \mathbb{R}^{n \times k}$
Output: Approximate SVD $A \approx U\Sigma V^\top$

$Y \leftarrow AR$;
$Q \leftarrow \texttt{QR}(Y)$;
for $t = 1, \ldots, T^{max}$ **do**
  $Y \leftarrow A(A^\top Q)$;
  $Q \leftarrow \texttt{QR}(Y)$;
end
$B \leftarrow Q^\top A$;
$[\hat{U}, \Sigma, V] = \texttt{SVD}(B)$;
$U = Q\hat{U}$;

3.3. Step 2: Solving the $k \times k$ Sub-Problem

After selecting the subspace U_A, V_A, we need to solve (8). Since the variable S in (8) is a small $k \times k$ matrix, computing the SVD of S is cheap. Moreover, the gradient and Hessian vector product of $\hat{f}(S)$ can also be computed efficiently.

For general matrix-scalar functions, using the chain rule we have

$$\nabla \hat{f}(S) = \frac{\partial f(USV^\top)}{\partial S} = U^\top \left(\nabla f(Y)\,|_{Y=USV^\top}\right) V,$$

and if we define $\nabla^2 f(X)$ to be a $\mathbb{R}^{mn \times mn}$ matrix with elements $\frac{\partial^2 f}{\partial X_{ij} \partial X_{pq}}(X)$, then

$$\nabla^2 \hat{f}(S) = U^\top \otimes V^\top \left(\nabla^2 f(Y)\,|_{Y=USV^\top}\right) U \otimes V. \quad (9)$$

Usually by utilizing the structure of $\nabla f(Y)$ and $\nabla^2 f(Y)$, combined with $(U \otimes V)\,\text{vec}(D) = \text{vec}(VDU^\top)$, both gradient and Hessian can be computed efficiently. The following are some examples:

- Matrix Completion problems:

$$\nabla \hat{f}(S) = U^\top (X - A)_\Omega V,$$
$$\nabla^2 \hat{f}(S)\,\mathrm{vec}(D) = \mathrm{vec}(U^\top (UDV^\top)_\Omega V),$$

  so both the gradient and Hessian vector product can be computed in $O(k|\Omega| + nk^2)$ flops.
- Multivariate regression:

$$\nabla \hat{f}(S) = U^\top A^\top AUSV^\top V - U^\top A^\top BV,$$
$$\nabla^2 \hat{f}(S)\,\mathrm{vec}(D) = \mathrm{vec}(U^\top A^\top AUDV^\top V)$$

  so both the gradient and Hessian vector product can be computed in $O(kl(m+n))$ time (assume $k < l \ll m, n$).

In the reduced $k \times k$ problem, we can utilize the second order information, to achieve faster convergence. In the following we propose two novel algorithms for minimizing the sub-problem (8). For both algorithms, the most time consuming step is to compute the gradient or Hessian vector product, so the time complexity is proportional to that.

3.3.1. Alternating Minimization of an Auxiliary function

To compute the nuclear norm it is necessary to compute the square root $(S^\top S)^{1/2}$. If Z_0 commutes with $S^\top S$ (e.g. $Z_0 = S^\top S$) then the iteration $Z_{k+1} = \frac{1}{2}\left(Z_k + Z_k^{-1} S^\top S\right)$ coincides with Newton's algorithm and converges to the square root, (Higham, 1986). This motivated the following reformulation

Lemma 1 (A Convex Function)**.** *Let* $s(Z) = \frac{1}{2}\mathrm{trace}(Z + Z^{-1}S^\top S)$ *then* s *is a convex function (strictly convex when* S *is invertible) with*

$$\inf_{Z \succ 0} s(Z) = \|S\|_* \tag{10}$$

and the infimum is attained when S *is invertible for* $Z = (S^\top S)^{1/2}$*. When* S *is not invertible we can get arbitrarily close to the infimum by approaching* $(S^\top S)^{1/2}$ *from inside the cone of positive definite matrices.*

The Lemma follows directly from the expressions of the gradient and Hessian of s.

Based on Lemma 1, we can rewrite the sub-problem as

$$\min_S \inf_{Z \succ 0} f(USV^\top) + \frac{\lambda}{2}\mathrm{trace}(Z + (SS^\top)Z^{-1}) \equiv \min_S \inf_{Z \succ 0} g(S, Z). \tag{11}$$

By Lemma 1 it follows that this function is jointly convex in S, Z as stated in the following theorem

Theorem 5. *$g(S, Z)$ is jointly convex on S, Z on the domain $S \in \mathbb{R}^{k \times k}, Z \succeq 0$.*

Therefore, we can alternatingly minimize S and Z to solve (11). The update rules are described below.

Update S. When Z is fixed, $g(S, Z)$ in (11) is a convex quadratic function in S, so we can update S by Newtons method. When $f(X)$ is quadratic (as in matrix completion or multivariate regression), Newtons method converges in one iteration. Each Newton step can be computed by the Conjugate Gradient method (CG), which only requires us to compute Hessian vector products. The gradient is

$$\nabla_S g(S, Z) = \nabla \hat{f}(S) + \frac{\lambda}{2}(Z^{-\top} S + Z^{-1} S),$$

and since $\nabla_S^2 \mathrm{tr}(SSZ) = \frac{1}{2}(Z^{-\top} + Z^{-1}) \otimes I$, we have

$$\nabla_S^2 g(S, Z)\,\mathrm{vec}(D) = \nabla^2 \hat{f}(S)\,\mathrm{vec}(D) + \frac{\lambda}{2}\,\mathrm{vec}(DZ^{-\top} + DZ^{-1}).$$

We can further assume that Z is symmetric which gives

$$\nabla_S^2 g(S, Z)\,\mathrm{vec}(D) = \nabla^2 \hat{f}(S)\,\mathrm{vec}(D) + \lambda\,\mathrm{vec}(DZ^{-1}).$$

When the iteration cannot be computed efficiently, we can use limited memory BFGS (Nocedal & Wright, 1999) with line search instead.

Closed form solution for multivariate regression problem. Interestingly, when $f(X) = \|AX - B\|_F^2$, we can derive a closed form solution of $\min_S g(S, Z)$. By setting $\nabla_S g(S, Z) = 0$ we get

$$U^\top A^\top AUSV^\top V + \lambda Z^{-1} S = U^\top A^\top BV.$$

Assuming AU is full rank, then this equation is equivalent to

$$SP + QS = K, \tag{12}$$

where $P = V^\top V$, $Q = \lambda(U^\top A^\top AU)^{-1} Z^{-1}$ and $K = (AU)^\dagger BV$. Eq (12) is a Sylvester equation and can be solved in $\mathcal{O}(k^3)$ time by forming the SVD of P, Q and K or by using the even more efficient Bartels-Stewart algorithm, (Bartels & Stewart, 1972). Since all of them are k by k matrices and $k \ll m, n$, we can compute the exact solution efficiently.

Update Z. Lemma 1 shows that $Z = \sqrt{SS^\top}$ is a minimizer of $g(S, Z)$.

Since (11) is a convex problem and our method is a block coordinate descent method with two blocks, our method is guaranteed to converge (see Proposition 7.2.1 in (Bertsekas, 1999)). Moreover, Lemma 1 implies that $\hat{F}(S_t) = \min_Z g(S_t, Z) = g(S_t, Z_t)$, so $\hat{F}(S_1), \hat{F}(S_2), \ldots$ is a decreasing sequence and converges to the optimum of the sub-problem.

The details of this approach for solving the matrix completion problem is in Algorithm 3.

Algorithm 3: Our proposed algorithm Active ALT

Input : Active subspace U_A, V_A, initial S
Output: Solution S of (8)

$Z \leftarrow (SS^\top)^{1/2}$
for $t_{inner} = 0, 1, \ldots$ **do**
  $G \leftarrow U_A^\top (U_A S V_A^\top)_\Omega V + \frac{1}{2}\lambda(Z^{-1} + Z^{-\top})S$
  Solve $\nabla_S^2 g(S, Z)\,\mathrm{vec}(D) = \mathrm{vec}(G)$ by CG
  $S \leftarrow S - D$
  $Z \leftarrow (SS^\top)^{1/2}$
end

3.4. Cone Projection Newton Descent Method

Here we consider an alternative method to solve the $k \times k$ sub-problem that takes advantage of the fact that any norm is linear on at least a one dimensional cone since $\|tx\| = t\|x\|$ for $t > 0$. In both of the proposed focus problems the smooth part of the objective is a quadratic and therefore the entire problem becomes a smooth quadratic on the linearization cone. If the cone is only one dimensional this is not of great benefit, but for the ℓ_1 norm for example the cone is one of the mn dimensional orthants of $\mathbb{R}^{m\times n}$. The cone is also non-trivial for many other norms such as ℓ_∞ and the nuclear norm.

Theorem 6. *Let $U \in \mathbb{R}^{m\times k}$ and $V \in \mathbb{R}^{n\times k}$ be orthogonal matrices then the sub-differential of $\|X\|$ is constant on the cone $\{X = USV^\top : S = S^\top$ and $S \succ 0\}$.*

The proof is in Appendix 6.4. The positive definite cone is $\binom{k}{2}$ dimensional, which is nontrivial whenever $k > 1$. If we change the asymmetric portion of S then the derivative changes, much in the way it would for an ℓ_2 norm. The nuclear norm resembles both the ℓ_1 norm and the ℓ_2 norm. The ℓ_1 norm aspect we already saw, while the ℓ_2 norm is apparent when $m = 1$ since then $\|X\|_* = \|X\|_F$.

We propose to solve the quadratic problem by use of a quasi Newton method such as the conjugate gradient algorithm or limited memory BFGS (L-BFGS), (Nocedal & Wright, 1999). If the optimum over the symmetric matrices is not positive definite we do a backtracking line search. We additionally project the entire search line segment onto the cone to ensure convergence. This also encourages further reduction in rank. For a particular point on the line segment the projection consists simply of setting any negative eigenvalues to zero. This is analogous to what was done for the graphical LASSO problem in (Olsen et al., 2012).

4. Convergence Analysis

In our framework (Algorithm 1) there are two key tasks; 1) **step 2**: to compute U_A by `ApproxSVD` and 2) **step 5**: to solve the $k\times k$-size sub-problem. Both tasks incorporate an iterative solver. When both tasks are solved exactly, our algorithm converges to the global optimum (we omit the proof because this is a special case of Theorem 9).

As the sequence X_t converges to the global optimum X^*, we show that the active column and row subspace (U_A, V_A) will converge to the column and row space of X^* in finite number of iterations.

Theorem 7. *Assume step 2 and 5 are exact, and*

$$\lambda \text{ is not a singular value of } X^* - \nabla f(X^*) \qquad (13)$$

then span(U_A) = span(U^*), span(V_A) = span(V^*) after a finite number of iterations, where U^*, V^* are column and row space of the global optimum X^*.

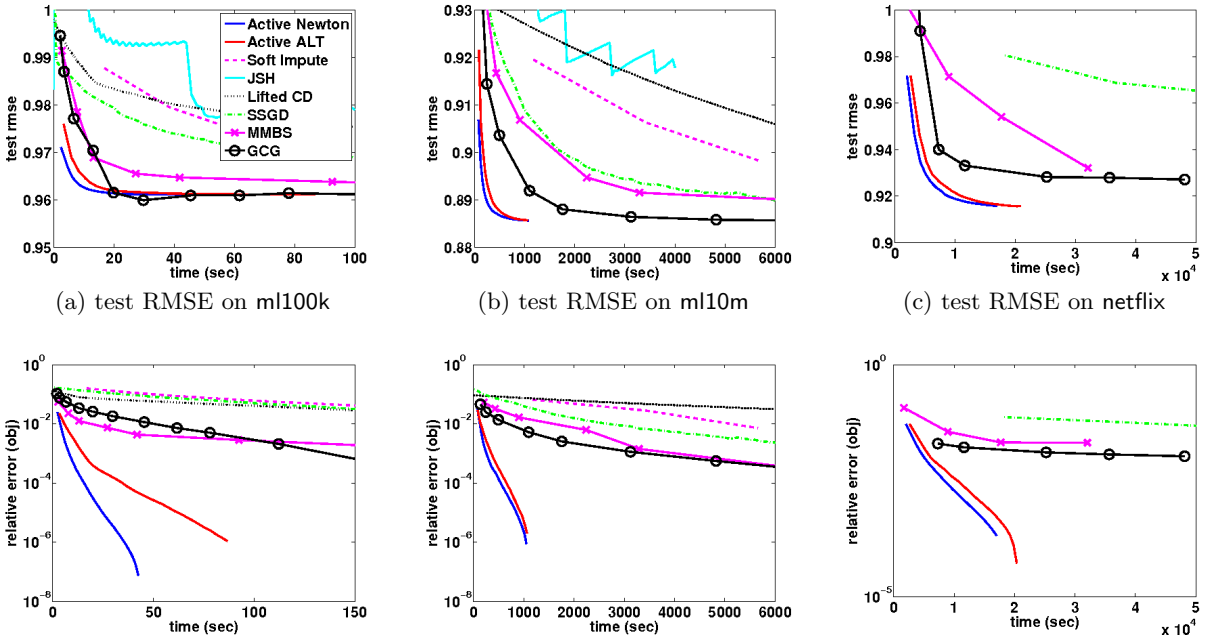
The proof is in Appendix 6.5. Note that Theorem 7 holds for any convergent sequence with $\lim_{t\to\infty} X_t = X^*$, and the assumption (13) can be shown to happen with very low probability, and was satisfied in our experiments. As long as U_A, V_A span the column/row space of X^*, Algorithm 1 can terminate in one step, so we have the following theorem:

Theorem 8. *If step 2 and 5 are exact and* (13) *holds, then Algorithm 1 terminates in a finite number of iterations.*

Since there is no closed-form solution to a general singular value decomposition, we consider the case where singular vectors are identified by the power method, as discussed in Algorithm 2. Assume in each iteration we use the previous U_G, V_G (denoted by $(U_G)_t, (V_G)_t$) as the initial subspace for the power method and run the power method for *more than one iteration*. In general power method cannot converge to the top k eigenvalues of A unless $V^\top R$ is nonsingular for the initial guess $R \in \mathbb{R}^{n\times k}$, where $V \in \mathbb{R}^{n\times k}$ is the top k singular vectors of A. This conditions is usually satisfied in practice. Under this condition, we prove the following theorem:

Theorem 9. *When step 2 is computed by power method with more than one iteration and step 5 is solved exactly, then Algorithm 1 converges to the optimum with an asymptotic linear convergence rate.*

The proof is in Appendix 6.6. This remarkable result shows that our algorithm converges fast even if the SVD in step 2 is computed inaccurately (i.e., with only one power iteration), if we initialize it by the previous U_G, V_G.



(a) test RMSE on ml100k (b) test RMSE on ml10m (c) test RMSE on netflix

(d) objective function on ml100k (e) objective function on ml10m (f) objective function on netflix

Figure 2: Comparison to other nuclear norm minimization solvers for the matrix completion problem. Methods with test RMSE or relative error above the top of y-axis are not shown in the figures. Our methods Active ALT and Active Newton are much faster than other methods.

Table 1: Dataset statistics and parameters.

dataset	m	n	$\lvert\Omega\rvert$	λ	k
ml100k	943	1,682	90,567	15	65
ml10m	69,878	10,677	9,301,260	100	44
netflix	2,649,429	17,770	99,072,112	300	53
yahoo	1,000,990	624,961	252,800,275	10000	54

5. Experimental Results

In this section, we compare our proposed nuclear norm solver with other state-of-the-art solvers. All the experiments are conducted on an Intel Xeon X5355 2.66GHz CPU with 32G RAM.

5.1. Matrix Completion: Comparison with nuclear norm solvers

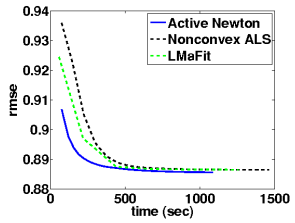
We compare the following methods:

- Soft-Impute: the gradient descent method proposed by (Mazumder et al., 2010).
- JSH: based on the work by (Jaggi & Sulovsky, 2010; Hazan, 2008), an extension of Frank-Wolfe method for optimizing a bounded SDP problem.
- SSGD: a Stochastic Sub-Gradient Descent method proposed by (Avron et al., 2012).
- Lifted CD: a greedy coordinate descent method proposed by (Dudik et al., 2012).
- MMBS: the method iteratively increases the rank and solves each fixed rank sub-problem by a trust-region Newton method (Mishra et al., 2013).
- GCG: A Generalized Conditional Gradient method proposed by (Zhang et al., 2012).
- Active ALT: our method with sub-problems solved by ALTernating minimization of S and Z.
- Active Newton: our method with sub-problems solved by the cone projection Newton method.

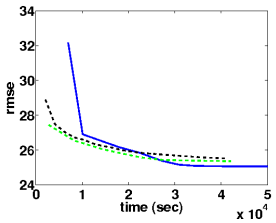
The implementation detail for the competing algorithms are described in Appendix 6.7. We use the real-world recommendation system datasets, MovieLens (ml100k, ml10m), Netflix, and Yahoo Music (Dror et al., 2012) as shown in Table 1. The balancing parameter λ was chosen by 3-fold cross validation on subsamples and the resulting rank k are also shown in Table 1. We compare the methods in terms of objective function value and test data RMSE. The results are shown in Figure 2. Notice the relative error on the y-axis is defined as $|(F(X) - F(X^*))/F(X^*)|$, where X^* is the optimal solution. JSH, SoftImpute, and Lifted CD are too slow on the Netflix dataset so we omit the results in Figure 2f and 2c. Since JSH solves a constrained version of (1), we omit the objective function value of JSH in the plots. The results show that our methods are more than 6 times faster than other solvers on large datasets.

Nuclear norm regularization is often considered too slow to solve large problems, and another approach is used to solve the non-convex problem:

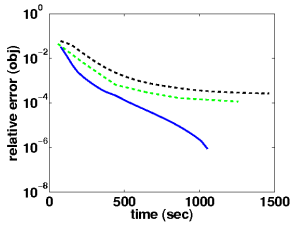
$$\min_{U\in\mathbb{R}^{m\times k},V\in\mathbb{R}^{n\times k}} f(UV^\top) + \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right). \quad (14)$$



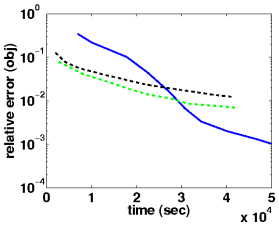
(a) test RMSE on ml10m



(b) test RMSE on yahoo



(c) objective function on ml10m



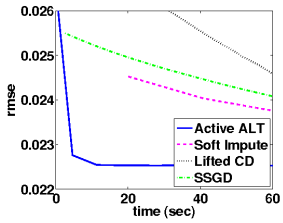
(d) objective function on yahoo

Figure 3: Comparison to non-convex methods (ALS and LMaFit). The speed of our method is competitive to non-convex methods, and they may not converge to the global optimum.

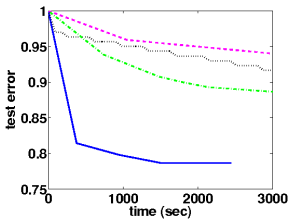
For some sufficiently large k, $\min_{U,V:X=UV^\top} \frac{1}{2}(\|U\|_F^2 + \|V\|_F^2) = \|X\|_*$, so solving (14) is equivalent to solving (1). However, (14) is not jointly convex in U, V, so solvers are *not* guaranteed to converge to the global optimum. We compare our method to non-convex solvers – Alternating Least Squares (ALS) and LMaFit (Wen et al., 2012) (a successive over-relaxation version of ALS) in Figure 3. Since Active ALS has very similar performance to Active Newton on large dataset, we only compare Active Newton with ALS in the figures. Non-convex solvers (especially ALS) were widely used in the Netflix price because of its scalability (Koren et al., 2009). We observe that our method is faster than ALS and LMaFit on ml10m, while on yahoo non-convex solvers are faster in the beginning, but converges to an inferior solution (because it may get stuck in saddle points). Therefore, our method is competitive in terms of time, is more stable, and has theoretical guarantees. This is the first paper to scale nuclear norm solvers to the yahoo dataset, and the first time in the literature that the efficiency of nuclear norm solvers can be compared with non-convex solvers.

5.2. Other Applications

Next, we apply our method to other problems with nuclear norm regularization. Since Active Newton and Active ALT has similar performance, we only show results for Active ALT in this section. We consider $m = 3724$ stocks, each with daily closing price recorded in 2012 ($l = 200$ days) downloaded from Yahoo Finance. Assume $\mathbf{p}_t \in \mathbb{R}^m$ is the stock return of all the m stocks at day t, and we model the change



(a) test RMSE on stock regression problem



(b) test error on ILSVRC multiclass problem

Figure 4: Comparison on regression and multi-class problems. Our proposed method is much faster than other nuclear norm solvers.

of return by the auto regression model $\mathbf{p}_t = X\mathbf{p}_{t-1}$. To estimate the transition matrix $X \in \mathbb{R}^{m\times m}$, we solve the multivariate regression problem (3) where $A = [\mathbf{p}_1\mathbf{p}_2 \dots \mathbf{p}_{l-1}]^\top$ and $B = [\mathbf{p}_2\mathbf{p}_3 \dots \mathbf{p}_l]^\top$. It was shown (Yuan et al., 2007) that a low rank assumption of X corresponds to the idea of feature sharing. We set $\lambda = 5$, which gives us a solution with rank 186. The model is tested on next 200 days data and evaluated using the root mean square error. The experimental results in Figure 4a show that our method is much faster than other methods.

We also test our method on a multi-class classification problem. We use the dataset from the ILSVRC-2010 competition. This dataset is a subset of ImageNet (Deng et al., 2009) with roughly 1000 images in each of the 1000 categories. There are 1.2 million training images and 150,000 testing images, the 1000 bag-of-visual-word features provided in the original dataset is used for classification. We model this as a multivariate regression problem, where each row of A is a training data, and each row of B is a unit vector $\mathbf{e}_{y_i}$ where y_i is the label of i-th training data. The nuclear norm regularization is useful and has theoretical benefit as shown in (Amit et al., 2007). We solve the nuclear norm regularized multivariate regression problem with $\lambda = 600$ to get a solution X^* with rank 189. The performance of our proposed algorithm and other methods are shown in Figure 4b. Our method achieves 18.57% test accuracy in 6 minutes, while no other nuclear norm solver can achieve this accuracy in 4 hours. Our method achieve the final accuracy 21.36% after 0.5 hours. Notice that this accuracy is already good since random guess for 1000 classes would just achieve a 0.1% accuracy.

Acknowledgements

We would like to thank Haim Avron and Vikas Sindhwani for providing the SSGD and JSH implementation as well as for valuable discussions. We also thank the anonymous reviewers for their suggestions. C.-J. Hsieh acknowledge support from an IBM PhD fellowship.

References

Amit, Y., Fink, M., Srebro, N., and Ullman, S. Uncovering shared structures in multiclass classification. In *ICML*, 2007.

Arbenz, Peter. *Lecture Notes on Solving Large Scale Eigenvalue Problems*. 2010.

Argryiou, A., Evgeniou, T., and Pontil, M. Convex multitask feature learning. *Machine Learning*, 73:243–272, 2008.

Avron, H., Kale, S., Kasiviswanathan, S., and Sindhwani, V. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *ICML*, 2012.

Bartels, RH and Stewart, GW. Algorithm 432: Solution of the matrix equation AX+ XB= C [F4]. *Communications of the ACM*, 15(9):820–826, 1972. ISSN 0001-0782.

Bertsekas, Dimitri P. *Nonlinear Programming*. Athena Scientific, Belmont, MA 02178-9998, second edition, 1999.

Cabral, R. S., Torre, F., Costeira, J. P., and Bernardino, A. Matrix completion for multi-label image classification. In *NIPS*, pp. 190–198, 2011.

Cai, J. F., Candes, E. J., and Zhen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Optimization*, 20(4), 2010.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Candés, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2009.

Daubechies, I., Defrise, M., and Mol, C. De. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on pure and applied mathematics*, 57(11):1413–1457, 2004.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. A large-scale hierarchical image database. In *CVPR*, 2009.

Dror, G., Koenigstein, N., Koren, Y., and Weimer, M. The Yahoo! music dataset and KDD-Cup'11. In *JMLR Workshop and Conference Proceedings: Proceedings of KDD Cup 2011 Competition*, volume 18, pp. 3–18, 2012.

Dudik, M., Harchaoui, Z., and Malick, J. Lifted coordinate descent for learning with trace-norm regularization. In *AISTATS*, 2012.

Giraud, C. Low rank multivariate regression. *Electronic Journal of Statistics*, 5:775–799, 2011.

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

Hazan, E. Sparse approximate solutions to semidefinite programs. *LATIN*, pp. 306–316, 2008.

Higham, N. J. Newtons method for the matrix square root. *Mathematics of Computation*, 46(174):537–549, 1986.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., and Ravikumar, P. Sparse inverse covariance matrix estimation using quadratic approximation. In *NIPS*, 2011.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S., Ravikumar, P., and Poldrack, R. A. Big & Quic: Sparse inverse covariance estimation for a million variables. In *NIPS*, 2013.

Jaggi, M. and Sulovsky, M. A simple algorithm for nuclear norm regularized problems. In *ICML*, 2010.

Ji, S. and Ye, J. An accelerated gradient method for trace norm minimization. In *ICML*, 2009.

Koren, Yehuda, Bell, Robert M., and Volinsky, Chris. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42:30–37, 2009.

Li, R.-C. Relative Perturbation Theory: II. Eigenspace and Singular subspace Variations. *SIAM Journal on Matrix Analysis and Applications*, 20(2):471–492, 1998.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *JMLR*, 11:2287–2322, 2010.

Mishra, B., Meyer, G., Bach, F., and Sepulchre, R. Low-rank optimization with trace norm penalty. *arxiv:1112.2318*, 2013.

Nocedal, J. and Wright, S. J. *Numerical optimization*, volume 2. Springer New York, 1999.

Olsen, P., Oztoprak, F., Nocedal, J., and Rennie, S. Newton-like methods for sparse inverse covariance estimation. In *NIPS*, 2012.

Recht, B., Fazel, M., and Parrilo, P. A. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

Shalev-Shwartz, S., Gonen, A., and Shamir, O. Large-scale convex minimization with a low-rank constraint. In *ICML*, 2011.

Toh, K.-C. and Yun, S. An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *J. Optimization*, 6:615–640, 2010.

Watson, G. A. Characterization of the subdifferential of some matrix norms. *Linear Algebra and its Applications*, 170:33–45, 1992.

Wen, Z., Yin, W., and Zhang, Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathemetical Programming Computation*, 4(4):333–361, 2012.

Yu, Y. and Schuurmans, D. Rank/norm regularization with closed-form solutions: application to subspace clustering. In *UAI*, 2011.

Yuan, G.-X., Ho, C.-H., and Lin, C.-J. An improved GLMNET for L1-regularized logistic regression. *JMLR*, 13: 1999–2030, 2012.

Yuan, M., Ekici, A., Lu, Z., and Monteiro, R. Dimension reduction and coefficient estimation in multivariate linear regression. *J. R. Statist. Soc*, pp. 329–346, 2007.

Zhang, X., Yu, Y., and Schuurmans, D. Accelerated training for matrix-norm regularization: A boosting approach. In *NIPS*, 2012.

6. Appendix

6.1. Proof sketch for Theorem 2

This theorem is proved in (Mazumder et al., 2010) by considering the auxilliary function

$$\begin{aligned} Q(X,Y) &= \frac{1}{2}\|\Pi_\Omega(A)+\Pi_\Omega^\perp(Y)-X\|_F^2+\lambda\|X\|_* \\ &= F(X)+\frac{1}{2}\|\Pi_\Omega^\perp(Y-X)\|_F^2, \end{aligned}$$

for which $Q(X,Y)\geq F(X)$ and $Q(X,X)=F(X)$. We can minimize the auxiliary function by noting that the minimum with respect to Y for fixed X is $Y=X$ and for fixed Y the minimum with respect to X is $X=S_\lambda(\Pi_\Omega(A)+\Pi_\Omega^\perp(Y))$. Alternating the minimization gives the iteration in the theorem. This algorithm is known as Soft-Impute.

6.2. Proof sketch for Theorem 3

For the regression problem we can form a different auxilliary function. If $cI\succ A^\top A=f''(X)$ then $-\frac{1}{2}\|AX-AY\|_F^2+\frac{c}{2}\|X-Y\|_F^2\geq 0$ for all X,Y and the auxilliary function

$$\begin{aligned} Q(X,Y) &= \frac{1}{2}\|AX-B\|_F^2-\frac{1}{2}\|A(X-Y)\|_F^2 \\ &\quad +\frac{c}{2}\|X-Y\|_F^2+\lambda\|X\|_* \\ &= \frac{c}{2}\|X-Y-\frac{1}{c}(A^\top B-A^\top AY)\|_F^2 \\ &\quad +\lambda\|X\|_*+\text{const} \\ &= c(\frac{1}{2}\|X^\top-(Y-\frac{1}{c}f'(Y))\|_F^2+\frac{\lambda}{c}\|X\|_*) \\ &\quad +\text{const} \end{aligned}$$

satisfy $Q(X,Y)\geq F(X)$ and $Q(X,X)=F(X)$. For fixed Y_k the minimum over X is $X_{k+1}=S_{\lambda/c}(Y_k-\frac{1}{c}f'(Y_k))$ and for fixed X_k the minimum over Y is $Y_k=X_k$. This auxilliary function is constructed completely analogously to the ℓ_1 case, for which global convergence is formally proved in (Daubechies et al., 2004).

6.3. Proof of Theorem 4

Proof. If $\mathbf{u}\in U_A^\perp$, then (1) $U^\top\mathbf{u}=0$, which implies $\mathbf{u}^\top X=0$; (2) $U_G^\top\mathbf{u}=0$, which implies $|\mathbf{u}^\top(X-\nabla f(X))\mathbf{v}|<\lambda$ for any $\mathbf{v}$ (by the definition of soft-thresholding operator $\mathcal{S}$). Combining (1) and (2) we have $\mathbf{u}\mathbf{v}^\top\in\mathcal{F}$ for all $\mathbf{v}$ if $\mathbf{u}\in U_A^\perp$. By the same argument we can prove $\mathbf{u}\mathbf{v}^\top\in\mathcal{F}$ for all $\mathbf{u}$ if $\mathbf{v}\in V_A^\perp$. □

6.4. Proof of Theorem 6

Proof. Since S is positive definite it has an eigenvalue decomposition $S=P\Sigma P^\top$ with $\Sigma\succ 0$ a diagonal matrix. Therefore the SVD of X can be written $X=(UP)\Sigma(VP)^\top$ and the sub-differential is

$$\partial\|X\|_*=\{UV^\top+W:U^\top W=0,WV=0,\|W\|_2\leq 1\},$$

independent of S since $(UP)(VP)^\top=UV^\top$. □

6.5. Proof of Theorem 7

Proof. Assume $X^*=U^*\Sigma^*V^*$ is the reduced SVD of X^*. Since X^* is the global optimum,

$$\begin{aligned} X^* &= S_\lambda(X^*-\nabla f(X^*)) \\ &= \bar{U}^*(\bar{\Sigma}^*-\lambda I)_+(\bar{V}^*)^\top. \end{aligned} \tag{15}$$

If there are k singular values in $\bar{\Sigma}^*$ larger than λ, then it is clear that the first k columns of $\bar{U}^*$ is U^*, and the first k columns of $\bar{V}^*$ is V^*. By our assumption, $\Sigma_{ii}\neq\lambda$ for all i, so we can assume $\Sigma_{kk}>\lambda$ and $\Sigma_{k+1,k+1}<\lambda-\epsilon$ with some $\epsilon>0$.

We consider the set

$$\mathcal{Z}\equiv\{(\mathbf{u},\mathbf{v})\mid\mathbf{u}\in(U^*)^\perp\text{ or }\mathbf{v}\in(V^*)^\perp\}.$$

For $(\mathbf{u},\mathbf{v})\in\mathcal{Z}$, $\mathbf{u}^\top X^*\mathbf{v}=0$, so

$$|\mathbf{u}^\top(X^*-\nabla f(X^*))\mathbf{v}|=|\mathbf{u}^\top\nabla f(X^*)\mathbf{v}|<\lambda-\epsilon.$$

Since the sequence X_t generated by Algorithm 1 converges to the global optimum X^*, there exists a T such that

$$\|\nabla f(X_t)-\nabla f(X^*)\|<\epsilon \tag{16}$$

and

$$|\mathbf{u}^\top\nabla f(X_t)\mathbf{v}|<\lambda \tag{17}$$

for all $t>T$ and any $(u,v)\in\mathcal{Z}$. Now for any $(u,v)\in\mathcal{Z}$ we consider two cases:

1. If $\mathbf{u}^\top X_{t-1}\mathbf{v}\neq 0$, then $\mathbf{u}\in(U_A)_{t-1}$ and $\mathbf{v}\in(V_A)_{t-1}$. Since we exactly solve the sub-problem (7) and we already know $|\mathbf{u}^\top\nabla f(X_t)\mathbf{v}|<\lambda$, the optimality condition of (7) implies $\mathbf{u}^\top X_t\mathbf{v}=0$.
2. If $\mathbf{u}^\top X_{t-1}\mathbf{v}=0$, then combined with (17) we know $\mathbf{u},\mathbf{v}$ are not in the active subspace, so $\mathbf{u}^\top X_t\mathbf{v}=0$.

Therefore, once $t>T$, for any $\mathbf{u}\in(U^*)^\perp$ or $\mathbf{v}\in(V^*)^\perp$, $\mathbf{u}^\top X_t\mathbf{v}$ will be zero and will never be selected in $(U_A)_t,(V_A)_t$. This implies that $\text{span}((U_A)_t)\subseteq\text{span}(U^*)$ and $\text{span}((V_A)_t)\subseteq\text{span}(V^*)$.

Next we prove the equality part. For all $\mathbf{u},\mathbf{v}$ such that $\mathbf{u}^\top X^*\mathbf{v}\neq 0$, there exists a T such that $\mathbf{u}^\top(X_t)\mathbf{v}\neq 0$ for all $t>T$ (since the smallest eigenvalue >0). Therefore, all such $\mathbf{u},\mathbf{v}$ will belong to $(U_A)_t,(V_A)_t$ after $t>T$. Combined with the previous argument, we have $\text{span}((U_A)_t)=\text{span}(U^*)$ and $\text{span}((V_A)_t)=\text{span}(V^*)$ after $t>T$. □

6.6. Proof of Theorem 9

Proof. We first introduce an important property of the power method, which will be useful for proving the theorem.

The power method (subspace iteration) described in Algorithm 2 has a linear convergence rate: assume U, V are the top-k singular vectors of A, σ_k, σ_{k+1} are the k-th and $(k+1)$-st singular values, and the approximate SVD given by Algorithm 2 with R as initial and with T^{max} steps. If the initial matrix R satisfies the condition that $V^\top R$ is nonsingular, then

$$\|\hat{U}\hat{U}^\top - UU^\top\| \le (\frac{\sigma_{k+1}}{\sigma_k})^{T^{max}} \|U_R U_R^\top - UU^\top\|,$$
$$\|\hat{V}\hat{V}^\top - VV^\top\| \le (\frac{\sigma_{k+1}}{\sigma_k})^{T^{max}} \|V_R V_R^\top - VV^\top\|. \quad (18)$$

where U_R is the orthogonal subspace of R, V_R is the orthogonal subspace of AR, and $\hat{U}$ is the subspace after one power iteration. This property is shown in Theorem 7.2 in (Arbenz, 2010).

Now we prove that the sequence X_t generated by Algorithm 1 converges to the global optimum. For convenience, we define $P(X) := \mathcal{S}_\lambda(X - \nabla f(X))$, and $\tilde{P}(X)$ to be the *computed* value (by the power method with one iteration) of $P(X)$. The reduced SVD of $\mathcal{S}_\lambda(X - \nabla f(X))$ is denoted by $U_G(X)\Sigma_G(X)(V_G(X))^\top$, and the computed subspace vectors is $\tilde{U}_G(X), \tilde{V}_G(X)$. We use $\tilde{U}_G(X_t)$ to denote the computed value at the t-th iteration, and $U_G(X_t)$ to denote the true subspace vectors at the t-th iteration.

Since Algorithm 1 ensures that the objective function value decreases at each iteration, the sequence $\{X_t\}$ is in a compact set. Therefore, there exists a subsequence of X_{s_t} converges to a limit point $\bar{X}$. For convenience we denote s_t by t in the following. We want to prove $\bar{X}$ is the global optimum by contradiction, so we first assume $\bar{X} \neq X^*$, so $P(\bar{X}) \neq \bar{X}$.

First we want to show $\tilde{U}(X_t), \tilde{V}(X_t)$ converges to $U_G(\bar{X}), V_G(\bar{X})$ (the computed subspace converges to the true subspace). Assume $\tilde{U}_G(X_t), \tilde{V}_G(X_t)$ converges to $\tilde{U}, \tilde{V}$, then what we want to show is that $\text{span}(\tilde{U}) = \text{span}(U_G(\bar{X}))$ and $\text{span}(\tilde{V}) = \text{span}(V_G(\bar{X}))$. Since $\{X_t\}$ converges to $\bar{X}$ and $X - \nabla f(X)$ is a continuous function, for any $\epsilon > 0$ there exists a T_1 such that $\forall t > T_1$,

$$\|(X_t - \nabla f(X_t)) - (\bar{X} - \nabla f(\bar{X}))\| \le \epsilon. \quad (19)$$

By perturbation theory (Li, 1998), for any matrix A and a small perturbation Δ, we have

$$\max(\|U(A)U(A)^\top - U(A+\Delta)U(A+\Delta)^\top\|,$$
$$\|V(A)V(A)^\top - V(A+\Delta)V(A+\Delta)^\top\|) \le \|\Delta\|/\delta,$$

where δ is the singular-gap between $\sigma_k(A)$ and $\sigma_{k+1}(A)$, and $U(A), V(A)$ are the top-k singular vectors of A. Now we consider $A = P(\bar{X}), \Delta = P(X_t) - P(\bar{X})$, then we have

$$\max(\|U_G(X_t)U_G(X_t)^\top - U_G(\bar{X})U_G(\bar{X})^\top\|,$$
$$\|V_G(X_t)V_G(X_t)^\top - V_G(\bar{X})V_G(\bar{X})\|) \le \|P(X_t) - P(\bar{X})\|/\delta,$$

Combining with (19) we get

$$\|U_G(X_t)U_G(X_t)^\top - U_G(\bar{X})U_G(\bar{X})^\top\| \le \frac{\epsilon}{\delta} \quad \forall t > T_1. \quad (20)$$

Now assume t is large enough so that

$$\|\tilde{U}\tilde{U}^\top - \tilde{U}_G(X_{t-1})\tilde{U}_G(X_{t-1}^\top)\| < \epsilon_1, \quad (21)$$

so we have

$$\begin{aligned}
&\|\tilde{U}_G(X_t)\tilde{U}_G(X_t)^\top - U_G(\bar{X})U_G(\bar{X})^\top\| \\
&\le \|\tilde{U}_G(X_t)\tilde{U}_G(X_t)^\top - U_G(X_t)U_G(X_t)^\top\| + \frac{\epsilon}{\delta} \text{ (by (20))} \\
&\le (\frac{\sigma_{k+1}}{\sigma_k})\|\tilde{U}_G(X_{t-1})\tilde{U}_G(X_{t-1})^\top - U_G(X_t)U_G(X_t)^\top\| + \frac{\epsilon}{\delta} \text{ (by (18))} \\
&\le (\frac{\sigma_{k+1}}{\sigma_k})\|\tilde{U}\tilde{U}^\top - U_G(X_t)U_G(X_t)^\top\| + \frac{\epsilon}{\delta} + \epsilon_1. \text{ (by (21))} \\
&\le (\frac{\sigma_{k+1}}{\sigma_k})\|\tilde{U}\tilde{U}^\top - U_G(\bar{X})U_G(\bar{X})^\top\| + 2\frac{\epsilon}{\delta} + \epsilon_1 \text{ (by (20))}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\|\tilde{U}_G(X_t)\tilde{U}_G(X_t)^\top - \tilde{U}\tilde{U}^\top\| \\
&\ge \|U_G(\bar{X})U_G(\bar{X})^\top - \tilde{U}\tilde{U}^\top\| \\
&- \|\tilde{U}_G(X_t)\tilde{U}_G(X_t) - U_G(\bar{X})U_G(\bar{X})^\top\| \\
&\ge (1 - \frac{\sigma_{k+1}}{\sigma_k})\|\tilde{U}\tilde{U}^\top - U_G(\bar{X})U_G(\bar{X})^\top\| - 2\frac{\epsilon}{\delta} - \epsilon_1.
\end{aligned}$$

Taking $t \to \infty$ on both side and $\epsilon, \epsilon_1 \to 0$ we have

$$0 \ge (1 - \sigma_{k+1}/\sigma_k)\|\bar{U}\bar{U}^\top - U_G(\bar{X})U_G(\bar{X})^\top\|.$$

So $\text{span}(U_G(\bar{X})) = \text{span}(\tilde{U})$. Using the same derivations on the right singular vectors V, we can get $\text{span}(V_G(\bar{X})) = \text{span}(\tilde{V})$.

The above argument shows that $\tilde{P}(X_t) \to P(\bar{X})$. If $\bar{X}$ is not a global optimum, then $P(\bar{X}) \neq \bar{X}$. Since all the fixed points are global optimum, by a typical convergence property for the fixed-point operation we can show that $\bar{X}$ is a global optimum.

Next, we prove the asymptotic convergence rate. By Theorem 7, we know after finite steps T_1, $U_A(X_t) = U^*, V_A(X_t) = V^*$. Moreover, $\sigma_k(X_t - \nabla f(X_t))$ converges to $\sigma_k(X^* - \nabla f(X^*))$ and $\sigma_{k+1}(X_t - \nabla f(X_t))$ converges to $\sigma_{k+1}(X^* - \nabla f(X^*))$, so there exists a T_2 such that for all $t > T_2$,

$$\frac{\sigma_{k+1}(X_t - \nabla f(X_t))}{\sigma_k(X_t - \nabla f(X_t))} \le \frac{\lambda - \epsilon/2}{\lambda}. \quad (22)$$

Assume $\bar{T} = \max(T_1, T_2)$. Since for each iteration we run one power iteration on $X_t - \nabla f(X_t)$ and the gap of the k-th and $(k+1)$-st singular values are guaranteed in (22), from (18) we can bound the error between subspaces $(U_A)_t$ and U^* by

$$\begin{aligned}&\|U_A(X_t)U_A(X_t)^\top - U^*(U^*)^\top\| \\ &\leq (1 - \frac{\epsilon}{2\lambda})\|U_A(X_{t-1})U_A(X_{t-1})^\top - U^*(U^*)^\top\|\end{aligned}$$

when $t > \bar{T}$. Therefore

$$\begin{aligned}&\|U_A(X_{\bar{T}+t})U_A(X_{\bar{T}+t})^\top - U^*(U^*)^\top\| \\ &\leq (1 - \frac{\epsilon}{2\lambda})^t\|U_A(X_{\bar{T}})U_A(X_{\bar{T}})^\top - U^*(U^*)^\top\|. \quad (23)\end{aligned}$$

At the t-th iteration, it is clear that $\bar{S}_t = U_A(X_t)^\top X^* V_A(X_t)$ is a feasible solution for the subproblem (8). Let $\bar{X}_t = U_A(X_t)\bar{S}_t V_A(X_t)^\top$. Since $\bar{S}_t$ is the minimizer of (8), $\bar{X}_t$ is the minimizer within the U_A, V_A subspace. By definition, the subspace of X_t is a subset of U_A, V_A, therefore $F(X_t) \geq F(\bar{X}_t)$.

Also, $X^* = U^*(U^*)^\top X^* = X^* V^*(V^*)^\top$, so

$$\begin{aligned}&\|\bar{X}_t - X^*\| \\ &\leq \|U_A(X_t)U_A(X_t)^\top X^* V_A(X_t)V_A(X_t)^\top - U_A(X_t)U_A(X_t)^\top X^*\| \\ &\quad + \|U_A(X_t)U_A(X_t)^\top X^* - X^*\| \\ &= \|U_A(X_t)U_A(X_t)^\top X^* \left(V^*(V^*)^\top - V_A(X_t)V_A(X_t)^\top\right)\| \\ &\quad + \|(U^*(U^*)^\top - U_A(X_t)U_A(X_t)^\top)X^*\| \\ &\leq (\|U^*(U^*)^\top - U_A(X_t)U_A(X_t)^\top\| + \\ &\quad \|V^*(V^*)^\top - V_A(X_t)V_A(X_t)^\top\|)\|X^*\|.\end{aligned}$$

Next we relate this quantity with the objective function value $F(X_t)$. From Lemma 3.1 in (Ji & Ye, 2009),

$$F(X) - F(X^*) \leq L\|X - X^*\|_F^2,$$

where L is the Lipschitz constant for $\nabla f(X)$. Substituting $\bar{X}_t$ into the above inequality we get

$$\begin{aligned}F(X_t) - F(X^*) &\leq F(\bar{X}_t) - F(X^*) \\ &\leq LR(\|U^*(U^*)^\top - U_A(X_t)U_A(X_t)^\top\| \\ &\quad + \|V^*(V^*)^\top - V_A(X_t)V_A(X_t)^\top\|),\end{aligned}$$

where $R = \|X^*\|$ is a constant. Applying (23) we can get

$$F(X_t) - F(X^*) \leq LR(1 - \frac{\epsilon}{2\lambda})^{t-\bar{T}}$$

when $t > \bar{T}$. Therefore our algorithm has an asymptotically linear convergence rate.

□

6.7. Implementation Details for the comparison

We discuss the implementation detail for other algorithms in our comparison. The code for Soft-Impute is downloaded from `http://statweb.stanford.edu/~rahulm/SoftImpute/`. In their code, the top-k singular vectors is computed by Lanczos algorithm. We use the same JSH and SSGD implementation as in (Avron et al., 2012), where the largest singular value is computed by the `SVDS` function in MATLAB and the parameters are tuned by the authors. More specifically, $\delta = 0.04$ for ml100k, $\delta = 0.015$ for ml10m and netflix, and $\nu = 0.005$ for all datasets. We implement LiftedCD by ourselves and compute the largest singular value by the power method. For MMBS the code is downloaded from `http://www.montefiore.ulg.ac.be/~mishra/softwares/traceNorm.html`, and the GCG code is downloaded from `http://users.cecs.anu.edu.au/~xzhang/GCG/`.