Nuclear Norm Regularized Least Squares Optimization on Grassmannian Manifolds

Yuanyuan Liu
Hong Cheng
Dept. of Systems Engineering and Engineering
Management, The Chinese University of Hong Kong
{yyliu, hcheng}@se.cuhk.edu.hk

Fanhua Shang[*]
James Cheng
Dept. of Computer Science and Engineering
The Chinese University of Hong Kong
{fhshang, jcheng}@cse.cuhk.edu.hk

Abstract

This paper aims to address a class of nuclear norm regularized least square (NNLS) problems. By exploiting the underlying low-rank matrix manifold structure, the problem with nuclear norm regularization is cast to a Riemannian optimization problem over matrix manifolds. Compared with existing NNLS algorithms involving singular value decomposition (SVD) of large-scale matrices, our method achieves significant reduction in computational complexity. Moreover, the uniqueness of matrix factorization can be guaranteed by our Grassmannian manifold method. In our solution, we first introduce the bilateral factorization into the original NNLS problem and convert it into a Grassmannian optimization problem by using a linearized technique. Then the conjugate gradient procedure on the Grassmannian manifold is developed for our method with a guarantee of local convergence. Finally, our method can be extended to address the graph regularized problem. Experimental results verified both the efficiency and effectiveness of our method.

1 Introduction

In recent years, matrix approximation problems with nuclear norm regularization have occurred in many machine learning and compressed sensing applications such as matrix completion, matrix classification, multi-task learning and dimensionality reduction [6]. In this paper, we consider the following optimization problem over matrices:

$$\min_{X\in\mathbb{R}^{m\times n}} f(X) := g(X) + \mu\|X\|_*, \tag{1}$$

where $g(X)$ is any differentiable convex function (usually called the loss function, e.g. $g(X) = \|\mathcal{A}(X) - b\|_2^2$, where $\mathcal{A}(\cdot)$ is a linear operator), $\mu > 0$ is a regularization parameter, and $\|X\|_*$ denotes the nuclear (or trace) norm of the matrix X with rank r, that is, the l_1-norm of the matrix spectrum as $\|X\|_* = \sum_{i=1}^{r}\sigma_i$, where $\{\sigma_i\}$ are the singular values of X.

Most algorithms for solving the nuclear norm minimization (NNM) problem (1) do not require the rank to be specified and iteratively optimize the nuclear norm penalized problem. Naturally, the singular value decomposition (SVD) tends to paly a critical computational role in the design of various nuclear norm solvers, e.g., the singular value thresholding (SVT) [5], soft-impute [14], accelerated proximal gradient approach [9], and so on. Those algorithms involving SVD and applying a soft-thresholding operator on the singular values at each iteration suffer from high computational cost of multiple SVDs [14, 15]. In particular, if the iterations need to pass through a region where the spectrum is dense, those algorithms can become potentially become prohibitively expensive [3]. Noticing that only those singular values exceeding a threshold and their corresponding singular vectors contribute to the soft-thresholding operator, a commonly used strategy is to compute the partial SVD instead of the full one, such as APGL [17] and IALM [12] both use PROPACK [11]. However, it can compute only a given number of largest singular values, and the soft-thresholding operator requires the principal singular values that are greater than a given threshold.

If the rank is known, a class of existing matrix factorization algorithms [10, 4, 22, 15, 18] cast the low-rank matrix estimation problem (1) as the following non-convex model,

$$\min_{X\in\mathbb{R}^{m\times n}} g(X), \quad \text{s.t., rank}(X) = k. \tag{2}$$

Matrix factorization is arguably the most widely applied method for the low-rank matrix completion problem, due to its high accuracy, scalability and flexibility to incorporating side-information [19]. LMaFit [22] fixes the rank by explicitly formulating the matrix as the product of its low-rank factors and using an optimization technique based on successive over-relaxation to solve (2). In [15] and [18], two improved versions were proposed to optimize it on the

[*]Corresponding author

Grassmannian manifolds, and improve its convergence by using conjugate gradients rather than the standard gradient descent. Moreover, [10] proved that exact recovery can be obtained with high probability by solving a non-convex optimization problem. In the model (2), the correct rank needs to be known as a priori. Unfortunately, the determination of the reduced rank is also an open problem, especially for the noisy matrix estimation.

To address these key problems mentioned above, we propose an effective approximation method for solving nuclear norm regularized least squares problems, which can reduce the SVD computational cost. We achieve it by converting the original NNM problem into a Grassmannian optimization problem. In our framework, we use the nuclear norm term to promote the robustness of the fixed-rank manifold optimization problem with respect to the given rank, in other words, to avoid the over-fitting problems of matrix factorization. Moreover, we present an efficient conjugate gradient descent algorithm on the Grassmannian manifolds with a guarantee of local convergence. Finally, our method is also extended to address the graph regularized problem. In summary, our method inherits the superiority of two classes of frameworks, i.e., the NNM methods and Riemannian manifold optimization methods based on matrix factorizations.

2 Background

When choosing $g(X) := \frac{1}{2}\|\mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(Z)\|_F^2$ for some linear projection operators $\mathcal{P}_\Omega(\cdot)$, i.e., $\mathcal{P}_\Omega(X_{ij}) = X_{ij}$ if $(i,j) \in \Omega$, and $\mathcal{P}_\Omega(X_{ij}) = 0$ otherwise, the above formulation (1) is the low-rank matrix completion (MC) problem. The MC problem is to find out a matrix of the lowest rank whose entries in the observed set Ω correspond to the entries of Z:

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{2}\|\mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(Z)\|_F^2 + \mu\|X\|_*, \qquad (3)$$

or the noiseless version,

$$\min_{X\in\mathbb{R}^{m\times n}} \|X\|_*, \qquad \text{s.t., } X_\Omega = Z_\Omega. \qquad (4)$$

Recently, many low-complexity algorithms have emerged, such as APGL [17], IALM [12], and FPCA [13]. Involving SVDs in their each iteration, thus those algorithms based on the soft-thresholding operator suffer from high computational cost. This limits the usage of the matrix completion techniques in real-world applications.

Alternatively, low-rank matrix completion based on fixed-rank matrix factorization has received a significant amount of attention [15]. Suppose that the low-rank matrix $X \in \mathbb{R}^{m\times n}$ with rank r is decomposed as $X = UM$, where $U \in \mathbb{R}^{m\times r}$ and $M \in \mathbb{R}^{r\times n}$. LMaFit [22] applies a successive over-relaxation iteration scheme to alternatively solve the following least-squares problem,

$$\min_{U\in\mathbb{R}^{m\times r}} \min_{M\in\mathbb{R}^{r\times n}} \sum_{(i,j)\in\Omega} \|(UM)_{ij} - Z_{ij}\|^2. \qquad (5)$$

However, the factorization of the matrix X into the product UM is not unique. Indeed, for any r-by-r invertible matrix O, we have $UM = (UO)(O^{-1}M)$. Hence, some researchers convert the matrix factorization problem into some corresponding Riemannian manifold optimization problems, such as OptSpace [10], RTRMC [4], RiemannCG [18], and ScGrass [15]. However, in those algorithms we need to know the exact rank which is usually difficult to obtain. Furthermore, they often suffer badly from overfitting due to their least-squares loss functions, especially on noisy matrices.

2.1 Grassmannian Manifold

We will briefly recall the related notions of matrix manifolds (readers may refer to [2] for details).

Definition 1. ***Grassmannian manifold:*** *The set of* r*-dimensional vector subspaces of $\mathbb{R}^m$ is defined as $\mathcal{G}_{m,r}$. Each point $\mathcal{U} \in \mathcal{G}_{m,r}$ can be presented by a generator matrix $U \in \mathcal{N}_{m,r}$, where $\mathcal{N}_{m,r}$ is the set of $m \times r$ matrices with orthonormal columns, i.e., $\mathcal{N}_{m,r} = \{U \in \mathbb{R}^{m\times r} : U^T U = I_r\}$.*

Definition 2. ***Tangent space:*** *Consider an arbitrary point on the Grassmannian manifold, $\mathcal{U} \in \mathcal{G}_{m,r}$. To perform differential calculus, the tangent space at U (the generator matrix of $\mathcal{U}$) is denoted as $T_U\mathcal{G}_{m,r}$. And the tangent space is represented as $T_U\mathcal{G}_{m,r} = \{\eta \in \mathbb{R}^{m\times r} : U^T\eta = 0\}$.*

As the generalization of the standard optimization methods, some Riemannian manifold optimization methods can be used for solving the following low-rank matrix learning problem with the fixed rank *r*,

$$\min_{U\in\mathcal{G}_{m,r}} f(U), \qquad (6)$$

where $f(\cdot)$ is a smooth function on Grassmannian manifolds.

2.2 Skeleton of CG Algorithms on Grassmannian Manifolds

In general, the typical nonlinear conjugate gradient (CG) algorithm on Grassmannian manifolds with a line-search rule for the unconstrained optimization problem (6) is outlined in **Algorithm 1**, which we elaborate as follows.

- Ambient gradient: To obtain the Euclidean gradient $\nabla f(U_k)$ in the ambient space.
- Riemannian gradient: It, denoted by $\text{grad}f(U_k)$, is a specific tangent vector η_k which corresponds to the

Algorithm 1 Geometric CG

```
Input: The fixed rank $r$ and tol $> 0$.
Output: $X = UM$.
while not converged do
    Compute the ambient gradient: $\nabla f(U_k)$.
    Compute the Grassmannian gradient:
    grad$f(U_k)$.
    Check convergence: $\|\text{grad} f(U_k)\| \le$ tol.
    Compute $\beta_k$ by the PR+ updating rule,
    and compute a conjugate direction $\xi_k$:
        $\xi_k = -\text{grad} f(U_k) + \beta_k T_{U_{k-1} \to U_k} \mathcal{G}_{m,r}(\xi_{k-1})$.
    Find an appropriate step size $t_k$ and compute
    $U$: $U_{k+1} = \Re_{U_k}(t_k \xi_k)$.
end while
```

direction of steepest ascent of $f(U_k)$, but is restricted to only directions in the tangent space $T_{U_k}\mathcal{G}_{m,r}$.

- The conjugate direction: It, denoted by $\xi_k \in T_{U_k}\mathcal{G}_{m,r}$, is conjugate to the gradient, and requires taking a linear combination of the Riemannian gradient with the previous search direction ξ_{k-1}. Since ξ_{k-1} does not lie in $T_{U_k}\mathcal{G}_{m,r}$, it needs to be transported to $T_{U_k}\mathcal{G}_{m,r}$. This is done by a mapping $T_{U_{k-1} \to U_k}\mathcal{G}_{m,r} : T_{U_{k-1}}\mathcal{G}_{m,r} \to T_{U_k}\mathcal{G}_{m,r}$, the so-called *vector transport*. In total, the conjugate direction $\xi_k = -\text{grad} f(U_k) + \beta_k T_{U_{k-1} \to U_k}\ \mathcal{G}_{m,r}(\xi_{k-1})$ can be computed by a variant of the classical Polak-Ribière (PR+) updating rule in the non-linear CG.

- Retraction: Because a tangent vector only gives a direction but not the line search itself on the manifold, a smooth mapping $\Re_{U_k} : T_{U_k}\mathcal{G}_{m,r} \to \mathcal{G}_{m,r}$, named as *retraction*, is needed to map tangent vectors to the manifold. To retract the search direction ξ_k with a line-search step size t_k back to the manifold is denoted as: $U_{k+1} = \Re_{U_k}(t_k \xi_k)$.

3 Grassmannian Optimization

3.1 Linearization Technique

As in [17], the problem (1) can be approximated iteratively by minimizing the following linearized function,

$$\begin{aligned}\mathcal{L}(X) =& \mu\|X\|_* + g(X_k) + \langle \nabla g(X_k), X - X_k \rangle \\ &+ \frac{1}{2\tau}\|X - X_k\|_F^2,\end{aligned} \tag{7}$$

where $\tau > 0$ is a proximal parameter. Without loss of generality, suppose d is an upper bound for rank$(X) = r$, i.e., $r \le d$. $X \in \mathbb{R}^{m\times n}$ is decomposed as $X = UM$, where $U \in \mathcal{N}_{m,d}$ and $M \in \mathbb{R}^{d\times n}$. Furthermore, the quotient geometry (i.e., Grassmannian manifold) is used in our paper to guarantee the uniqueness of matrix factorization. Hence, $U \in \mathcal{N}_{m,d}$ can be viewed as the generator matrix of $\mathcal{U} \in \mathcal{G}_{m,d}$ and is an orthonormal basis of $\mathcal{U}$. With $U^T U = I$, we have $\|X\|_* = \|M\|_*$. Thus, the problem (7) is rewritten in the following form

$$\begin{aligned}\mathcal{L}(U, M) = \mu\|M\|_* + \langle \nabla g(U_k M_k), UM - U_k M_k \rangle \\ + g(U_k M_k) + \frac{1}{2\tau}\|UM - U_k M_k\|_F^2.\end{aligned} \tag{8}$$

For solving the problem (8), then we formulate the following subproblem at the k-th iteration,

$$\begin{aligned}\min_{\mathcal{U}\in\mathcal{G}_{m,d}} \min_{M\in\mathbb{R}^{d\times n}} \widetilde{f}_{U_k M_k}(U, M) := \\ \mu\tau\|M\|_* + \frac{1}{2}\|UM - U_k M_k + \tau\nabla g(U_k M_k)\|_F^2.\end{aligned} \tag{9}$$

In the following, the problem (9) is equally converted into a Grassmannian manifold optimization problem with respect to U.

3.2 Objective Function on Grassmannian Manifolds

Similar to [4], we now derive the objective function on Grassmannian manifolds. Given the variable U, computing the matrix M that minimizes $\widetilde{f}_{U_k M_k}$ is a nuclear norm regularized least-squares problem. The mapping between U and this (unique) optimal M, denoted by M_U, is given by

$$\begin{aligned} U \mapsto M_U = \\ \arg\min_{M\in\mathbb{R}^{d\times n}} \mu\tau\|M\|_* + \frac{1}{2}\|UM - P_k\|_F^2,\end{aligned} \tag{10}$$

where $P_k = U_k M_k - \tau\nabla g(U_k M_k)$. Following [5], we can obtain a unique closed-form solution to the problem (10) via the SVT operator,

$$M_U = \text{SVT}_{\mu\tau}(U^T P_k), \tag{11}$$

where $\text{SVT}_{\mu\tau}(A) := \overline{U}\text{diag}(\max\{\sigma - \mu\tau, 0\})\overline{V}$ and $\overline{U}\text{diag}(\sigma)\overline{V}$ is the SVD of A. Substituting (11) into the function $\widetilde{f}_{U_k M_k}$, then the cost function $f_{U_k M_k} : \mathcal{G}_{m,r} \to \mathbb{R}$ on Grassmannian manifolds is given by

$$\min_{\mathcal{U}\in\mathcal{G}_{m,d}} f_{U_k M_k}(U) := \mu\tau\|M_U\|_* + \frac{1}{2}\|UM_U - P_k\|_F^2. \tag{12}$$

3.3 Riemannian Gradient

For solving our problem (12), we first derive the formulas for the Euclidean gradient of the cost function $f_{U_k M_k}$ in (12) at U. Using the chain rule, we have

$$\begin{aligned}&\nabla f_{U_k M_k}(U) = \frac{\text{d}}{\text{d}U} f_{U_k M_k}(U) \\ =& \frac{\partial}{\partial U}\widetilde{f}_{U_k M_k}(U, M_U) + \frac{\partial}{\partial M_U}\widetilde{f}_{U_k M_k}(U, M_U)\frac{\text{d}}{\text{d}U}M_U,\end{aligned} \tag{13}$$

where $\widetilde{f}_{U_k M_k}(U, M_U)$, $f_{U_k M_k}(U)$ and the map M_U have been defined in (9), (12) and (11), respectively. The first term of (13), $\frac{\partial}{\partial U}\widetilde{f}_{U_k M_k}(U, M_U)$, can be computed easily. To compute the second term in (13), i.e., $\frac{\partial}{\partial M_U}\widetilde{f}_{U_k M_k}(U, M_U)\frac{\mathrm{d}}{\mathrm{d}U}M_U$, we will present the following derivation using the singular value and singular subspace perturbation theories.

3.3.1 Computation of Ambient Gradient

To compute the ambient gradient, we first introduce the following two definitions and give their property, respectively.

Definition 3. ***Subdifferential**: Let $\partial_M \widetilde{f}_{U_k M_k}(U, M)$ denote the subdifferential of the non-smooth function $\widetilde{f}_{U_k M_k}$ (U, M) at M, then*

$$\partial_M \widetilde{f}_{U_k M_k}(U, M) = \mu\tau\partial\|M\|_* + (M - U^T P_k), \quad (14)$$

where $\partial\|\cdot\|_*$ denotes the subdifferential of the non-smooth convex function $\|\cdot\|_*$, and is a closed convex set. Specifically, let $M = \hat{U}\hat{\Lambda}\hat{V}$ be the SVD of $M \in \mathbb{R}^{d\times n}$, then $\partial\|M\|_*$ is given by [5], i.e.,

$$\begin{aligned}&\partial\|M\|_* = \\ &\{\hat{U}\hat{V} + W : \hat{U}^T W = 0, W\hat{V}^T = 0, \|W\|_2 \leq 1\},\end{aligned} \quad (15)$$

where $\|\cdot\|_2$ is a spectrum norm.

By Definition 3, we can obtain the following property.

Lemma 1. *Let M_U be the solution of problem (10), $M_U = \tilde{U}\tilde{\Lambda}\tilde{V}$ be the SVD of M_U, and $\Gamma = \{W : \tilde{U}^T W = 0, W\tilde{V} = 0, \|W\|_2 \leq 1\}$, then $\exists\tilde{W} \in \Gamma$ satisfies*

$$\partial_{M_U} \widetilde{f}_{U_k M_k} = \{\mu\tau(W - \tilde{W}),\ W \in \Gamma\}. \quad (16)$$

Proof. Since M_U is a optimal solution, then the first-order optimality condition of the problem (10) is given by,

$$0 \in \mu\tau\partial\|M_U\|_* + (M_U - U^T P_k). \quad (17)$$

By (15), then $\exists\tilde{W} \in \Gamma$ satisfies

$$\mu\tau\widetilde{U}\widetilde{V} + \mu\tau\widetilde{W} + (M_U - U^T P_k) = 0. \quad (18)$$

Furthermore, substituting (18) into the subdifferential in (14), we have

$$\begin{aligned}&\partial_M \widetilde{f}_{U_k M_k} = \mu\tau\partial\|M_U\|_* + (M_U - U^T P_k) \\ =&\{\mu\tau\tilde{U}\tilde{V}^T + \mu\tau W + M - U^T P_k,\ W \in \Gamma\} \\ =&\{\mu\tau(W - \tilde{W}),\ W \in \Gamma\}.\end{aligned} \quad (19)$$

This completes the proof. □

Definition 4. ***Directional Derivative**: Let $M_U = SVT_{\mu\tau}(U^T P_k)$, the directional derivative of the mapping M_U at U along the direction H is defined as*

$$\begin{aligned}&M_{U,H} = \\ &\lim_{\gamma\to 0}\frac{SVT_{\mu\tau}((U+\gamma H)^T P_k) - SVT_{\mu\tau}(U^T P_k)}{\gamma}.\end{aligned} \quad (20)$$

Furthermore, we give the following result by the singular value and singular subspaces perturbation theorems.

Lemma 2. *With the same notations as Lemma 1, then for any matrix $W \in \Gamma$, we have*

$$\langle M_{U,H}, W\rangle = 0. \quad (21)$$

Proof. To prove the lemma, the classical perturbation theory for singular value and singular subspaces problems is introduced. We use the classical result of [20] that the eigenvalues of a matrix which is an analytic function of a single variable can always be numbered so that they are each analytic functions of the variable. Using the relationship between eigenvalues and singular values, it follows that if the singular values of the matrix $B = A + \gamma R$, where A and R are $m \times n$ matrices, denoted by $\sigma_i(\gamma), i = 1, 2, \ldots, n$, then

$$\sigma_i(\gamma) = \sigma_i + \gamma u_i^T R v_i + O_i(\gamma), i = 1, 2, \ldots n, \quad (22)$$

where $O_i(\gamma)$ is an infinitesimal of higher order than γ, u_i and v_i are singular vectors of A corresponding to σ_i. Let $A = \widetilde{U}\widetilde{\Sigma}\widetilde{V} + \widetilde{U}'\widetilde{\Sigma}'\widetilde{V}'$ and $B = \hat{U}\hat{\Sigma}\hat{V} + \hat{U}'\hat{\Sigma}'\hat{V}'$ be the SVDs of A and B respectively, where $\widetilde{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_s)$ and $\hat{\Sigma} = \mathrm{diag}(\sigma_1(\gamma), \ldots, \sigma_s(\gamma))$ are s largest singular values of A and B, respectively. $\widetilde{\mathcal{U}}$ and $\hat{\mathcal{U}}$ denote the spaces of $\widetilde{U}$ and $\hat{U}$, and $\widetilde{\mathcal{V}}$ and $\hat{\mathcal{V}}$ denote the spaces of $\widetilde{V}$ and $\hat{V}$, respectively. Then the classic theorem on the perturbation of singular subspaces is due to [21],

$$\begin{aligned}&\sqrt{\|\sin\Theta(\widetilde{\mathcal{U}}, \hat{\mathcal{U}})\|_F^2 + \|\sin\Theta(\widetilde{\mathcal{V}}, \hat{\mathcal{V}})\|_F^2} \\ \leq&\frac{\sqrt{\|E_1\|_F^2 + \|E_2\|_F^2}}{\delta},\end{aligned} \quad (23)$$

where $E_1 = B\widetilde{V} - \widetilde{U}\widetilde{\Sigma} \equiv \gamma R\widetilde{V}$, $E_2 = B^T\widetilde{U} - \widetilde{V}\widetilde{\Sigma} \equiv \gamma R^T\widetilde{U}$, and $\|\sin\Theta(\widetilde{\mathcal{U}}, \hat{\mathcal{U}})\|_F^2$ is a measure that is related to the canonical angles between the subspace $\widetilde{\mathcal{U}}$ and $\hat{\mathcal{U}}$. Moreover, the gap δ is the distance between two sets of singular values in $\widetilde{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_s)$ and $\widetilde{\Sigma}' = \mathrm{diag}(\sigma_{s+1}, \ldots, \sigma_n)$ (Please see the details in [21]).

Let $A := U^T P_k$, $R := H^T P_k$, $B := A + \gamma R = U^T P_k + \gamma H^T P_k$, $\sigma_1 \geq \ldots \geq \sigma_s \geq \mu\tau$ and $\sigma_{s+1} < \mu\tau$. By using the result in (22) and the definition of the SVT operator with $\gamma \to 0$, we have

$$\begin{aligned}&\mathrm{SVT}_{\mu\tau}((U+\gamma H)^T P_k) - \mathrm{SVT}_{\mu\tau}(U^T P_k) \\ =&\hat{U}(\hat{\Sigma} - \mu\tau)\hat{V} - \widetilde{U}(\widetilde{\Sigma} - \mu\tau)\widetilde{V} \\ =&T_1 + T_2 + O(\gamma),\end{aligned} \quad (24)$$

where $T_1 = \hat{U}(\widetilde{\Sigma} - \mu\tau)\hat{V} - \widetilde{U}(\widetilde{\Sigma} - \mu\tau)\widetilde{V}$, $T_2 = \gamma\hat{U}\Delta\hat{V}$, the i-th element of the diagonal matrix Δ is $\Delta_i = \widetilde{u}_i^T H^T P_k \widetilde{v}_i$, $O(\gamma) \in \mathbb{R}^{d\times n}$ and its all entries are infinitesimals of higher order than γ.

By the singular subspace perturbation theory of in (23), the subspace $\hat{\mathcal{U}} \to \widetilde{\mathcal{U}}$, while $\gamma \to 0$, i.e., $\exists D_1$, such that $\widehat{U} \to$

$\widetilde{U}D_1$. Similarly, $\exists D_2$, such that $\widehat{V} \to \widetilde{V}D_2$. Thus, it is not difficult to verify $\widehat{U} = \widetilde{U}D_1 + \delta_1(\gamma)$ and $\widehat{V}^T = \widetilde{V}^T D_2 + \delta_2(\gamma)^T$, where $\delta_1(\gamma) \in \mathbb{R}^{m\times d}$ and $\delta_2(\gamma) \in \mathbb{R}^{d\times n}$, and all of their entries are infinitesimals of the same order as γ. Then we have

$$\begin{aligned}
&\langle W, \hat{U}(\widetilde{\Sigma} - \mu\tau)\hat{V}\rangle = \langle \hat{U}^T W \hat{V}^T, (\widetilde{\Sigma} - \mu\tau)\rangle \\
=&\langle D_1^T \widetilde{U}^T W \hat{V}^T, (\widetilde{\Sigma} - \mu\tau)\rangle \\
&+ \langle \delta_1(\gamma)^T W \widetilde{V}^T D_2, (\widetilde{\Sigma} - \mu\tau)\rangle \\
&+ \langle \delta_1(\gamma)^T W \delta_2(\gamma), (\widetilde{\Sigma} - \mu\tau)\rangle,
\end{aligned} \tag{25}$$

where W is defined in (15), $\widetilde{U}^T W = 0$, $W\widetilde{V}^T = 0$, and by (25), then

$$\begin{aligned}
\langle W, T_1\rangle &= \langle W, \hat{U}(\widetilde{\Sigma} - \mu\tau)\hat{V}\rangle + \langle W, \widetilde{U}(\widetilde{\Sigma} - \mu\tau)\widetilde{V}\rangle \\
&= \langle \delta_1(\gamma)^T W \delta_2(\gamma), (\widetilde{\Sigma} - \mu\tau)\rangle.
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
&\langle W, T_2\rangle = \langle W, \gamma\hat{U}\Delta\hat{V}\rangle \\
=&\langle D_1^T \widetilde{U}^T W \hat{V}^T, \gamma\Delta\rangle + \langle \delta_1^T W \widetilde{V}^T D_2, \gamma\Delta\rangle \\
&+ \langle \delta_1^T(\gamma) W \delta_2(\gamma), \gamma\Delta\rangle \\
=&\langle \delta_1^T(\gamma) W \delta_2(\gamma), \gamma\Delta\rangle.
\end{aligned}$$

Thus, we have

$$\langle W, M_{U,H}\rangle = \lim_{\gamma\to 0}\frac{\langle W, T_1 + T_2 + O(\gamma)\rangle}{\gamma} = 0.$$

Thus, this completes the proof. □

Next we will compute the ambient gradient. Let $\forall\zeta \in \frac{\partial}{\partial M_U}\widetilde{f}_{U_k M_k}(U, M_U)\frac{\mathrm{d}}{\mathrm{d}U}M_U$, by the chain rule of composite function, and substituting the result in Lemma 1 into the chain rule, then $\exists(W - \widetilde{W}) \in \Gamma$ satisfies

$$\begin{aligned}
\zeta_{ij} &= \langle W - \widetilde{W}, M_{U,\widetilde{H}^{ij}}\rangle, \\
i &= 1, 2, \ldots, m, \quad j = 1, 2, \ldots, d,
\end{aligned}$$

where ζ_{ij} denotes the element in the i-th row and the j-th column of ζ, and $M_{U,\widetilde{H}^{ij}}$ is given by Definition 2, and the direction $\widetilde{H}^{ij} \in \mathbb{R}^{m\times d}$ is defined as

$$(\widetilde{H}^{ij})_{m,n} = \begin{cases} 1 & m = i \text{ and } n = j, \\ 0 & \text{otherwise.} \end{cases}$$

And by Lemma 2, then

$$\zeta_{ij} = \langle W - \widetilde{W}, M_{U,\widetilde{H}^{ij}}\rangle = 0. \tag{26}$$

Thus, we have

$$\frac{\partial}{\partial M_U}\widetilde{f}_{U_k M_k}(U, M_U)\frac{\mathrm{d}}{\mathrm{d}U}M_U = 0. \tag{27}$$

By (27), then the ambient gradient in (13) can be rewritten as follows:

$$\begin{aligned}
\nabla f_{U_k M_k}(U) &= \frac{\partial}{\partial U}\widetilde{f}_{U_k M_k}(U, M_U) \\
&= (UM_U - P_k)M_U^T.
\end{aligned} \tag{28}$$

Note the above result implies that the function $f_{U_k M_k}(\cdot)$ is continuously differentiable.

3.3.2 Computation of Riemannian Gradient

Following [18], and $(I - UU^T)U = 0$, then the Grassmannian gradient of $f_{U_k M_k}$ at U is given by

$$\begin{aligned}
\mathrm{grad} f_{U_k M_k}(U) &= (I - UU^T)\nabla f_{U_k M_k}(U) \\
&= -(I - UU^T)P_k M_U^T.
\end{aligned} \tag{29}$$

3.4 Conjugate Gradient Iteration

In the part, we describe the nonlinear CG algorithm on the Grassmannian manifold for solving the proposed model. The main additional ingredient we need is vector transport which is used to transport the old search direction to the current point on the manifold, i.e., $T_{U_{k-1}\to U_k}\mathcal{G}_{m,d} : T_{U_{k-1}}\mathcal{G}_{m,d} \to T_{U_k}\mathcal{G}_{m,d}$. The transport search direction is then combined with the gradient at the current point, e.g. by the Polak-Ribière formula (see [2]), to derive the new search direction. Vector transport can be defined using the Riemann connection, which in turn is defined based on the Riemann metric [1]. In this paper, we will use the canonical metric to derive vector transport when considering the natural quotient manifold structure of the Grassmannian manifold. Following [15], the previous search direction ξ_{k-1} at U_{k-1} will be transported to U_{k+1} as $T_{U_{k-1}\to U_k}\mathcal{G}_{m,d} = (I - U_k U_k^T)\xi_{k-1}$. Then the new search direction is

$$\xi_k = -\mathrm{grad} f(U_k) + \beta_k T_{U_{k-1}\to U_k}\mathcal{G}_{m,r}(\xi_{k-1}), \tag{30}$$

where β_k can be calculated by using the Polak-Ribiere formula in [7].

Furthermore, U is updated by

$$U_{k+1} = R(U_k + t_k\xi_k) = \mathrm{qf}(U_k + t_k\xi_k), \tag{31}$$

where $\mathrm{qf}(A)$ is used as a retraction operator, which is the Q factor in the QR factorization of A, and the step size t_k is obtained by the Armijo linear search rule [2].

To solve the Riemannian optimization subproblem (12) at each iteration, we present a non-linear conjugate gradient decent algorithm on Grassmannian manifolds. Overall, the skeleton of our method is listed in **Algorithm 2**.

3.5 Convergence Analysis

In this part, we analyze the convergence of Algorithm 2 using the non-linear conjugate gradient descent scheme.

Algorithm 2 A Riemannian optimization framework for solving the problem (12)

```
Input: The rank $d$, the parameters $\mu, \tau$ and tol.
Output: $X = UM$.
while not converged do
    Formulate the cost function $f_{U_k M_k}$ by (12).
    Compute the Grassmannian gradient by (29),
        $\eta_k = \text{grad} f_{U_k M_k}(U_k)$.
    Check convergence: $\eta_k \leq$ tol.
    Compute a conjugate direction $\xi_k$ by (30).
    Find an appropriate step size $t_k$ using Armijo rule,
    and compute $U_{k+1}$ by (31).
    Compute $M_{k+1}$ by (11).
end while
```

Lemma 3. *Let $g(X) = \|\mathcal{A}(X) - b\|_F^2$, $\{(U_k, M_k)\}$ be an infinite sequence of iterates generated by Algorithm 2 with the Armijo backtracking rule, and $\tau \in (0, 1/\rho(\mathcal{A}^T\mathcal{A}))$, where $\rho(\cdot)$ denotes the spectral radius operator, then we have the following results:*
(I) $\lim_{k\to\infty} \|\textit{grad}f_{U_k M_k}(U_k)\| = 0$.
(II) $\lim_{k\to\infty} \|U_{k+1} - U_k\| = 0$, *and*

$$\lim_{k\to\infty} \|U_{k+1}M_{k+1} - U_k M_k\| = 0.$$

Proof: The detailed proof can be found in the supplementary material.

Theorem 4. *Let $\{(U_k, M_k)\}$ be an infinite sequence of iterates generated by Algorithm 2. Then each accumulation point of $\{(U_k, M_k)\}$ is a critical point of the following optimization problem*

$$\min_{\mathcal{U}\in\mathcal{G}_{m,d}} \min_{M\in\mathbb{R}^{d\times n}} \mu\tau\|M\|_* + \frac{1}{2}g(UM). \tag{32}$$

Proof: The detailed proof of the theorem can be found in the the supplementary material.

3.6 Complexity Analysis

In this part, we discuss the time complexity of our method. For the matrix completion problem (12), the main running time of our algorithm is consumed by performing SVD for the SVT operator, some multiplications and retraction operator. The time complexity of performing the SVT operator in (11) is $O_1 := O(d^2n)$. The time complexity of some multiplication and retraction operators is $O_2 := O(dmn + d^2m)$. The time complexity of performing retraction operator is $O_3 := O(d^2m)$. Thus, the total time complexity of our method is $O(T(O_1 + O_2 + O_3))$, where T is the number of iterations.

4 Graph Regularization Extensions

In this paper, we mainly consider the problem of recovering a noisy low-rank matrix from a few observed entries as a matrix completion application of our nuclear norm regularized least squares model. In addition, our method is quite general, and can be easily extended to incorporate the contextual information, including social relations of users, social tags issued by users, movie genres, user demographic information, etc. In order to incorporate the social network information, our social network aided context-aware recommender model is formulated as follows:

$$\min_{U\in\mathcal{N}_{m,d}} \min_{M\in\mathbb{R}^{d\times n}} f(U,M) := \frac{1}{2}\|\mathcal{P}_\Omega(UM) - \mathcal{P}_\Omega(Z)\|_F^2 + \mu\|M\|_* + \frac{\lambda_1}{2}\text{tr}(U^T L_U U) + \frac{\lambda_2}{2}\text{tr}(M L_M M^T), \tag{33}$$

where tr(A) denotes the trace of the matrix A, L_U and L_M are the graph Laplacian matrices, i.e., $L_U = D_U - W_U$, W_U is the weight matrix for the user set, and D_U is the diagonal matrix whose entries are column sums of W_U, i.e., $(D_U)_{ii} = \sum_j (W_U)_{ij}$, and $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$ are regularization constants. It is not difficult to verify, for any matrix $O \in \mathcal{N}_{d,d}$, we have $f(UO, O^T M) = f(U, M)$. Hence, the Grassmannian manifold is also used in our social network aided context-aware recommender model, and it is reformulated as follows:

$$\min_{\mathcal{U}\in\mathcal{G}_{m,d}} \min_{M\in\mathbb{R}^{d\times n}} \frac{1}{2}\|\mathcal{P}_\Omega(UM) - \mathcal{P}_\Omega(Z)\|_F^2 + \mu\|M\|_* + \frac{\lambda_1}{2}\text{tr}(U^T L_U U) + \frac{\lambda_2}{2}\text{tr}(M L_M M^T), \tag{34}$$

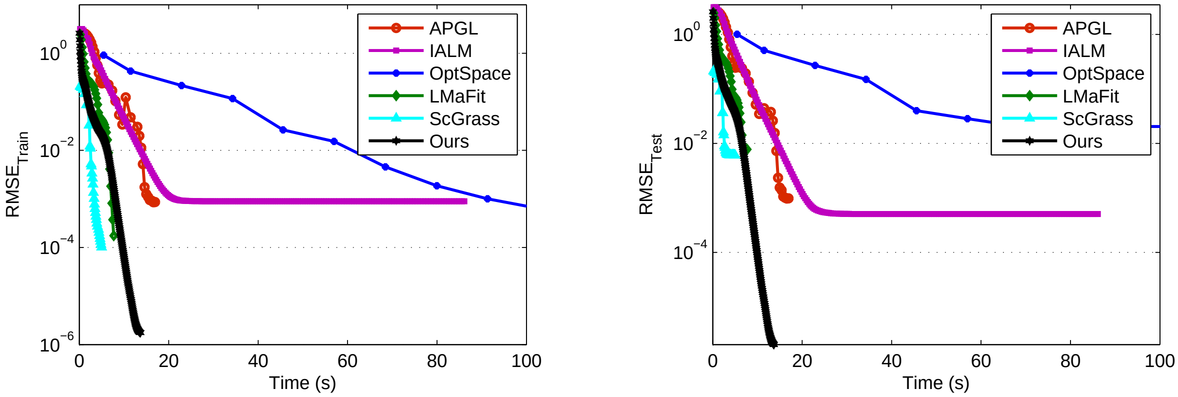
where the column orthonormal matrix U is viewed as the generator matrix of $\mathcal{U}$. Moreover, Algorithm 2 can be extended to solve our graph regularized matrix completion problem (34).

5 Experimental Results

In this section, we evaluate both the effectiveness and efficiency of our method for solving matrix completion problems on both synthetic and real-world data.

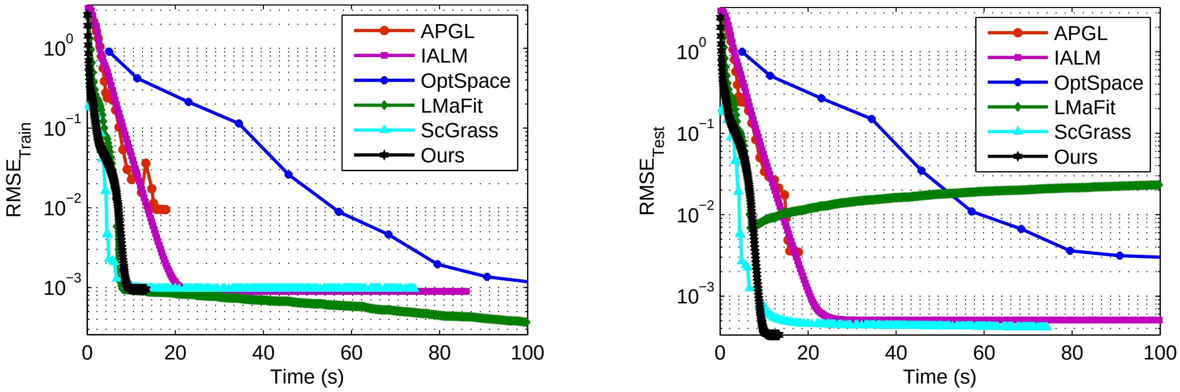
5.1 Synthetic Data

The synthetic matrices $Z \in \mathbb{R}^{m\times n}$ with rank r in this subsection are created randomly by the following procedure: two random matrices $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ with i.i.d. standard Gaussian entries are first generated, and then $X = UV^T$ is assembled. Two test experiments are conducted on random matrix without or with noise, where the observed subset is corrupted by i.i.d. standard Gaussian random variables as in [17]. In both cases, only 10% observed entries are sampled uniformly at random as the training set, and the remaining is used as the testing set. Summaries of the computational results are presented in Figure 1 on noiseless matrices of size 2000 × 2000 and



(a) (b)

Figure 1: The recovery accuracy on noiseless data vs. running time (seconds): training RMSE (left) and testing RMSE (right).



(a) (b)

Figure 2: The recovery accuracy on noisy data (where the noise level $nf = 0.001$) vs. running time (seconds): training RMSE (left) and testing RMSE (right).

Figure 2 on noisy matrices of size 2000×2000, respectively.

We compare our method with APGL[1] [17], IALM[2] [12], OptSpace[3] [10], LMaFit[4] [22], and ScGrass[5] [15] on the noiseless or noisy matrices, and illustrate the training and testing recovery accuracies (RMSE) in Figures 1 and 2, respectively, where APGL and IALM use the PROPACK package [11] to compute a partial SVD. All the methods are implemented in Matlab and use mex files. In terms of running time, the results show that for the noiseless data, our method, LMaFit and ScGrass converge much faster than the other three methods including APGL, IALM, and OptSpace. However, for the noisy data, the testing RMSE of LMaFit becomes worse due to overfitting while the training RMSE gradually decreases.

We also test the robustness of all these methods against the noise, and demonstrate the experimental results (the testing RMSE and running time) in Figure 3. It is clear that when the noise level is higher, our method usually outperforms the other methods in terms of the testing RMSE, that is, our method has the good generation ability. With the increase of the noise level, the running time of the other algorithms dramatically grows except for our method and OptSpace. In contrast, the runtime of our method increases slightly.

[1] http://www.math.nus.edu.sg/~mattohkc/NNLS.html

[2] http://www.cis.pku.edu.cn/faculty/vision/zlin/zlin.htm

[3] http://web.engr.illinois.edu/~swoh/software/optspace/

[4] http://lmafit.blogs.rice.edu/

[5] http://www-users.cs.umn.edu/~thango/

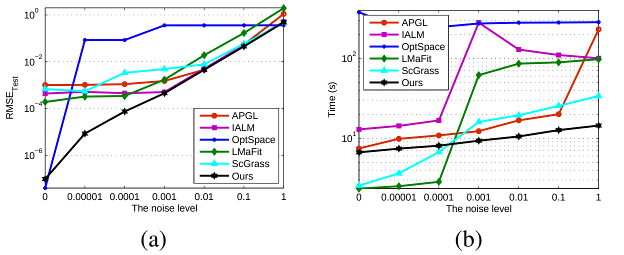


Figure 3: The recovery results vs. the noise level: RMSE (left) and running time (right).

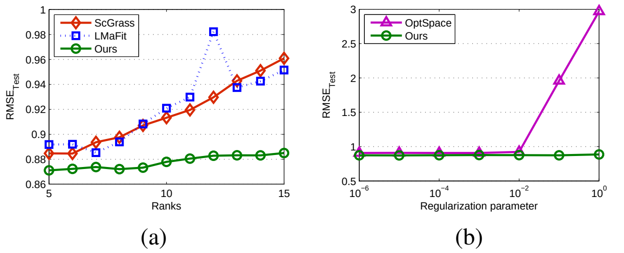


Figure 4: Results of our method with varying parameter values on the MovieLens1M data set.

5.2 Real-World Data

In order to evaluate our method, experiments were conducted on three widely used recommendation systems data sets[6]: MovieLens100K (ML-100K) with 100K ratings, MovieLens1M (ML-1M) with 1M ratings, and MovieLens10M (ML-10M) with 10M ratings. We randomly split these three data sets to train and test sets such that the ratio of the train set to test set is 9:1, and the experimental results are reported over 20 independent runs. Except for APG, IALM, OptSpace, and LMaFit, we also compare our method with two state-of-the-art optimization methods on manifolds: ScGrass and RTRMC[7] [4]. For our method, we set the rank $d = 5, 6, 7$, and $\mu = 10^{-2}$. The stopping tolerance for all algorithms is set to $\varepsilon = 10^{-4}$. All other parameters are set to their default values for the algorithms that we compare with. We also use the Root Mean Squared Error (RMSE) as the evaluation measure.

The average RMSE on these three data sets is reported over 20 independent runs and is shown in Table 1. The results show that for some fixed ranks, the matrix factorization methods including OptSpace, ScGrass, RTRMC, LMaFit and our method usually perform better than two nuclear norm minimization methods including APGL and IALM. As expected, our method with $d = 5$ on the MovieLens (1M) data set achieved a RMSE of 0.8711, slightly outperforming the well-known restricted Boltzeman machines's RMSE of 0.8817 [16]. Moreover, our matrix factorization method with nuclear norm regularization consistently outperforms the other matrix factorization methods including OptSpace, ScGrass, RTRMC and LMaFit, and the two nuclear norm minimization methods including APGL and IALM. This confirms that the proposed matrix factorization model with nuclear norm regularization can avoid the over-fitting problems of matrix factorization.

Furthermore, we also analyze the robustness of our method with regard to its parameters: the given rank and the regularization parameter μ on the MovieLens1M data set, as shown in Figure 4, from which we can see that our method is robust against variations in its parameters. For comparison, we also show the results of two related methods: ScGrass and LMaFit with varying ranks in Figure 4(a). It is clear that, by increasing the number of the given ranks, the RMSE of ScGrass and LMaFit becomes worse. In contrast, the RMSE of our method increases slightly when the number of the given ranks increases. This further confirms that our matrix factorization model with nuclear norm regularization is effective and can avoid overfitting. OptSpace also has a spectral regularization version: $\min_{U,S,V}(1/2)\|\mathcal{P}_\Omega(USV^T - X)\|_F^2 + \mu\|S\|_F^2$. From Figure 4(b), we observe that our method is much more robust than OptSpace in terms of the regularization parameter μ.

Finally, we conduct the running time comparison of all those algorithms on the MovieLens100K and MovieLens1M data sets, as shown in Figure 5. The experiments were performed with Matlab 7.11 on an Intel Core 2 Duo (2.33 GHz) PC running Windows 7 with 2GB main memory. From the results shown in Figure 5, we can observe that our method, ScGrass, RTRMC, and LMaFit are much faster than the other three state-of-the-art algorithms including APGL, IALM and OptSpace. For APGL and IALM, SVD-related calculations essentially dominate their total costs. Therefore, avoiding SVD-related calculations on relative large-scale matrices is a main reason why our method is much faster than the nuclear norm minimization algorithms such as APGL and IALM, validating our original motivation of solving the matrix factorization model with nuclear norm regularization.

5.3 The Impact of Social Context

We also investigate the effects of social context on the MovieLens100K data set, which is suitable to evaluate the impacts of user demographic information and item genre information because it consists of demographic information (e.g. gender, age and occupation) of users and genre of movies. According to [8], a two dimensional feature vector is used to characterize the user's gender, that is, if the user is male, then the first feature is 1 while the second is 0, and vice versa. The users are partitioned into 7 age group: 1-17,

[6] http://www.grouplens.org/node/73

[7] http://perso.uclouvain.be/nicolas.boumal/RTRMC/

Table 1: RMSE of different methods on three data sets: MovieLens 100K, MovieLens 1M, and MovieLens 10M.

Methods	MovieLens (100K)			MovieLens (1M)			MovieLens (10M)		
APGL	1.2142			1.1528			0.8637		
IALM	1.2585			1.0153			0.8989		
OptSpace	0.9411			0.9068			1.1357		
Ranks	5	6	7	5	6	7	5	6	7
ScGrass	0.9236	0.9392	0.9411	0.8847	0.8846	0.8936	0.8359	0.8290	0.8247
RTRMC	0.9837	1.0617	1.1642	0.8901	0.8906	0.8977	0.8463	0.8442	0.8386
LMaFit	0.9468	0.9540	0.9568	0.8918	0.8920	0.8853	0.8576	0.8530	0.8423
Ours	**0.9216**	**0.9243**	**0.9330**	**0.8711**	**0.8723**	**0.8738**	**0.8330**	**0.8261**	**0.8217**

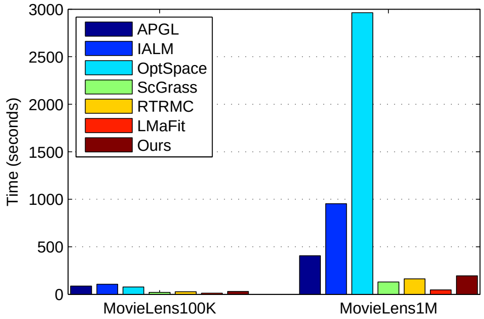
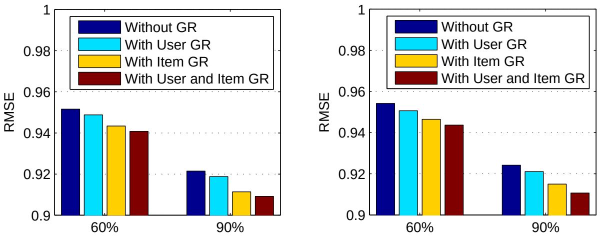


Figure 5: Running time (seconds) for comparison on the MovieLens100K and MovieLens1M data sets.

18-24, 25-34, 35-44, 45-49, 50-55, 56+. Then a seven dimensional feature vector is used to describe the user's age group. In addition, there are totally 21 occupations: administrator, doctor, educator, engineer, entertainment, executive, healthcare, homemaker, lawyer, librarian, marketing, programmer, retired, salesman, scientist, student, technician, writer, other and none. Thus, a 21 dimensional feature vector is used to describe the user's occupation. In total, a 30 dimensional feature vector is achieved for user *i*. On the other hand, there are 19 genres of movies. Likewise, we use a 19 dimensional feature vector for movie *j*. We evaluate the impact of user demographic and item genre information on this data set with 60% and 90% training sets, and we report in Fig. 6 the RMSE results yielded by our method without graph regularization, with user or item graph regularization and both graph regularizations. When with the effect of the user demographic or the item genre context, the performance of our method improves. For example, compared with our method without graph regularization, on average, our method with user or item graph regularization have 0.35% and 1.04% relative performance improvement in terms of RMSE, respectively. When with the effects of both the user demographic and the item genre context, our method obtains the best performance, suggesting that the user demographic and the item genre context contain complementary information to each other for recommendation.



(a) Rank=5

(b) Rank=10

Figure 6: The performance of variants of our method on the the MovieLens100K data set.

6 Conclusions

In this paper, we proposed a Grassmannian manifold optimization method to tackle the nuclear norm regularized least squares problems with a guarantee of local convergence, such as the noisy matrix completion problem. Our method inherits the superiority of two classes of methods, i.e. soft thresholding approaches and hard thresholding approaches, and has good generation ability. In addition, our method is extended to address the graph regularized problem. We demonstrated with convincing experimental results that our regularized formulation is effective, and our method is robust to noise or against variations in its parameters.

Acknowledgements

We thank the reviewers for their useful comments that have helped improve the paper significantly. This research has been supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF) Project Nos. CUHK 411211 and 411310, the Chinese University of Hong Kong Direct Grant Nos. 4055015 and 4055017, and Microsoft Research Asia Grant 6903555.

References

[1] P.-A. Absil, C.G. Baker, and K. Gallivan. Trust-region methods on riemannian manifolds. *Found. Comput. Math.*, 7(3):303–330, 2007.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2008.

[3] H. Avron, S. Kale, S. Kasiviswanathan, and V. Sindhwani. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *ICML*, pages 1231–1238, 2012.

[4] N. Boumal and P.-A Absil. Rtrmc: A riemannian trust-region method for low-rank matrix completion. In *NIPS*, pages 406–414, 2011.

[5] J-F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SAIM Journal on Optimization*, 20(4):1956–1982, 2010.

[6] E.J. Candes and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010.

[7] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 20(2):303–353, 1998.

[8] Q. Gu, Jie Zhou, and C. Ding. Collaborative filtering: weighted nonnegative matrix factorization incorporating user and item graphs. In *SDM*, pages 199–210, 2010.

[9] S. Ji and J. Ye. An accelerated gradient method for trace norm minimization. In *ICML*, pages 457–464, 2009.

[10] R.H. Keshavan, S. Oh, and A. Montanari. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 56(6):2980–2998, 2010.

[11] R. Larsen. Propack-software for large and sparse svd calculations. 2004.

[12] Z. Lin, M. Chen, and L. Wu. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, Univ. Illinois, Urbana-Champaign, 2009.

[13] S. Ma, D. Goldfarb, and L. Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128(1):321–353, 2011.

[14] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *J. Mach. Learn. Res.*, 11:2287–2322, 2010.

[15] T. Ngo and Y. Saad. Scaled gradients on grassmann manifolds for matrix completion. In *NIPS*, pages 1421–1429, 2012.

[16] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *ICML*, pages 791–798, 2007.

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

[18] B. Vandereycken. Low-rank matrix completion by riemannian optimization. *SAIM Journal on Optimization*, 23(2):1214–1236, 2013.

[19] Y. Wang and H. Xu. Stability of matrix factorization for collaborative filtering. In *ICML*, pages 417–424, 2012.

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

[21] P. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, 12:99–111, 1972.

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