Scalable Maximum Margin Matrix Factorization by Active Riemannian Subspace Search

Yan Yan[1], Mingkui Tan[2*], Ivor Tsang[1], Yi Yang[1], Chengqi Zhang[1] and Qinfeng Shi[2]
[1]Centre for Quantum Computation and Intelligent Systems, University of Technology, Sydney, Australia
[2]Australian Centre for Visual Technologies, The University of Adelaide, Australia
yan.yan-3@student.uts.edu.au,mingkui.tan@adelaide.edu.au,ivor.tsang@uts.edu.au,
yi.yang@uts.edu.au,chengqi.zhang@uts.edu.au, javen.shi@adelaide.edu.au

Abstract

The user ratings in recommendation systems are usually in the form of ordinal discrete values. To give more accurate prediction of such rating data, maximum margin matrix factorization (M^3F) was proposed. Existing M^3F algorithms, however, either have massive computational cost or require expensive model selection procedures to determine the number of latent factors (i.e. the rank of the matrix to be recovered), making them less practical for large scale data sets. To address these two challenges, in this paper, we formulate M^3F with a known number of latent factors as the Riemannian optimization problem on a fixed-rank matrix manifold and present a block-wise nonlinear Riemannian conjugate gradient method to solve it efficiently. We then apply a simple and efficient active subspace search scheme to automatically detect the number of latent factors. Empirical studies on both synthetic data sets and large real-world data sets demonstrate the superior efficiency and effectiveness of the proposed method.

1 Introduction

The rapid increase of Web services has witnessed an increasing demand for predicting the preferences of users on products of interest, such as movies and music tracks [Su and Khoshgoftaar, 2009]. This task, also known as the collaborative filtering (CF), is a principal task in recommender systems [Weimer *et al.*, 2008; Huang *et al.*, 2013]. In general, the user ratings are given in discrete values, including binary ratings and ordinal ratings [Srebro *et al.*, 2005]. The binary ratings can be either "+1" (*like*) or "-1" (*dislike*); while the ordinal ratings are in discrete values such as 1-5 "stars", which are more popular in applications.

Given a small number of user ratings $\mathbf{Y} \in \mathbb{R}^{m\times n}$ (from m users on n items), the aim of CF is to reconstruct the unobserved ratings. Let Ω be a subset containing the indices of the observed entries. To perform the reconstruction, a common approach is to learn a low-rank matrix $\mathbf{X}$ to fit $\mathbf{Y}$ by solving the following optimization problem:

$$\min_{\mathbf{X}} f(\mathbf{X}), \;\; s.t. \;\; \text{rank}(\mathbf{X}) \leq k, \tag{1}$$

where k denotes the number of latent factors (i.e. the rank of $\mathbf{X}$) and $f(\mathbf{X})$ denotes some loss functions. The low-rank property has been studied in a variety of applications [Xiao *et al.*, 2014; 2015; Yang *et al.*, 2013]. In many studies, such as matrix completion, the least-square loss function $f(\mathbf{X}) = \sum_{ij\in\Omega}(\mathbf{X}_{ij} - \mathbf{Y}_{ij})^2$ is used [Candés and Recht, 2009; Candés and Plan, 2010; Vandereycken, 2013]. Despite of its popularity, the least-square loss may not perform well when the ratings are discrete values [Srebro *et al.*, 2005].

To deal with rating data, the **maximum margin matrix factorization** (M^3F) is proposed using the hinge loss [Srebro *et al.*, 2005; Rennie and Srebro, 2005; Weimer *et al.*, 2008]. For binary ratings, the objective function can be written as

$$\min_{\mathbf{X}} \;\; f(\mathbf{X}) = \min_{\mathbf{X}} \;\; \sum_{ij\in\Omega} h(\mathbf{Y}_{ij}\mathbf{X}_{ij}), \tag{2}$$

where $h(z) = \max(0, 1 - z)$. The hinge loss $h(z)$ for binary ratings can be easily extended to general ordinal ratings where $\mathbf{Y}_{ij} \in \{1, 2, ...L\}$ by applying $L + 1$ thresholds $\theta_0 \leq \theta_1 \leq ... \leq \theta_L$ learned from data [Rennie and Srebro, 2005]. For the discrete valued rating data, hinge loss would achieve better performance compared to the least square loss.

Problem (1) is known to be NP-hard. Many researchers [Fazel, 2002; Recht *et al.*, 2010] thus propose to solve its nuclear-norm convex relaxation $\min_{\mathbf{X}} \lambda||\mathbf{X}||_* + f(\mathbf{X})$, where $||\mathbf{X}||_*$ denotes the nuclear norm of $\mathbf{X}$ and λ is a regularization parameter. Many convex optimization methods, such as proximal gradient methods [Toh and Yun, 2010; Nie *et al.*, 2012] can be adopted to solve this problem. However, these methods may scale poorly due to the requirement of singular value decompositions (SVDs) of large ranks.

To improve the scalability, some researchers assumes that the rank of $\mathbf{X}$ (i.e. k) is known, and $\mathbf{X}$ can be explicitly factorized as $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n\times k}$ and $\mathbf{V} \in \mathbb{R}^{m\times k}$ [Rennie and Srebro, 2005; Mnih and Salakhutdinov, 2007]. They then solve the following variational formulation instead:

$$\min_{\mathbf{U},\mathbf{V}} \;\; \frac{\lambda}{2}(||\mathbf{U}||_F^2 + ||\mathbf{V}||_F^2) + f(\mathbf{U}\mathbf{V}^\top), \tag{3}$$

where λ is a regularization parameter. Many methods, such as the stochastic gradient descent (SGD), can be used to solve

*The corresponding author.

this problem. However, in real applications, the prior knowledge about k is not likely accessible. Consequently, these algorithms may have to perform expensive model selections to determine k, which is unaffordable in computation [Xu *et al.*, 2012; 2013]. Additionally, since problem (3) is non-convex w.r.t. $\mathbf{U}$ and $\mathbf{V}$ simultaneously, most methods may face the premature convergence problem [Hsieh and Olsen, 2014].

Regarding the scalability issue and the latent factor detection issue of existing methods, we propose an active Riemannian subspace search for M^{3}F (ARSS-M^{3}F). The main contributions of this paper are as follows:

- Leveraging the nonlinear Riemannian conjugate gradient, we propose an efficient block-wise nonlinear Riemannian conjugate gradient (BNRCG) algorithm, which reconstructs $\mathbf{X}$ and learns multiple thresholds $\boldsymbol{\theta}$ in M^{3}F in a joint framework. Compared to existing M3F algorithms, the proposed algorithm is much more efficient.
- Based on BNRCG, we proposed the ARSS-M^{3}F method which applies a simple and efficient pursuit scheme to automatically compute the number of latent factors, which avoids expensive model selections.
- Extensive experiments on both synthetic data sets and real-world data sets demonstrate the superior efficiency and effectiveness of the proposed methods.

2 Related Studies

The M^{3}F problem can be formulated as a semi-definite programming (SDP) problem, thus it can be solved using standard SDP solvers [Srebro *et al.*, 2005]. However, the SDP solver scales very poorly. To improve the scalability, a fast M^{3}F method is proposed to solve problem (3) by investigating the gradient-based optimization method [Rennie and Srebro, 2005]. A low-rank matrix fitting algorithm (LMAFIT) is proposed to solve (3) with the least square loss [Wen *et al.*, 2012]. More recently, a lock-free approach to parallelizing stochastic gradient descent is proposed [Recht *et al.*, 2011]. However, it is nontrivial for them to solve M^{3}F.

Note that the fixed-rank matrices belong to a smooth matrix manifold [Absil *et al.*, 2008; Vandereycken, 2013]. Manifold has been also exploited in a range of applications [Chang *et al.*, 2015; Han *et al.*, 2013; Lu *et al.*, 2013]. Many manifold optimization methods have been proposed to solve (3) [Meyer *et al.*, 2011; Boumal and Absil, 2011; Vandereycken, 2013], such as the Riemannian trust-region method for MC (RTRMC) [Boumal and Absil, 2011], the low-rank geometric conjugate gradient method (LRGeomCG) [Vandereycken, 2013], the quotient geometric matrix completion method (qGeomMC) [Mishra *et al.*, 2012], Grassmannian rank-one update subspace estimation (GROUSE) and the method of scaled gradients on Grassmann manifolds for matrix completion (ScGrassMC) [Ngo and Saad, 2012]. However, all these methods are not applicable to solve M^{3}F.

A number of M^{3}F extensions have been introduced in the last decades [Weimer *et al.*, 2008; 2007; Karatzoglou *et al.*, 2010]. For example, the authors in [Weimer *et al.*, 2007] presented a method using M^{3}F to optimize ranking rather than ratings. Some researcher further improved the performance of M^{3}F by casting it within ensemble approaches [DeCoste, 2006; Wu, 2007].

The importance of automatic latent factor detection (i.e. the model selection problem) has been recognized by many researchers [Xu *et al.*, 2012; 2013; Mnih and Salakhutdinov, 2007]. For example, a probabilistic M^{3}F model is proposed in [Xu *et al.*, 2012; 2013], where the number of latent factors can be inferred from data. However, these methods are usually very expensive as the probabilistic model requires a large amount of computation, which is avoided in our method.

3 M^{3}F on Fixed-rank Manifold

Without loss of generality, we first study M^{3}F where the rank of the rating matrix $\mathbf{X}$ to be recovered is known. We propose the BNRCG method by exploiting the Riemannian geometries to address it.

3.1 Notations

Throughout the paper, we denote by the superscript $\top$ the transpose of a vector/matrix, $\mathbf{0}$ a vector/matrix with all zeros, $\text{diag}(\mathbf{v})$ a diagonal matrix with a vector of diagonal entries equal to $\mathbf{v}$. Let $\mathbf{A} \odot \mathbf{B}$ and $\langle \mathbf{A}, \mathbf{B} \rangle = tr(\mathbf{A}\mathbf{B}^\top)$ represent the element-wise product and inner product of the matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. The singular value decomposition (SVD) of matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ is given by $\mathbf{X} = \mathbf{U}(\text{diag}(\boldsymbol{\sigma}))\mathbf{V}^\top$. Based on the SVD, the **nuclear norm** (or trace-norm) of $\mathbf{X}$ is defined as $\|\mathbf{X}\|_* = \|\boldsymbol{\sigma}\|_1 = \sum_i |\sigma_i|$, and the Frobenius norm of $\mathbf{X}$ is defined as $\|\mathbf{X}\|_F = \|\boldsymbol{\sigma}\|_2$.

3.2 The Proposed Model

In collaborative filtering tasks, the preference scores are often ordinal ratings, where $\mathbf{Y}_{ij} \in \{1, 2, ...L\}$. To generalize the hinge loss for binary case to ordinal ratings, we introduce $L+1$ thresholds $\theta_0 \leq \theta_1 \leq ..., \leq \theta_L$. By default, we have $\theta_0 = -\infty$ and $\theta_L = +\infty$. Therefore, there are $L-1$ free threshold parameters to be determined, namely $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_{L-1}]^\mathsf{T} \in \mathbb{R}^{L-1}$. In a hard-margin case, $\mathbf{X}$ must satisfy the following conditions on observed entries

$$\theta_{\mathbf{Y}_{ij}-1} + 1 \leq \mathbf{X}_{ij} \leq \theta_{\mathbf{Y}_{ij}} - 1.$$

In a soft-margin setting, the hinge loss error for each entry of $\mathbf{X}$ can be written as

$$\boldsymbol{\xi}_{ij} = \sum_{z=1}^{L-1} h(T_{ij}^z \cdot (\theta_z - \mathbf{X}_{ij})), \forall ij \in \Omega, \tag{4}$$

where $T_{ij}^z = \begin{cases} +1 & \text{for } z \geq Y_{ij} \\ -1 & \text{for } z < Y_{ij} \end{cases}$ and $h(z) = \max(0, 1-z)$.

Principally, we propose to reconstruct $\mathbf{X}$ by minimizing the squared hinge loss error

$$\ell(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{2} \sum_{ij \in \Omega} \boldsymbol{\xi}_{ij}^2.$$

Additionally, to prevent from over-fitting, we regularize $\ell(\mathbf{X}, \boldsymbol{\theta})$ by a regularizer $\Upsilon(\mathbf{X}) = \frac{1}{2}(||\mathbf{X}||_F^2 + \nu||\mathbf{X}^\dagger||_F^2)$, where $\mathbf{X}^\dagger$ denotes the pseudo-inverse and $\nu > 0$ is a small scalar (e.g., $\nu = 0.0001$ in this paper by default) and $||\mathbf{X}^\dagger||_F^2$

is a barrier to avoid decreasing of the rank of $\mathbf{X}$ [Vandereycken, 2013]. The M^{3}F problem is formulated as the following optimization problem

$$\min_{\mathbf{X},\boldsymbol{\theta}} f(\mathbf{X},\boldsymbol{\theta}), \text{ s.t. } \text{rank}(\mathbf{X}) = k, \tag{5}$$

where $f(\mathbf{X},\boldsymbol{\theta}) = \lambda\Upsilon(\mathbf{X}) + \ell(\mathbf{X},\boldsymbol{\theta})$ and $0 < \lambda < 1$ denotes the regularization parameter. Note that this regularizer is different from that used in [Vandereycken, 2013], and it is very important for preventing from the over-fitting issue in the context of M^{3}F (see more details in experimental studies).

After addressing problem (5), the prediction can be easily made by

$$\mathbf{Y}^*_{ij} = \max\{z|\mathbf{X}_{ij} \geq \theta_z, z = 1, ..., L\}. \tag{6}$$

Unfortunately, since $f(\mathbf{X},\boldsymbol{\theta})$ is non-convex due to the constraint $\text{rank}(\mathbf{X}) = k$, the optimization of (5) is very difficult. Noting $\mathbf{X}$ is restricted on fixed-rank matrices, we accordingly propose to address it by exploiting the Riemannian geometries on fixed-rank matrices.

3.3 Riemannian Geometry of Fixed-rank Matrices

Suppose $\text{rank}(\mathbf{X}) = r$ with r being known, then $\mathbf{X}$ lies on a smooth manifold of fixed rank-r matrices [Vandereycken, 2013], which is defined as

$$\begin{aligned}\mathcal{M}_r &= \{\mathbf{X} \in \mathbb{R}^{m\times n} : \text{rank}(\mathbf{X}) = r\} \\ &= \{\mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T} : \mathbf{U} \in \text{St}^m_r, \mathbf{V} \in \text{St}^n_r, ||\boldsymbol{\sigma}||_0 = r\}\end{aligned}$$

with $\text{St}^m_r = \{\mathbf{U} \in \mathbb{R}^{m\times r} : \mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}\}$ the Stiefel manifold of $m \times r$ real and orthonormal matrices. The tangent space $T_\mathbf{X}\mathcal{M}_r$ of $\mathcal{M}_r$ at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T} \in \mathbb{R}^{m\times n}$ is given by

$$\begin{aligned}T_\mathbf{X}\mathcal{M}_r = \{&\mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T} : \mathbf{M} \in \mathbb{R}^{r\times r}, \\ &\mathbf{U}_p \in \mathbb{R}^{m\times r}, \mathbf{U}_p^\mathsf{T}\mathbf{U} = \mathbf{0}, \mathbf{V}_p \in \mathbb{R}^{n\times r}, \mathbf{V}_p^\mathsf{T}\mathbf{V} = \mathbf{0}\}.\end{aligned} \tag{7}$$

By defining a metric $g_\mathbf{X}(\mathbf{A},\mathbf{B}) = \langle\mathbf{A},\mathbf{B}\rangle$ on $\mathcal{M}_r$, where $\mathbf{X} \in \mathcal{M}_r$ and $\mathbf{A},\mathbf{B} \in T_\mathbf{X}\mathcal{M}_r$, then $\mathcal{M}_r$ becomes a Riemannian manifold by restricting $\langle\mathbf{A},\mathbf{B}\rangle$ to the *tangent bundle*, which is defined as the disjoint union of all tangent spaces $T\mathcal{M}_r = \bigcup_{\mathbf{X}\in\mathcal{M}_r}\{\mathbf{X}\} \times T_\mathbf{X}\mathcal{M}_r = \{(\mathbf{X},\mathbf{E}) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \mathbf{X} \in \mathcal{M}_r, \mathbf{E} \in T_\mathbf{X}\mathcal{M}_r\}$.

Let $\mathbf{G}$ be the gradient of any smoothing function $f(\mathbf{X})$ in Euclidian space at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$. The Riemannian gradient of $f(\mathbf{X})$ on $\mathcal{M}_r$ is given as the orthogonal projection of $\mathbf{G}$ onto the tangent space at $\mathbf{X}$:

$$\textbf{grad}f(\mathbf{X}) = P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G}). \tag{8}$$

Here $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{Z}) : \mathbf{Z} \mapsto P_U\mathbf{Z}P_V + P_U^\perp\mathbf{Z}P_V + P_U\mathbf{Z}P_V^\perp$ denotes the orthogonal projection of any $\mathbf{Z} \in \mathbb{R}^{m\times n}$ onto the tangent space at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$, where $P_U = \mathbf{U}\mathbf{U}^\mathsf{T}$ and $P_U^\perp = \mathbf{I} - \mathbf{U}\mathbf{U}^\mathsf{T}$ for any $\mathbf{U} \in \text{St}^m_r$.

With prior knowledge about differential geometries on fixed-Rank matrices, we can compute the **Riemannian gradient** of $f(\mathbf{X},\boldsymbol{\theta})$ w.r.t. $\mathbf{X}$ on $\mathcal{M}_r$. Let $\textbf{grad}f(\mathbf{X},\theta)$ denote the **Riemannian gradient**. To compute $\textbf{grad}f(\mathbf{X},\theta)$, we need to calculate the gradient of $f(\mathbf{X},\boldsymbol{\theta})$ on Euclidean space. Firstly, the gradient of $\ell(\mathbf{X},\theta)$ w.r.t. $\mathbf{X}$, denoted by $\widehat{\mathbf{G}}$, can be calculated by

$$\widehat{\mathbf{G}}_{ij} = \frac{\partial\ell(\mathbf{X},\theta)}{\partial\mathbf{X}_{ij}} = \sum_{z=1}^{L-1} T^z_{ij} \cdot h(T^z_{ij} \cdot (\theta_z - \mathbf{X}_{ij})). \tag{9}$$

where $ij \in \Omega$. Note that the gradient of $\Upsilon(\mathbf{X})$ w.r.t. $\mathbf{X}$ is $\mathbf{U}\text{diag}(\boldsymbol{\sigma} - \nu/\boldsymbol{\sigma}^3)\mathbf{V}^\mathsf{T}$ at $\mathbf{X} = \mathbf{U}\text{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$. The gradient of $f(\mathbf{X},\boldsymbol{\theta})$ w.r.t. $\mathbf{X}$ in Euclidian space, denoted by $\mathbf{G}$, can be computed by

$$\mathbf{G} = \widehat{\mathbf{G}} + \lambda\mathbf{U}\text{diag}(\boldsymbol{\sigma} - \nu/\boldsymbol{\sigma}^3)\mathbf{V}^\mathsf{T}. \tag{10}$$

Once $\mathbf{G}$ is computed, $\textbf{grad}f(\mathbf{X},\theta)$ can be calculated according to equation (8). The details of computation can be found in Appendix A.

Finally, the gradient of $f(\mathbf{X},\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$, denoted by $\mathbf{g} = [g_1, g_2, ..., g_{L-1}]^\mathsf{T}$, can be calculated by

$$g_z = \frac{\partial f(\mathbf{X},\boldsymbol{\theta})}{\partial\theta_z} = \sum_{ij\in\Omega} -T^z_{ij} \cdot h(T^z_{ij} \cdot (\theta_z - \mathbf{X}_{ij})), \tag{11}$$

where $z \in \{1, 2, ..., L-1\}$.

3.4 Block-wise Nonlinear Riemannian Conjugate Gradient Descent for M^{3}F

The objective function in (5) involves two types of variables, namely the rating matrix $\mathbf{X} \in \mathcal{M}_r$ and the thresholding parameter $\boldsymbol{\theta} \in \mathbb{R}^{L-1}$. Accordingly, we propose a Block-wise Nonlinear Riemannian Conjugate Gradient (BNRCG) to solve problem (5), which is shown in Algorithm 1. The basic idea is that, at each iteration, we first minimize $f(\mathbf{X},\boldsymbol{\theta})$ w.r.t. $\mathbf{X}$ with fixed $\boldsymbol{\theta}$ by a Nonlinear Riemannian Conjugate Gradient method (Steps 1-3), and then minimize $f(\mathbf{X},\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ with fixed $\mathbf{X}$ by applying a standard gradient descent method (Steps 4-5). We will illustrate Steps 2-5 in details.

Algorithm 1 BNRCG for Fixed-rank M^{3}F.

Given $\text{rank}(\mathbf{X}) = r$. Initialize $\mathbf{X}_1$, $\boldsymbol{\eta}_0$, and $\boldsymbol{\theta}_1$. Let $t = 1$.
1: Compute $\mathbf{E}_t = -\textbf{grad}f(\mathbf{X}_t,\theta_t)$ according to (8).
2: Compute the conjugate direction with PR+ rule: $\boldsymbol{\eta}_t = \mathbf{E}_t + \beta_t\mathcal{T}_{\mathbf{X}_{t-1}\to\mathbf{X}_t}(\boldsymbol{\eta}_{t-1}) \in T\mathcal{M}_r$.
3: Choose a step size α_t and set $\mathbf{X}_{t+1} = R_{\mathbf{X}_t}(\alpha_t\boldsymbol{\eta}_t)$.
4: Compute $\mathbf{g}_t$ according to (11).
5: Choose a step size γ_t and set $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \gamma_t\mathbf{g}_t$.
6: Quit if stopping conditions achieve.
7: Let $t = t + 1$ and go to step 1.

When updating $\mathbf{X}$, different from the classical gradient methods on Euclidean space, the search direction in manifold optimization needs to follow a path on the manifold. Let $\mathbf{X}_t$ be the iteration variable in the BNRCG method on Euclidean space, the search direction $\boldsymbol{\eta}_t$ is calculated by

$$\boldsymbol{\eta}_t = -\text{grad}f(\mathbf{X}_t) + \beta_t\boldsymbol{\eta}_{t-1}, \tag{12}$$

where β_t can be calculated by a Polak-Ribière (PR+) rule [Vandereycken, 2013]:

$$\beta_t = \frac{\text{grad}f(\mathbf{X}_t)^\mathsf{T}(\text{grad}f(\mathbf{X}_t) - \text{grad}f(\mathbf{X}_{t-1}))}{\langle\text{grad}f(\mathbf{X}_{t-1}), \text{grad}f(\mathbf{X}_{t-1})\rangle}. \tag{13}$$

Unfortunately, since $\text{grad}f(\mathbf{X}_t)$, $\text{grad}f(\mathbf{X}_{t-1})$ and $\boldsymbol{\eta}_{t-1}$ are in different tangent spaces $T_{\mathbf{X}_t}\mathcal{M}$ and $T_{\mathbf{X}_{t-1}}\mathcal{M}$, the above two equations are not applicable on Riemannian manifolds. To address this issue, we need two geometric operations, namely, *Retraction* and *Vector Transport*. With the

retraction mapping, one can move points in the direction of a tangent vector and stay on the manifold. In [Vandereycken, 2013], the retraction on $\mathcal{M}_\rho$ can be computed in a closed form by

$$R_{\mathbf{X}}(\mathbf{E}) = P_{\mathcal{M}_\rho}(\mathbf{X}+\mathbf{E}) = \sum_{i=1}^{\rho} \sigma_i \mathbf{p}_i \mathbf{q}_i^\mathsf{T}, \tag{14}$$

where $\sum_{i=1}^{\rho} \sigma_i \mathbf{p}_i \mathbf{q}_i^\mathsf{T}$ denotes the best rank-ρ approximation to $\mathbf{X}+\mathbf{E}$. In addition, the following *Vector Transport* makes the calculations of (12) and (13) meaningful. A vector transport $\mathcal{T}$ on a manifold $\mathcal{M}$ is a smooth map which transports tangent vectors from one tangent space to another. For convenience, let $\mathcal{T}_{\mathbf{X}\to\mathbf{Y}}(\boldsymbol{\eta}_{\mathbf{X}})$ denote the transport from one tangent space $T_{\mathbf{X}}\mathcal{M}$ to another tangent space $T_{\mathbf{Y}}\mathcal{M}$, where $\boldsymbol{\eta}_{\mathbf{X}}$ denotes the tangent vector on $\mathbf{X}$. The step size in the Step 3 and Step 5 is computed by the line search method. When updating $\mathbf{X}_{k+1}$, given a descent direction $\boldsymbol{\eta}_k \in T_{\mathbf{X}_k}\mathcal{M}_r$, the step size α_k is determined such that

$$f(R_{\mathbf{X}_k}(\alpha_k \boldsymbol{\eta}_k)) \le f(\mathbf{X}_k) + c_1 \alpha_k \langle \mathrm{grad} f(\mathbf{X}_k), \boldsymbol{\eta}_k \rangle, \tag{15}$$

where c_1 is the parameter. When updating θ_{k+1} by the standard gradient descent method, the step size γ_t can be computed by the line search on the following condition

$$f(\mathbf{X}_{k+1}, \theta_{k+1}) \le f(\mathbf{X}_{k+1}, \theta_k) + c_2 \gamma_k g_t, \tag{16}$$

where c_2 is the parameter and $0 < c_1 < c_2 < 1/2$.

Lastly, Algorithm 1 is guaranteed to converge to a stationary point of $f(\mathbf{X}, \boldsymbol{\theta})$.

Proposition 1. *The BNRCG algorithm is guaranteed to converge to a stationary point* $(\mathbf{X}^*, \boldsymbol{\theta}^*)$ *of* $f(\mathbf{X}, \boldsymbol{\theta})$ *where* $\boldsymbol{grad} f(\mathbf{X}^*, \boldsymbol{\theta}^*) = \mathbf{0}$ *and* $\nabla_{\boldsymbol{\theta}} f(\mathbf{X}^*, \boldsymbol{\theta}^*) = \mathbf{0}$.

The proof can be found in Appendix B.

3.5 Automatic Latent Factor Detection by Active Subspace Search

Based on BNRCG for fixed-rank M^{3}F, we propose an active subspace search method to detect the number of latent factors automatically presented in Algorithm 2.

Starting from $\mathbf{X} = \mathbf{0}$ where $\boldsymbol{\xi}^0 = \mathbf{b}$, ARSS-M^{3}F iterates with two main steps: to identify the most-active subspace through the worst-case analysis in Step 1, and to find the solution of the fixed-rank M^{3}F problem by BNRCG in step 2. In the following, we present the details of the two main steps.

In the first step, we compute the gradient $\mathbf{G}$ of $f(\mathbf{X}, \theta)$ w.r.t. $\mathbf{X}$ and the active subspace can be found by performing a truncated SVD on $\mathbf{G}$ with the dimensionality of ρ. In the second step, we initialize $\mathbf{X}^k = R_{\mathbf{X}^{k-1}}(-t_{\min}\bar{\mathbf{X}}_\rho)$ where the step size t_{min} is determined by the line search method on the following condition:

$$f(R_{\mathbf{X}^{k-1}}(-t_{\min}\mathbf{G}^{k-1})) \le f(\mathbf{X}^{k-1}) - \frac{t_{\min}}{2}\langle \mathbf{G}^{k-1}, \mathbf{G}^{k-1}\rangle \tag{17}$$

Then, the initialized $\mathbf{X}^k$ is used as the input of the Algorithm 1, namely BNRCG, by which $\mathbf{X}^k$ and θ^k can be updated iteratively. Note that after initializing $\mathbf{X}^k$ in the step 2(a), we increase the estimated rank of BNRCG by ρ. Due to (17), the objective value $f(\mathbf{X}^k)$ monotonically decrease w.r.t. k. Therefore, we stop Algorithm 2 once the following condition is achieved

$$(f(\mathbf{X}^{k-1}) - f(\mathbf{X}^k))/(\rho f(\mathbf{X}^{k-1})) \le \epsilon, \tag{18}$$

where ϵ is a stopping tolerance. In this way, as the algorithm is performed iteratively, we are able to detect the rank of the matrix to be recovered.

Algorithm 2 Active Riemannian Subspace Search for M^{3}F .

Initialize $\mathbf{X}^0 = \mathbf{0}$, $r = 0$, $\boldsymbol{\xi}^0 = \mathbf{b}$ and θ. Let $k = 1$.
1: Find active subspaces as follows:
(a): Compute $\mathbf{G} = \frac{\partial f(\mathbf{X}^k, \theta)}{\partial \mathbf{X}^k}$;
(b): Do thin SVD on $\mathbf{G}$: $[\mathbf{P}_\rho, \Sigma_\rho, \mathbf{Q}_\rho] = \mathrm{SVD}(\mathbf{G}, \rho)$.
2: Let $\bar{\mathbf{X}}_\rho = \mathbf{P}_\rho \Sigma_\rho \mathbf{Q}_\rho^\mathsf{T}$, do master problem optimization:
(a): Find an appropriate step size $t_{\min}$ by (17) and initialize $\mathbf{X}^k = R_{\mathbf{X}^{k-1}}(-t_{\min}\bar{\mathbf{X}}_\rho)$ (Warm Start).
(b): Let $r = r + \rho$ and update $\mathbf{X}^k$ and θ^k by Algorithm 1
3: Quit if stopping conditions are achieved. Let $k = k + 1$ and go to step 1.

4 Empirical Studies

We demonstrate the performance of the proposed methods, namely BNRCG-M^{3}F with fixed-rank problems and ARSS-M^{3}F, by comparing with several related state-of-the-art methods, including FM^{3}F [Rennie and Srebro, 2005], GROUSE [Balzano *et al.*, 2010], LMAFIT [Wen *et al.*, 2012], ScGrassMC [Ngo and Saad, 2012], LRGeomCG [Vandereycken, 2013] and RTRMC [Boumal and Absil, 2011] , on both synthetic and real-world CF tasks. Seven data sets are used in the experiments, including three synthetic data sets and four real-world data sets, Movielens 1M, Movielens 10M [Herlocker *et al.*, 1999], Netflix [Bennett and Lanning, 2007] and Yahoo! Music Track 1 data set [Dror *et al.*, 2012].

The root-mean-square error (RMSE) on both training and testing set will be used as the comparison metric: RMSE $= \sqrt{\sum_{ij\in\Pi}(\mathbf{Y}^*_{ij} - \mathbf{Y}_{ij})^2/|\Pi|}$, where $\mathbf{Y}^*$ denoted the reconstructed ratings according to (6), and $|\Pi|$ denotes number of emblements in the set Π. All the experiments are conducted in Matlab on a work station with an Intel(R) CPU (Xeon(R) E5-2690 v2 @ 3.00GHz) and 256GB memory.

4.1 Synthetic Experiments

In the synthetic experiments where we know the ground-truth, we will demonstrate four points: 1) The sensitivity of the regularization of the proposed M^{3}F methods; 2) The scalability of BNRCG-M^{3}F and ARSS-M^{3}F over other methods; 3) The importance of the squared hinge loss measure over other measures for rating data, e.g., the least square error; 4) The effectiveness of latent factor detection by ARSS-M^{3}F. To demonstrate the above points, we study three synthetic problems of two scales.

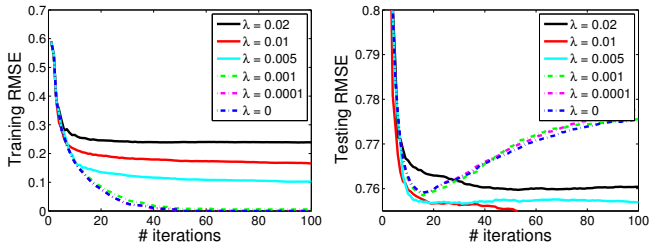
Synthetic Problem

For each of the three synthetic problems, motivated by [Ngo and Saad, 2012; Tan *et al.*, 2014], we first generate a ground-truth low-rank matrix by $\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\mathrm{diag}(\widehat{\boldsymbol{\delta}})\widehat{\mathbf{V}}^\mathsf{T}$, where $\boldsymbol{\delta}$ is a r-sparse vector with each nonzero entry sampled from *Gaussian* distribution $\mathcal{N}(0, 1000)$, $\widehat{\mathbf{U}} \in \mathrm{St}_r^m$ and $\widehat{\mathbf{V}} \in \mathrm{St}_r^n$. In the both two small-scale problems, $\widehat{\mathbf{X}}$ is of size $1,000 \times 1,000$

with $r = 20$, while the large-scale problem $\widehat{\mathbf{X}}$ is of size $20,000 \times 20,000$ with $r = 50$. After sampling the original entries, we respectively produce the binary ratings by $\widehat{\mathbf{Y}}_{ij} = \text{sgn}(\widehat{\mathbf{X}}_{ij})$, and the ordinal ratings $\{1, 2, 3, 4, 5\}$ by projecting the entries of $\widehat{\mathbf{X}}$ into five bins according to their values, which results in a rating matrix $\widehat{\mathbf{Y}}$. Once $\widehat{\mathbf{Y}}$ is generated, we sample $l = r(m + n - r) \times \zeta_{os}$ entries from $\widehat{\mathbf{Y}}$ uniformly to form the observed ratings $\mathbf{Y}$, where ζ_{os} is the oversampling factor [Lin *et al.*, 2010]. In the experiments we set $\zeta_{os} = 3.5$.

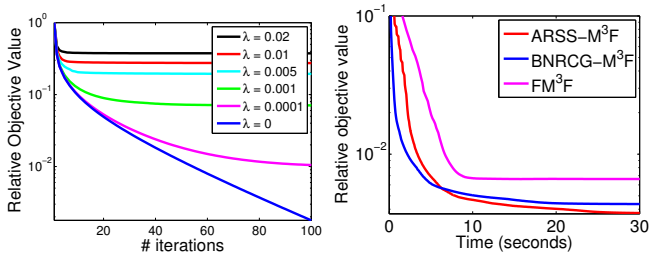
Sensitivity of Regularization Parameter

In this section, to demonstrate the sensitivity of regularization, we perform experiments on the small-scale binary matrix. To illustrate the impact of the regularization in the proposed methods, we test BNRCG-M^{3}F with various regularization parameters λ. Figure 1 reports the training RMSE and testing RMSE. The convergence is shown in Figure 2(a). As can be seen, the regularization is crucial for preventing overfitting.



(a) RMSE on training data (b) RMSE on testing data

Figure 1: RMSE of BNRCG-M^{3}F on binary rating data.



(a) Binary rating data (b) Ordinal rating data

Figure 2: Relative objective values of various methods.

Convergence of M^{3}F on Ordinal Rating Data

In this section, we perform experiments on the small-scale ordinal matrix. We compare the proposed algorithms with the six baseline methods and collect the convergence behavior of the three M^{3}F methods. The ground-truth rank is used as the estimated rank for all methods excluding ARSS-M^{3}F.

The convergence behavior of our methods and FM^{3}F is illustrated in Figure 2(b), which shows that our methods can converge better and faster. Table 2 reports the resultant RMSE on the testing set and the computational time of each method on the small-scale synthetic ordinal rating data set.

Scalability of M^{3}F on Ordinal Rating Data

In this section, we perform experiments on the large-scale ordinal matrix. We compare our methods with the 5 baseline algorithms. We use the ground-truth rank as the estimated rank for all methods except ARSS-M^{3}F. The average estimated rank of ARSS-M^{3}F is 42, which is close to the groundtruth rank of 50. According to the estimated rank in the two synthetic datasets, the latent factor detection of ARSS-M^3 is effective. The RMSE on the testing set and computational time of each algorithm are listed in Table 2.

Table 1: Statistics of the Real-world Data Sets.

Data Sets	# users	# items	# ratings
Movielens 1M	6,040	3,952	1,000,209
Movielens 10M	71,567	10,681	10,000,054
Netflix	480,189	17,770	100,480,507
Yahoo! Music Track 1	1,000,990	624,961	262,810,175

4.2 Real-world Experiments

In real-world data experiments, to demonstrate the significance of the hinge loss to the rating data and effectiveness of latent factor estimation of our method, we study four real-world large scale data sets, namely Movielens 1M, Movielens 10M data set, Netflix data set and Yahoo! Music Track 1 data set. The baseline methods include FM^{3}F, GROUSE, LMAFIT, ScGrassMC, LRGeomCG and RTRMC.

Table 1 lists the size statistics of the four data sets. The vast majority (99.71%) of ratings in Yahoo! Music Track 1 are multiples of ten. For convenience, we only consider these ratings. For Movielen 10M and Yahoo! Music Track 1, we map the ratings to ordinal integer values before the experiment. For each data set, we sample 80% of data into the training set and the rest into the testing set.

Table 2 reports the computational time of all comparison methods and testing RMSE on the four data sets. According to the resultant RMSE, compared to other loss measure, i.e. least square loss, our method can recover the matrix with lower error. Note that in all experiments in both synthetic and real-world data, no model selection cost is included for all comparison methods. If model selections are considered, the comparison methods will cost much more time. Some results for GROUSE and M^{3}F are not available due to their high computation cost. From the table, ARSS-M^{3}F and BNRCG-M^{3}F recover the rating matrix efficiently and outperform other comparison methods in terms of RMSE on the four real-world data sets. It is worth mentioning that though LRGeomCG shows faster speed on Yahoo data set, it achieves much worse RMSE than M^{3}F based methods.

5 Conclusion

To deal with the ordinal discrete ratings in recommendation systems, M^{3}F is proposed. However, existing M^{3}F methods is faced with the scalability and latent factor detection issues. To address the two challenges, we present ARSS-M^{3}F, a scalable M^{3}F method based on active Riemannian subspace search. Specifically, the proposed algorithm first treat the M^{3}F problem as the fixed number of latent factors

Table 2: Experimental results on synthetic and real-world data sets. Computational time is recorded in seconds.

Methods	Small Synthetic*		Large Synthetic*		Movielens 1M†		Movielens 10M†		Netflix†		Yahoo Music†	
	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
FM^{3}F [Rennie and Srebro, 2005]	0.3811	11.99	0.3899	2186	0.9344	212.2051	0.9143	13001	1.0971	65662	-	-
GROUSE [Balzano *et al.*, 2010]	0.4718	27.84	0.512	11214	0.9225	39.4184	0.8653	3853	-	-	-	-
LMAFIT [Wen *et al.*, 2012]	0.4701	6.08	0.4973	827	0.9373	19.9465	0.8424	832	0.9221	4374	24.222	24349
ScGrassMC [Ngo and Saad, 2012]	0.4638	10.19	0.4714	2149	0.9372	21.3109	0.8427	917	0.9192	5787	24.7982	37705
LRGeomCG [Vandereycken, 2013]	0.4679	6.01	0.4904	814	0.9321	10.2484	0.849	312	0.9015	3151	25.2279	8666
RTRMC [Boumal and Absil, 2011]	0.4676	8.68	0.4715	884	0.9311	14.1038	0.846	673	0.9102	6465	24.5971	32592
BNRCG-M^{3}F	0.3698	5.34	0.3915	635	0.9285	13.4437	0.8437	714	0.9022	4118	23.8573	24631
ARSS-M^{3}F	**0.3693**	5.33	**0.3684**	542	**0.9222**	9.5482	**0.8411**	650	**0.9001**	3583	**23.7902**	22065

* No cost of model selections is included for all fix-rank methods as the ground-truth rank is available.
† The rank detected by ARSS-M^{3}F is used as the estimated rank for other methods. Thus no model selection is considered. The average ranks estimated by ARSS-M^{3}F on Movielens 1M, Movielens 10M, Netflix and Yahoo Music are 8, 14, 16 and 28 respectively.

and solve it using BNRCG. In the meantime, a simple and efficient active subspace search approach is applied to automatically compute the number of latent factors. Experiments on both synthetic and real-world data demonstrate that the proposed method can provide competitive performance.

Acknowledgement

This work was in part supported by the 973 Program (2015CB352300), in part supported by the Australian Research Council Future Fellowship FT130100746, the Data to Decisions Cooperative Research Centre, Australia, and the ARC Grant DP140102270 and DE130101311.

A Appendix A: Computation of $\mathbf{grad} f(\mathbf{X}, \boldsymbol{\theta})$

According to [Vandereycken, 2013], a tangent vector $\boldsymbol{\eta} \in \mathcal{T}\mathcal{M}_r$ is represented as $\boldsymbol{\eta} = \mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T}$ (see equation (7) for details). By definition, the Riemannian gradient of $f(\mathbf{X}, \boldsymbol{\theta})$ w.r.t. $\mathbf{X}$, denoted by $\mathbf{grad} f(\mathbf{X}, \boldsymbol{\theta})$, at $\mathbf{X} = \mathbf{U}\mathrm{diag}(\boldsymbol{\sigma})\mathbf{V}^\mathsf{T}$ can be calculated by $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G})$, where $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{Z}) = P_U\mathbf{Z}P_V + P_U^\perp\mathbf{Z}P_V + P_U\mathbf{Z}P_V^\perp$ is the projection of $\mathbf{G}$ onto the tangent space $\mathcal{T}\mathcal{M}_r$. Let $\Xi = \lambda\mathrm{diag}(\boldsymbol{\sigma} - \nu/\boldsymbol{\sigma}^3)$. For convenience, we first present the computation of $\mathbf{grad} f(\mathbf{X}, \boldsymbol{\theta})$ in Algorithm 3.

Lemma 1. *Suppose $\mathbf{U}_p$, $\mathbf{V}_p$, and $\mathbf{M}$ are obtained from Algorithm 3, then $\mathbf{grad} f(\mathbf{X}, \boldsymbol{\theta}) = \mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T}$.*

Algorithm 3 Compute Riemannian gradient $\mathbf{grad} f(\mathbf{X})$.

1: Let $\Xi = \lambda\mathrm{diag}(\boldsymbol{\sigma} - \nu/\boldsymbol{\sigma}^3)$, and compute $\widehat{\mathbf{G}}$ via (9).
2: Compute $\mathbf{G}_u = \widehat{\mathbf{G}}^\mathsf{T}\mathbf{U}$, and $\mathbf{G}_v = \widehat{\mathbf{G}}\mathbf{V}$.
3: Compute $\widehat{\mathbf{M}} = \mathbf{U}^\mathsf{T}\mathbf{G}_v$.
4: Compute $\mathbf{U}_p = \mathbf{G}_v - \mathbf{U}\widehat{\mathbf{M}}$, and $\mathbf{V}_p = \mathbf{G}_u - \mathbf{V}\widehat{\mathbf{M}}^\mathsf{T}$.
6: Update $\mathbf{M} = \widehat{\mathbf{M}} + \Xi$.
6: Output $\mathbf{U}_p$, $\mathbf{V}_p$, and $\mathbf{M}$, and $\mathbf{grad} f(\mathbf{X}, \boldsymbol{\theta}) = \mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T}$.

Proof. To verify the validity of Algorithm 3, we just need to show that, $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G}) = \mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T}$.

Notice that, $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}$ and $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$. On one hand, we have $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G}) = P_{T_\mathbf{X}\mathcal{M}_r}(\widehat{\mathbf{G}} + \mathbf{U}\Xi\mathbf{V}^\mathsf{T}) = \widehat{\mathbf{G}}P_V + P_U\widehat{\mathbf{G}} - P_U\widehat{\mathbf{G}}P_V + \mathbf{U}\Xi\mathbf{V}^\mathsf{T}$. On the other hand, according to Algorithm 3, we have $\boldsymbol{\eta} = \mathbf{U}\mathbf{M}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T} = \mathbf{U}\widehat{\mathbf{M}}\mathbf{V}^\mathsf{T} + \mathbf{U}_p\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{V}_p^\mathsf{T} + \mathbf{U}\Xi\mathbf{V}^\mathsf{T} = \widehat{\mathbf{G}}\mathbf{V}\mathbf{V}^\mathsf{T} + \mathbf{U}\mathbf{U}^\mathsf{T}\widehat{\mathbf{G}} - \mathbf{U}\widehat{\mathbf{M}}\mathbf{V}^\mathsf{T} + \mathbf{U}\Xi\mathbf{V}^\mathsf{T}$, which actually equals to $P_{T_\mathbf{X}\mathcal{M}_r}(\mathbf{G})$. This completes the proof. □

B Appendix B: Proof of Proposition 1

The proof parallels the proof in [Vandereycken, 2013]. Notice that, the optimization on $\boldsymbol{\theta}$ is conducted in Euclidian space $\mathbb{R}^{L-1}$. Moreover, $\{\boldsymbol{\theta}_t\}$ is bounded; otherwise $\ell(\mathbf{X}, \boldsymbol{\theta})$ will go to infinity according to (4). Without loss of generality, suppose $\boldsymbol{\theta}_t \in [-l, l]^{L-1}$, where $l > 0$ is a finite number. Following [Vandereycken, 2013], we can also show that $\{\mathbf{X}_t\}$ stay in a closed and bounded subset of $\mathcal{M}_r$.

Let $\Psi = \{\mathbf{X} \in \mathcal{M}_r, f(\mathbf{X}, \boldsymbol{\theta}) \leq f(\mathbf{X}_0, \boldsymbol{\theta}_0)\}$ be the level set at $(\mathbf{X}_0, \boldsymbol{\theta}_0)$. Due to the line search, we have $\ell(\mathbf{X}_t, \boldsymbol{\theta}_t) + \frac{\lambda}{2}(||\mathbf{X}_t||_F^2 + \nu||\mathbf{X}_t^\dagger||_F^2) \leq f(\mathbf{X}_0, \boldsymbol{\theta}_0)$. Therefore, we have $\frac{\lambda}{2}||\mathbf{X}_t||_F^2 \leq f(\mathbf{X}_0, \boldsymbol{\theta}_0)$, which implies $\sigma_1 = \sqrt{||\mathbf{X}_t||_F^2} \leq \sqrt{2f(\mathbf{X}_0, \boldsymbol{\theta}_0)/\lambda}$. Here, σ_1 denotes the largest singular value of $\mathbf{X}_t$. Similarly, we have $\frac{\nu\lambda}{2}||\mathbf{X}_t^\dagger||_F^2 = \sum_{i=1}^r \frac{\nu\lambda}{2\sigma_i^2} \leq f(\mathbf{X}_0, \boldsymbol{\theta}_0)$, which implies that $\frac{\nu\lambda}{2\sigma_i^2} \leq f(\mathbf{X}_0, \boldsymbol{\theta}_0), \forall i \in \{1, ..., r\}$. This further implies that $\sigma_r \geq \sqrt{\nu\lambda/2f(\mathbf{X}_0, \boldsymbol{\theta}_0)}$, where σ_r is the least singular value of $\mathbf{X}_t$.

Clearly, all $\mathbf{X}_t$ stay inside the set $\mathcal{S} = \{\mathbf{X} \in \mathcal{M}_r : \sigma_1 \leq \sqrt{2f(\mathbf{X}_0, \boldsymbol{\theta}_0)/\lambda}, \sigma_r \geq \sqrt{\nu\lambda/2f(\mathbf{X}_0, \boldsymbol{\theta}_0)}\}$, which is closed and bounded, hence compact.

Now we complete the proof by contradiction. Without loss of generality, suppose $\lim_{t\to\infty} ||\mathbf{grad} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_F + ||\nabla_{\boldsymbol{\theta}_t} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_2 \neq \mathbf{0}$, then there exists an $\epsilon > 0$, and a subsequence in $\{(\mathbf{X}_t, \boldsymbol{\theta}_t)\}_{t\in\Gamma}$ such that $||\mathbf{grad} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_F + ||\nabla_{\boldsymbol{\theta}} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_2 \geq \epsilon > 0$ for all $t \in \Gamma$. Since $\mathbf{X}_t \in \mathcal{S}$ and $\boldsymbol{\theta}_t$ is constrained in $[-l, l]^{L-1}$, the subsequence $\{(\mathbf{X}_t, \boldsymbol{\theta}_t)\}_{t\in\Gamma}$ should have a limit point $(\mathbf{X}^*, \boldsymbol{\theta}^*)$ in $\mathcal{S} \times [-l, l]^{L-1}$. By continuity of $\mathbf{grad} f(\mathbf{X}, \boldsymbol{\theta})$ and $\nabla_{\boldsymbol{\theta}} f(\mathbf{X}, \boldsymbol{\theta})$ (which can be easily verified for squared hinge loss), this implies that $||\mathbf{grad} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_F \geq \epsilon$ which contradicts Theorem 4.3.1 in [Absil *et al.*, 2008] that every accumulation point is a critical point of $f(\mathbf{X}, \boldsymbol{\theta})$. We therefore conclude that $\lim_{t\to\infty} ||\mathbf{grad} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_F = \mathbf{0}$ and $\lim_{t\to\infty} ||\nabla_{\boldsymbol{\theta}} f(\mathbf{X}_t, \boldsymbol{\theta}_t)||_2 = \mathbf{0}$.

References

[Absil *et al.*, 2008] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2008.

[Balzano *et al.*, 2010] L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Proceedings of Allerton*, 2010.

[Bennett and Lanning, 2007] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*. ACM, 2007.

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

[Candés and Plan, 2010] E. J. Candés and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[Candés and Recht, 2009] E. J. Candés and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9:717–772, 2009.

[Chang *et al.*, 2015] Xiaojun Chang, Feiping Nie, Zhigang Ma, Yi Yang, and Xiaofang Zhou. A convex formulation for spectral shrunk clustering. In *AAAI*, 2015.

[DeCoste, 2006] Dennis DeCoste. Collaborative prediction using ensembles of maximum margin matrix factorizations. In *ICML*. ACM, 2006.

[Dror *et al.*, 2012] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The Yahoo! Music Dataset and KDD-Cup'11. *JMLR Workshop and Conference Proceedings*, 18:3–18, 2012.

[Fazel, 2002] M. Fazel. Matrix rank minimization with applications. 2002. PhD thesis, Stanford University.

[Han *et al.*, 2013] Yahong Han, Zhongwen Xu, Zhigang Ma, and Zi Huang. Image classification with manifold learning for out-of-sample data. *Signal Processing*, 93(8):2169–2177, 2013.

[Herlocker *et al.*, 1999] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. In *SIGIR*. ACM, 1999.

[Hsieh and Olsen, 2014] Cho-Jui Hsieh and Peder Olsen. Nuclear norm minimization via active subspace selection. In *ICML*, 2014.

[Huang *et al.*, 2013] Jin Huang, Feiping Nie, and Heng Huang. Robust discrete matrix completion. In *AAAI*, 2013.

[Karatzoglou *et al.*, 2010] Alexandros Karatzoglou, Markus Weimer, and Alex J Smola. Collaborative filtering on a budget. In *AISTATS*, 2010.

[Lin *et al.*, 2010] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC, 2010.

[Lu *et al.*, 2013] Xinyan Lu, Fei Wu, Siliang Tang, Zhongfei Zhang, Xiaofei He, and Yueting Zhuang. A low rank structural large margin method for cross-modal ranking. In *SIGIR*, 2013.

[Meyer *et al.*, 2011] G. Meyer, S. Bonnabel, and R. Sepulchre. Linear regression under fixed-rank constraints: A riemannian approach. In *ICML*, 2011.

[Mishra *et al.*, 2012] B. Mishra, K. A. Apuroop, and R. Sepulchre. A riemannian geometry for low-rank matrix completion. Technical report, 2012.

[Mnih and Salakhutdinov, 2007] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *NIPS*, 2007.

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

[Nie *et al.*, 2012] Feiping Nie, Heng Huang, and Chris Ding. Low-rank matrix recovery via efficient schatten p-norm minimization. In *AAAI*, 2012.

[Recht *et al.*, 2010] B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum rank solutions to linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3), 2010.

[Recht *et al.*, 2011] Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *NIPS*, 2011.

[Rennie and Srebro, 2005] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *ICML*, 2005.

[Srebro *et al.*, 2005] Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakola. Maximum-margin matrix factorization. In *NIPS*. MIT Press, 2005.

[Su and Khoshgoftaar, 2009] Xiaoyuan Su and Taghi M. Khoshgoftaar. A survey of collaborative filtering techniques. *Adv. in Artif. Intell.*, 2009:4, 2009.

[Tan *et al.*, 2014] Mingkui Tan, Ivor W. Tsang, Li Wang, Bart Vandereycken, and Sinno Jialin Pan. Riemannian pursuit for big matrix recovery. *ICML*, 2014.

[Toh and Yun, 2010] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6:615–640, 2010.

[Vandereycken, 2013] Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

[Weimer *et al.*, 2007] Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. Maximum margin matrix factorization for collaborative ranking. *NIPS*, 2007.

[Weimer *et al.*, 2008] Markus Weimer, Alexandros Karatzoglou, and Alex Smola. Improving maximum margin matrix factorization. *Machine Learning*, 72(3):263–276, 2008.

[Wen *et al.*, 2012] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a non-linear successive over-relaxation algorithm. *Math. Program. Comput.*, 4(4):333–361, 2012.

[Wu, 2007] Mingrui Wu. Collaborative filtering via ensembles of matrix factorizations. In *Proceedings of KDD Cup and Workshop*, 2007.

[Xiao *et al.*, 2014] Shijie Xiao, Mingkui Tan, and Dong Xu. Weighted block-sparse low rank representation for face clustering in videos. In *ECCV*, 2014.

[Xiao *et al.*, 2015] Shijie Xiao, Wen Li, Dong Xu, and Dacheng Tao. FaLRR: A Fast Low Rank Representation Solver. In *CVPR*, 2015.

[Xu *et al.*, 2012] Minjie Xu, Jun Zhu, and Bo Zhang. Nonparametric max-margin matrix factorization for collaborative prediction. In *NIPS*, 2012.

[Xu *et al.*, 2013] Minjie Xu, Jun Zhu, and Bo Zhang. Fast max-margin matrix factorization with data augmentation. In *ICML*, 2013.

[Yang *et al.*, 2013] Yi Yang, Zhigang Ma, A.G. Hauptmann, and N. Sebe. Feature selection for multimedia analysis by sharing information among multiple tasks. *Multimedia, IEEE Transactions on*, 15(3):661–669, April 2013.