Tensor Completion with Side Information: A Riemannian Manifold Approach

Tengfei Zhou, Hui Qian*, Zebang Shen, Chao Zhang, Congfu Xu
College of Computer Science and Technology, Zhejiang University, China
{zhoutengfei,qianhui,shenzebang,zczju,xucongfu}@zju.edu.cn

Abstract

By restricting the iterate on a nonlinear manifold, the recently proposed Riemannian optimization methods prove to be both efficient and effective in low rank tensor completion problems. However, existing methods fail to exploit the easily accessible side information, due to their format mismatch. Consequently, there is still room for improvement. To fill the gap, in this paper, a novel Riemannian model is proposed to tightly integrate the original model and the side information by overcoming their inconsistency. For this model, an efficient Riemannian conjugate gradient descent solver is devised based on a new metric that captures the curvature of the objective. Numerical experiments suggest that our method is more accurate than the state-of-the-art without compromising the efficiency.

1 Introduction

Low Rank Tensor Completion (LRTC) problem, which aims to recover a *tensor* from its linear measurements, arises naturally in many artificial intelligence applications. In hyperspectral image inpainting, LRTC is applied to interpolate the unknown pixels based on the partial observation [Xu *et al.*, 2015]. In recommendation tasks, LRTC helps users find interesting items [Liu *et al.*, 2015]. In computational phenotyping, one adopts LRTC to discovery phenotypes in heterogeneous electronic health records [Wang *et al.*, 2015].

Euclidean Models: LRTC can be formulated by a variety of optimization models over the *Euclidean* space. Amongst them, convex models that encapsulate LRTC as a regression problem penalized by a tensor nuclear norm are the most popular and well-understood [Romera-Paredes and Pontil, 2013; Zhang *et al.*, 2014]. Though most of them have sound theoretical guarantees [Zhang and Aeron, 2016; Chen *et al.*, 2013; Yuan and Zhang, 2015], in general, their solvers are ill-suited for large tensors because these procedures involve Singular Value Decomposition (SVD) of huge matrices per iteration [Liu *et al.*, 2013]. Another class of Euclidean models is formulated as the decomposition problem that factorizes a low rank tensor into small factors [Jain and Oh, 2014; Filipović and Jukić, 2015; Xu *et al.*, 2015]. Many solvers for such decomposition based model have been proposed, and low per-iteration computational cost is illustrated [Beutel *et al.*, 2014; Liu *et al.*, 2014; Smith *et al.*, 2016].

Riemannian Models: LRTC can also be modeled by optimization constrained on *Riemannian* manifolds [Kressner *et al.*, 2014; Kasai and Mishra, 2016], which is easily handled by many manifold based solvers [Absil *et al.*, 2009]. Empirical comparison has shown that Riemannian solvers use significantly less CPU time to recover the underlying tensor in contrast to the Euclidean solvers [Kasai and Mishra, 2016]. The main reason resides in that they avoid SVD of huge matrices by explicitly exploiting the geometrical structure of LRTC, which makes them more suitable for massive problem.

Of all the Riemanian models, two search spaces, fix multilinear rank manifold [Kressner *et al.*, 2014] and Tucker manifold [Kasai and Mishra, 2016], are usually employed. The former is a sub-manifold of Euclidean space, and the latter is a quotient manifold induced by the *Tucker decomposition*. Generally, quotient manifold based solvers have higher convergence rates because it is usually easier to design a pre-conditioner for them [Kasai and Mishra, 2016; Mishra and Sepulchre, 2016].

Side Information: In the Euclidean models of LRTC, *side information* has been proved to be helpful in improving the accuracy [Narita *et al.*, 2011; Acar *et al.*, 2011; Beutel *et al.*, 2014]. One common form of the side information is the *feature matrix*, which measures the statistical properties of tensor modes [Kolda and Bader, 2009]. For example, in Netflix tasks, feature matrix can be built from the demography of users [Bell and Koren, 2007]. Another form is the *similarity matrix*, which quantifies the resemblance between two entities of a tensor mode. For instance, the social network generates the similarity matrix by utilizing the correspondence between users [Rai *et al.*, 2015]. In practice, these two matrices can be transformed to each other, and we only consider the feature matrix case.

However, as far as we know, side information has not been incorporated in any Riemannian model for LRTC. The first difficulty lies in the model design. Fusing the side information into the Riemannian model inevitably compromises the integrity of the low rank tensor due to the compactness of the manifold. The second difficulty results from the solver design. Incorporating the side information may aggravate the

*Corresponding author

ill-conditioning of LRTC problem and degenerates the convergence significantly.

Contributions: To address these difficulties, a novel Riemannian LRTC method is proposed from the perspective of both model and solver designs. By exploring the relation between the subspace spanned by the tensor fibers and the column space of the feature matrix, we explicitly integrate the side information in a compact way. Meanwhile, a first order solver is devised under the manifold optimization framework. To ease the ill-conditioning, we design a novel metric based on an approximated Hessian of the cost function. The metric implicitly induce an adaptive preconditioner for our solver. Empirical studies illustrate that our method achieves much more accurate solutions within comparable processing time than the state-of-the-art.

2 Notations and Preliminaries

In this paper, we only focus on the 3rd order tensor, but generalizing our method to higher order is straight forward. We use the notation $\mathbf{X} \in \mathbb{R}^{n\times m}$ to denote a matrix, and the notation $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ to denote a d-th order tensor. We also denote by $\boldsymbol{\mathcal{X}}(i_1,\cdots,i_d)$ the element in position $(i_1,\cdots,i_d)$ of $\boldsymbol{\mathcal{X}}$. For many cases, we use abbreviation like $\{\mathbf{O}_i\}_{i=1}^3$ to denote the sequence $\mathbf{O}_1,\mathbf{O}_2,\mathbf{O}_3$.

Mode-k Fiber and matricization: A fiber of a tensor is obtained by varying one index while fixing the others, i.e. $\boldsymbol{\mathcal{X}}(i_1,\cdots,i_{k-1},:,i_{k+1},\cdots,i_d)$ is the mode-k fiber of a d-th order tensor $\boldsymbol{\mathcal{X}}$. A *mode-k matricization* $\boldsymbol{\mathcal{X}}_{(k)} \in \mathbb{R}^{n_k\times(n_1\cdots n_{k-1}n_{k+1}\cdots n_d)}$ of a tensor $\boldsymbol{\mathcal{X}}$ is obtained by arranging the mode-n fibers of $\boldsymbol{\mathcal{X}}$ so that each of them is a column of $\boldsymbol{\mathcal{X}}_{(k)}$ [Kolda and Bader, 2009] The mode-k product of tensor $\boldsymbol{\mathcal{X}}$ and matrix $\mathbf{A}$ is denoted by $\boldsymbol{\mathcal{X}}\times_k\mathbf{A}$, whose mode-$k$ matricization can be expressed as $(\boldsymbol{\mathcal{X}}\times_k\mathbf{A})_{(k)} = \mathbf{A}\boldsymbol{\mathcal{X}}_{(k)}$.

Inner product and norm: The inner product of two tensors with the same size is defined by $\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Y}}\rangle = \sum_{i_1,\cdots,i_d}\boldsymbol{\mathcal{X}}(i_1,\cdots,i_d)\boldsymbol{\mathcal{Y}}(i_1,\cdots,i_d)$. The Frobenius norm of a tensor $\boldsymbol{\mathcal{X}}$ is defined by $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{X}}\rangle}$.

Multi-linear rank and Tucker decomposition: The multi-linear rank $\mathrm{rank}^{\mathrm{vec}}(\boldsymbol{\mathcal{X}})$ of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is defined as a vector $(\mathrm{rank}(\boldsymbol{\mathcal{X}}_{(1)}),\mathrm{rank}(\boldsymbol{\mathcal{X}}_{(2)}),\mathrm{rank}(\boldsymbol{\mathcal{X}}_{(3)}))$. If $\mathrm{rank}^{\mathrm{vec}}(\boldsymbol{\mathcal{X}}) = (r_1,r_2,r_3)$, tucker decomposition factorizes $\boldsymbol{\mathcal{X}}$ into a small core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1\times r_2\times r_3}$ and three matrices $\mathbf{U}_i \in \mathbb{R}^{n_i\times r_i}$ with orthogonal columns, that is $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}}\times_{i=1}^3\mathbf{U}_i$. Note that, the tucker decomposition of a tensor is not unique. In fact, if $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}}\times_{i=1}^3\mathbf{U}_i$, we can easily obtain $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{H}}\times_{i=1}^3\mathbf{V}_i$, with $\boldsymbol{\mathcal{H}} = \boldsymbol{\mathcal{G}}\times_{i=1}^3\mathbf{O}_i^\top$, $\mathbf{V}_i = \mathbf{U}_i\mathbf{O}_i$, where $\mathbf{O}_i \in \mathbb{R}^{r_i\times r_i}$ is any orthogonal matrix. Thus, we obtain the equivalent class

$$[\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3] \triangleq \{(\boldsymbol{\mathcal{G}}\times_{i=1}^3\mathbf{O}_i^\top,\{\mathbf{U}_i\mathbf{O}_i\}_{i=1}^3)|\mathbf{O}_i^\top\mathbf{O}_i = \mathbf{I}_i\}.$$

We denote $[\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3]$ by $[\boldsymbol{\mathcal{X}}]$, when $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}}\times_{i=1}^3\mathbf{U}_i$. Usually, $[\boldsymbol{\mathcal{X}}]$ is called the *Tucker representation* of $\boldsymbol{\mathcal{X}}$, while $\boldsymbol{\mathcal{X}}$ is call the *tensor representation* of $[\boldsymbol{\mathcal{X}}]$. We also use $\overline{\boldsymbol{\mathcal{X}}}$ to denote a specific decomposition of $\boldsymbol{\mathcal{X}}$, additionally $\overline{\boldsymbol{\mathcal{X}}} \in [\boldsymbol{\mathcal{X}}]$.

2.1 Search Space of Riemannian Models

The Tucker manifold that we used in our Riemannian model is a quotient manifold induced by the Tucker decomposition. In order to lay the ground for Tucker manifold, we first describe its counterpart, the fix multi-rank manifold, which will be helpful in understanding the whole derivation.

A fixed multi-linear rank manifold $\mathcal{F}_r$ consists of tensors with the same fixed multi-linear rank. Specifically

$$\mathcal{F}_r = \{\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times n_2\times n_3}|\,\mathrm{rank}^{\mathrm{vec}}(\boldsymbol{\mathcal{X}}) = r\}.$$

To define the Tucker manifold, we first define a total space

$$\mathcal{M}_r = \mathbb{R}^{r_1\times r_2\times r_3}\times\mathrm{S}(r_1,n_1)\times\mathrm{S}(r_2,n_2)\times\mathrm{S}(r_3,n_3), \quad (1)$$

in which $\mathrm{S}(r_i,n_i)$ is the *Stiefel manifold* of $n_i\times r_i$ matrices with orthogonal columns. Then, we can depict the Tucker manifold of multi-linear rank r as follows.

$$\mathcal{M}_r/\sim \triangleq \left\{[\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3]|(\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3) \in \mathcal{M}_r\right\}. \quad (2)$$

The Tucker manifold is a quotient manifold of the total space (1). We use the abstract quotient manifold, rather than the concrete total space, as search space because the non-uniqueness of the Tucker decomposition is undesirable for optimization. Note that such non-uniqueness will introduce more local optima into the minimization. The relation of manifold $\mathcal{F}_r$ and $\mathcal{M}_r/\sim$ is characterized as follows.

Proposition 1. *The quotient manifold $\mathcal{M}_r/\sim$ is diffeomorphic to the fix multi-linear rank manifold $\mathcal{F}_r$, with diffeomorphism $\rho(\cdot)$ from $\mathcal{F}_r$ to $\mathcal{M}_r/\sim$ defined by $\rho(\boldsymbol{\mathcal{X}}) = [\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3]$ where $[\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3]$ is the tucker representation of $\boldsymbol{\mathcal{X}}$.*

This proposition says that each tensor $\boldsymbol{\mathcal{X}} \in \mathcal{F}_r$ can be represented by a unique equivalent class $[\boldsymbol{\mathcal{G}},\{\mathbf{U}_i\}_{i=1}^3] \in \mathcal{M}/\sim$ and vice-versa.

2.2 Vanilla Riemannian Tensor Completion

The purest incarnation of Riemannian tensor completion model is the Riemannian model over the fix multi-linear rank manifold. Let $\boldsymbol{\mathcal{R}} \in \mathbb{R}^{n_1\times n_2\times n_3}$ be a partially observed tensor. Let Ω be the set which contains the indices of observed entries. The model can be expressed as:

$$\min_{\boldsymbol{\mathcal{X}}}\frac{1}{2}\|\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}-\boldsymbol{\mathcal{R}})\|_F^2 \quad \text{s.t.} \quad \boldsymbol{\mathcal{X}} \in \mathcal{F}_r, \quad (3)$$

with $\mathcal{P}_\Omega$ maps $\boldsymbol{\mathcal{X}}$ to the sparsified tensor $\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}})$, where $\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}})(i_1,i_2,i_3) = \boldsymbol{\mathcal{X}}(i_1,i_2,i_3)$ if $(i_1,i_2,i_3) \in \Omega$, and $\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}})(i_1,i_2,i_3) = 0$ otherwise.

Another popular model, Tucker model, is based on the quotient manifold $\mathcal{M}_r/\sim$, which can be expressed as:

$$\min_{\boldsymbol{\mathcal{X}}}\frac{1}{2}\|\mathcal{P}_\Omega(\rho^{-1}([\boldsymbol{\mathcal{X}}])-\boldsymbol{\mathcal{R}})\|_F^2 \quad \text{s.t.} \quad [\boldsymbol{\mathcal{X}}] \in \mathcal{M}_r/\sim, \quad (4)$$

with ρ defined in Prop. 1.

Note that since the dawn of Riemannian framework for LRTC, a quandary exists: on one hand, sparse measurement limits the capacity of the solution; on the other hand, rich side information can not be incorporated into this framework. In many artificial intelligence applications, demands for high accuracy further exacerbates such dilemma.

3 Riemannian Model with Side Information

We focus on the case that the side information is encoded in feature matrices $\mathbf{P}_i \in \mathbb{R}^{n_i \times k_i}$. Suppose $\boldsymbol{\mathcal{R}} \in \mathcal{F}_r$ has tucker factors $(\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3)$.Without loss of generality, we assume that $k_i \geq r_i$ and $\mathbf{P}_i$ has orthogonal columns.

In the ideal case, we assume that

$$\operatorname{span}(\mathbf{U}_i) \subset \operatorname{span}(\mathbf{P}_i). \tag{5}$$

Such relation means that the feature matrices contain all the information in the latent space of the underlying tensor. Equivalently, there exists a matrix $\mathbf{W}_i$ such that $\mathbf{U}_i = \mathbf{P}_i\mathbf{W}_i$. However, in practice, due to the existence of noise, one can only expect such relation to hold approximately, i.e. $\mathbf{U}_i \approx \mathbf{P}_i\mathbf{W}_i$. Incorporating such relation to a tensor completion model via penalization, we have the following formulation

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{G}},\mathbf{U}_i,\mathbf{W}_i} L(\boldsymbol{\mathcal{G}}, \{\mathbf{U}\}_{i=1}^3) + \sum_{i=1}^3 \frac{\alpha_i|\Omega|}{2}\|\mathbf{U}_i - \mathbf{P}_i\mathbf{W}_i\|_F^2, \\ &\text{s.t.} \quad (\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3) \in \mathcal{M}_r, \end{aligned} \tag{6}$$

where $L(\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3) = \|\mathcal{P}_\Omega(\boldsymbol{\mathcal{G}} \times_{i=1}^3 \mathbf{U}_i - \boldsymbol{\mathcal{R}})\|_F^2/2$. Fixing $\boldsymbol{\mathcal{G}}$ and $\mathbf{U}_i$, with respect to $\mathbf{W}_i$, (6) has a close form solution

$$\mathbf{W}_i = (\mathbf{P}_i^\top\mathbf{P}_i)^{-1}\mathbf{P}_i^\top\mathbf{U}_i = \mathbf{P}_i^\top\mathbf{U}_i. \tag{7}$$

Since $\min_{x,y} l(x,y) = \min_x l(x, y(x))$ where $y(x) = \arg\min_y l(x,y)$, one can substitute (7) into the above problem and obtain the following equivalence

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{G}},\mathbf{U}_i} L(\boldsymbol{\mathcal{G}}, \{\mathbf{U}\}_{i=1}^3) + \sum_{i=1}^3 \frac{\alpha_i|\Omega|}{2}\operatorname{trace}(\mathbf{U}_i^T(\mathbf{I}_i - \mathbf{P}_i\mathbf{P}_i^T)\mathbf{U}_i) \\ &\triangleq f(\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3) \\ &\text{s.t.} \quad (\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3) \in \mathcal{M}_r. \end{aligned} \tag{8}$$

Although the cost function is already smooth over the total space $\mathcal{M}_r$, due to its invariance over the equivalent class $[\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3]$, there can be infinite local optima, which is extremely undesirable. Indeed, if $(\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3)$ is a local optimal of the objective, then so is every point in the infinite set $[\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3]$. One way to reduce the number of local optima is to mathematically treated the entire set $[\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3]$ as a point. Consequently, we redefine the cost by $\tilde{f}([\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3]) = f(\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3)$ and obtain the following Remainnian optimization problem over the quotient manifold $\mathcal{M}_r/\sim$:

$$\min_{[\boldsymbol{\mathcal{X}}]} \tilde{f}([\boldsymbol{\mathcal{X}}]) \quad \text{s.t.} \quad [\boldsymbol{\mathcal{X}}] \in \mathcal{M}_r/\sim . \tag{9}$$

Remark 1. In Riemannian optimization literature, problem (8) is called the lifted representation of problem (9) over the total space [Absil *et al.*, 2009]. This model is closely related to the Laplace regularization model [Narita *et al.*, 2011]. Concretely, they share the same form:

$$\min_{\boldsymbol{\mathcal{G}},\mathbf{U}_i} L(\boldsymbol{\mathcal{G}}, \{\mathbf{U}_i\}_{i=1}^3) + \sum_{i=1}^3 \frac{C_i}{2}\operatorname{trace}(\mathbf{U}_i^\top\mathbf{L}_i\mathbf{U}_i). \tag{10}$$

The difference lies in that $\mathbf{L}_i$ is a projection matrix in our case, while, in the Laplace regularization model, $\mathbf{L}_i$ is a Laplacian matrix.

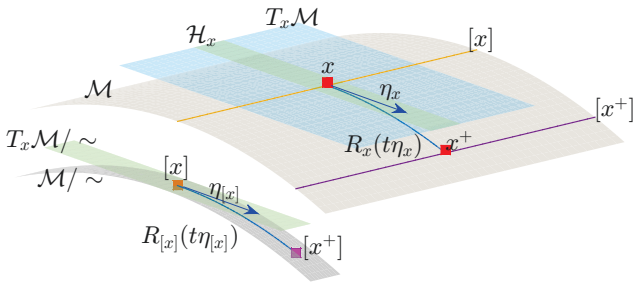

Figure 1: Optimization Framework for Quotient Manifold: most Riemannian solvers are based on the iteration formula: $[x^+] \leftarrow R_{[x]}(t\eta_{[x]})$, where $t > 0$ is the stepsize, $\eta_{[x]}$ is the search direction picked from current tangent space $T_{[x]}\mathcal{M}/\sim$, and $R_{[x]}(\cdot)$ is the retraction, i.e. a map from current tangent space to $\mathcal{M}/\sim$. Due to the abstractness of quotient manifold, such iteration is often lifted to (represented in) the total space as $x = R_x(t\eta_x)$ where $x \in [x]$, η_x is the horizontal lift of $\eta_{[x]}$, and $R_x(\cdot)$ is the lifted retraction. Such representation is possible only if $\mathcal{M}/\sim$ has the structure of Riemannian quotient, that is the total space is endowed with an invariant Riemannian metric.

Remark 2. Since each $[\boldsymbol{\mathcal{X}}] \in \mathcal{M}_r/\sim$ has a unique tensor representation in $\boldsymbol{\mathcal{X}} \in \mathcal{F}_r$, we show that the abstract model (9) can be represented as a concrete model over the manifold $\mathcal{F}_r$. Specifically, the following Proposition interprets the proposed model as an optimization problem with a regularizer that encourages the mode-i space of the estimated tensor close to $\operatorname{span}(\mathbf{P}_i)$.

Proposition 2. *if $[\boldsymbol{\mathcal{X}}]$ is a critical point of problem (9) then its tensor representation $\boldsymbol{\mathcal{X}}$ is a critical point of the following problem.*

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}} \in \mathcal{F}_r} &\frac{1}{2}\|\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{R}})\|_F^2 \\ &+ \sum_{i=1}^3 \frac{\alpha_i|\Omega|}{2}\operatorname{dist}^2(\operatorname{span}(\boldsymbol{\mathcal{X}}_{(i)}), \operatorname{span}(\mathbf{P}_i)) \end{aligned}$$

where $\operatorname{dist}(\cdot,\cdot)$ *is the Chodal distance [Ye and Lim, 2014] between two subspaces. And vice versa.*

4 Riemannian Conjugate Gradient Descent

We depict the optimization framework for quotient manifolds in Fig. 1. Under this framework, we solve the proposed problem (9) by Riemannian Conjugate Gradient descent (CG). With the details specified later, we list our CG solver for problem (9) in Alg. 1, where the CG direction is composed in the Polak-Ribiere+ manner with the momentum weight $\beta^{(k)}$ computed by Flecher-Reeves formula [Absil *et al.*, 2009], and $\mathcal{T}_k(\cdot)$ is the projector of horizontal space $\mathcal{H}_{\overline{\boldsymbol{\mathcal{X}}^{(k)}}}$. The convergence property of CGSI (Alg. 1) to a stationary point follows the general analysis of CG method [Sato and Iwai, 2015]. To represent CGSI in concrete tensor formulations, four items must be specified: the Riemannian metric $\langle\cdot,\cdot\rangle_{\overline{\boldsymbol{\mathcal{X}}}}$, the Riemannian gradient $\operatorname{grad} f(\overline{\boldsymbol{\mathcal{X}}})$, the retraction $R_{\overline{\boldsymbol{\mathcal{X}}}}(\cdot)$, and the projector onto horizontal space $\mathcal{T}_{\overline{\boldsymbol{\mathcal{X}}}}$.

4.1 Metric Tuning

Riemannian metric $\langle\cdot,\cdot\rangle_{\overline{\boldsymbol{\mathcal{X}}}}$ of $\mathcal{M}_r$ is an inner product defined over each tangent space $T_{\overline{\boldsymbol{\mathcal{X}}}}\mathcal{M}_r$. A high-quality Riemannian solver for a quotient manifold should be equipped with

Algorithm 1 CGSI: a Riemannian CG method

Require: Initializer $\overline{\boldsymbol{\mathcal{X}}^{(0)}} = (\boldsymbol{\mathcal{G}}^{(0)}, \{\mathbf{U}_i^{(0)}\}_{i=1}^3)$ and tolerance ϵ
1: $k = 0$;
2: $\eta^{(-1)} = (\mathbf{0}, \{\mathbf{0}\}_{i=1}^3)$;
3: **repeat**
4: compute current Riemannian gradient $\xi^{(k)} = \operatorname{grad} f(\boldsymbol{\mathcal{X}}^{(k)})$;
5: compose CG direction $\eta^{(k)} = -\xi^{(k)} + \beta^{(k)}\mathcal{T}_k(\eta^{(k-1)})$;
6: choose a step size $t_k > 0$;
7: update by retraction $\overline{\boldsymbol{\mathcal{X}}^{(k+1)}} = R_{\overline{\boldsymbol{\mathcal{X}}^{(k)}}}(t_k\eta^{(k)})$;
8: $k = k + 1$;
9: **until** $\langle \xi^{(k-1)}, \xi^{(k-1)} \rangle_{\overline{\boldsymbol{\mathcal{X}}^{(k-1)}}} \leq \epsilon$;
10: **return** $\overline{\boldsymbol{\mathcal{X}}^{(k)}}$.

a well-tuned metric, because (1) the metric determines the differential structure of the quotient manifold, and more importantly (2) it implicitly endows the solver with a preconditioner, which heavily affects the convergent rate [Mishra and Sepulchre, 2014; Mishra, 2014].

From the perspective of preconditioning, it seems that the best candidate is the Newton metric $\langle \eta, \xi \rangle_{\overline{\boldsymbol{\mathcal{X}}}} = D^2 f(\overline{\boldsymbol{\mathcal{X}}})[\eta, \xi] \forall \eta, \xi \in T_{\overline{\boldsymbol{\mathcal{X}}}}\mathcal{M}_r$ where $D^2 f(\overline{\boldsymbol{\mathcal{X}}})$ is the second order differential of the cost function. However, under such metric, computing the search direction involves solving a large system of linear equations, which precludes the Newton metric from the application to huge datasets. Therefore, we propose to use the following alternative:

$$\begin{aligned}\langle \eta_{\overline{\boldsymbol{\mathcal{X}}}}, \xi_{\overline{\boldsymbol{\mathcal{X}}}} \rangle_{\overline{\boldsymbol{\mathcal{X}}}} &= D_b^2 g(\overline{\boldsymbol{\mathcal{X}}})[\eta_{\overline{\boldsymbol{\mathcal{X}}}}, \xi_{\overline{\boldsymbol{\mathcal{X}}}}] \\ &= \sum_{i=1}^{3} \langle \eta_i, \xi_i \boldsymbol{\mathcal{G}}_{(i)} \boldsymbol{\mathcal{G}}_{(i)}^\top \rangle + \langle \eta_{\boldsymbol{\mathcal{G}}}, \xi_{\boldsymbol{\mathcal{G}}} \rangle \\ &\quad + \sum_{i=1}^{3} N\alpha_i \langle \eta_i, (\mathbf{I}_i - \mathbf{P}_i\mathbf{P}_i^\top)\xi_i \rangle,\end{aligned} \tag{11}$$

where $g(\overline{\boldsymbol{\mathcal{X}}})$ is a scaled approximation to the original cost function, and $D_b^2 g(\overline{\boldsymbol{\mathcal{X}}})$ is the block approximation of its second derivative, specifically $g(\overline{\boldsymbol{\mathcal{X}}}) \triangleq \frac{1}{2}\|\boldsymbol{\mathcal{G}} \times_1^3 \mathbf{U}_i - \boldsymbol{\mathcal{R}}\|_F^2 + \sum_{i=1}^3 \frac{\alpha_i N}{2} \operatorname{trace}(\mathbf{U}_i^T(\mathbf{I}_i - \mathbf{P}_i\mathbf{P}_i^T)\mathbf{U}_i)$ with $N = n_1 n_2 n_3$.

Our metric is more scalable than Newton metric. The following Proposition indicates that the scale gradient induced by this metric can be computed with $O(\sum_{i=1}^3 n_i k_i r_i + r_i^3)$ additional operations.

Proposition 3. *Suppose that the cost function $f(\cdot)$ has Euclidean gradient $\nabla f(\overline{\boldsymbol{\mathcal{X}}}) = (\nabla_{\boldsymbol{\mathcal{G}}} f, \{\nabla_{\mathbf{U}_i} f\}_{i=1}^3)$. Then its scaled gradient $\tilde{\nabla} f(\overline{\boldsymbol{\mathcal{X}}})$ under the metric (11) can be computed by:*

$$\begin{aligned}\tilde{\nabla}_{\boldsymbol{\mathcal{G}}} f(\overline{\boldsymbol{\mathcal{X}}}) &= \nabla_{\boldsymbol{\mathcal{G}}} f(\overline{\boldsymbol{\mathcal{X}}}) \\ \tilde{\nabla}_{\mathbf{U}_i} f(\overline{\boldsymbol{\mathcal{X}}}) &= \mathbf{E}_i \mathbf{G}_i^{-1} + \mathbf{F}_i (\mathbf{G}_i + N\alpha_i \mathbf{I}_i)^{-1}\end{aligned}$$

where $\mathbf{E}_i = \mathbf{P}_i\mathbf{P}_i^\top \nabla_{\mathbf{U}_i} f$, $\mathbf{F}_i = \nabla_{\mathbf{U}_i} f - \mathbf{E}_i$, and $\mathbf{G}_i = \boldsymbol{\mathcal{G}}_{(i)} \boldsymbol{\mathcal{G}}_{(i)}^\top$.

The final proposition suggests that the proposed metric makes the representation of solvers in the total space possible.

Proposition 4. *The quotient manifold $\mathcal{M}_r / \sim$ admits a structure of Riemannian quotient manifold, if $\mathcal{M}_r$ is endowed with the Riemannian metric defined in (11).*

Projector	Formulation
$\Psi_{\overline{\boldsymbol{\mathcal{X}}}}(\mathbf{Z}_{\boldsymbol{\mathcal{G}}}, \{\mathbf{Z}_i\}_{i=1}^3)$ projection of an ambient vector $(\mathbf{Z}_{\boldsymbol{\mathcal{G}}}, \{\mathbf{Z}_i\}_{i=1}^3)$ onto $T_{\overline{\boldsymbol{\mathcal{X}}}}\mathcal{M}_r$	$(\mathbf{Z}_{\boldsymbol{\mathcal{G}}}, \{\mathbf{Z}_i - \mathbf{V}_i\mathbf{S}_i\mathbf{G}_i^{-1} - \mathbf{W}_i\mathbf{S}_i\mathbf{G}_{\alpha_i}^{-1}\}_{i=1}^3)$ where $\mathbf{S}_i$ is the solution of : $\begin{cases} \operatorname{sym}(\mathbf{V}_i^T\mathbf{V}_i\mathbf{S}_i\mathbf{G}_i^{-1} - \mathbf{U}_i^\top\mathbf{Z}_i) \\ \quad = -\operatorname{sym}(\mathbf{W}_i^T\mathbf{W}_i\mathbf{S}_i\mathbf{G}_{\alpha_i}^{-1}) \\ \mathbf{S}_i = \mathbf{S}_i^\top \end{cases}$
$\Pi_{\overline{\boldsymbol{\mathcal{X}}}}(\eta_{\overline{\boldsymbol{\mathcal{X}}}})$ Projection of a tangent vector $\eta_{\overline{\boldsymbol{\mathcal{X}}}}$ of total space onto $\mathcal{H}_{\overline{\boldsymbol{\mathcal{X}}}}$	$(\eta_{\boldsymbol{\mathcal{G}}} + \sum_{1\leq i\leq 3} \boldsymbol{\mathcal{G}} \times_i \boldsymbol{\Omega}_i, \{\eta_i - \mathbf{U}_i\boldsymbol{\Omega}_i\}_{i=1}^3)$ where $(\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \boldsymbol{\Omega}_3)$ is the solution of $\begin{cases} \operatorname{skw}(\mathbf{V}_i^T\mathbf{V}_i\boldsymbol{\Omega}_i\mathbf{G}_i + \mathbf{G}_i\boldsymbol{\Omega}_i \\ \quad + \mathbf{W}_i^\top\mathbf{W}_i\boldsymbol{\Omega}_i\mathbf{G}_{\alpha_i}) \\ \quad - \boldsymbol{\mathcal{G}}_{(i)}(\mathbf{I}_{j_i} \otimes \boldsymbol{\Omega}_{k_i})\boldsymbol{\mathcal{G}}_{(i)}{}^\top \\ \quad - \boldsymbol{\mathcal{G}}_{(i)}(\boldsymbol{\Omega}_{j_i} \otimes \mathbf{I}_{k_i})\boldsymbol{\mathcal{G}}_{(i)}{}^\top \\ = \operatorname{skw}(\mathbf{V}_i^\top\eta_i\mathbf{G}_i + \mathbf{W}_i^\top\eta_i\mathbf{G}_{\alpha_i}) \\ \quad + \operatorname{skw}(\boldsymbol{\mathcal{G}}_{(i)}(\eta_{\boldsymbol{\mathcal{G}}})_{(i)}^\top) \\ \\ \boldsymbol{\Omega}_i^\top = -\boldsymbol{\Omega}_i \forall i \in \{1,2,3\} \end{cases}$

Table 1: Expressions of Projectors. We define the following matrices: $\mathbf{V}_i := \mathbf{P}_i\mathbf{P}_i^\top\mathbf{U}_i, \mathbf{W}_i := \mathbf{U}_i - \mathbf{V}_i$, $\mathbf{G}_i := \boldsymbol{\mathcal{G}}_{(i)}\boldsymbol{\mathcal{G}}_{(i)}{}^\top$, $\mathbf{G}_{\alpha_i} := N\alpha_i\mathbf{I}_i + \boldsymbol{\mathcal{G}}_{(i)}\boldsymbol{\mathcal{G}}_{(i)}{}^\top$. $j_i = \max\{k | k \in \{1,2,3\}, k \neq i\}$ and $k_i = \min\{k | k \in \{1,2,3\}, k \neq i\}$. And the operator $\operatorname{sym}(\cdot)$ and $\operatorname{skw}(\cdot)$ extract the symmetric and skew components of a matrix respectively, i.e. $\operatorname{sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^\top)/2$ and $\operatorname{skw}(\mathbf{A}) = (\mathbf{A} - \mathbf{A}^\top)/2$. Note that the above linear systems can be solved by MATLAB command *pcg* in $O(\sum_{1\leq i\leq 3}(n_i k_i^2 + r_i^3))$ flops.

4.2 Other Optimization Related Items

Projectors: To derive the optimization related items, two orthogonal projectors, $\Psi_{\overline{\boldsymbol{\mathcal{X}}}}(\cdot)$ and $\Pi_{\overline{\boldsymbol{\mathcal{X}}}}(\cdot)$, are required. The former projects a vector onto the tangent space $T_{\overline{\boldsymbol{\mathcal{X}}}}\mathcal{M}_r$, and the latter is a projector from the tangent space onto the horizontal space $\mathcal{H}_{\overline{\boldsymbol{\mathcal{X}}}}$. The orthogonality of both projectors is measured by the metric (11). For lack of space, the mathematical derivation is deferred to a long version of this paper.

Riemannian Gradient: According to [Absil *et al.*, 2009], the Riemannian gradient can be computed by projecting the scaled gradient onto tangent space, specifically

$$\operatorname{grad} f(\overline{\boldsymbol{\mathcal{X}}}) = \Psi_{\overline{\boldsymbol{\mathcal{X}}}}(\tilde{\nabla} f(\overline{\boldsymbol{\mathcal{X}}})). \tag{12}$$

Retraction: We use the retraction defined by

$$R_{\overline{\boldsymbol{\mathcal{X}}}}(\eta_{\overline{\boldsymbol{\mathcal{X}}}}) = (\boldsymbol{\mathcal{G}} + \eta_{\boldsymbol{\mathcal{G}}}, \{\operatorname{uf}(\mathbf{U}_i + \eta_i)\}_{i=1}^3). \tag{13}$$

where $\operatorname{uf}(\cdot)$ extracts the orthogonal component from a matrix. Such retraction is proposed by [Kasai and Mishra, 2016]. In the long version of this paper, we give rigorous analysis to prove that the above retraction is compatible with the proposed metric.

5 Experiments

We validate the effectiveness of the proposed solver CGSI by comparing it with the state-of-the-art. The baseline can be partitioned into three classes. The first class contains Riemannian solvers including GeomCG [Kressner *et al.*, 2014], FTC [Kasai and Mishra, 2016], and gHOI [Liu *et al.*, 2016]. The second class consists of Euclidean solvers that take no account of the side information, including AltMin [Romera-Paredes *et al.*, 2013] and HalRTC [Liu *et al.*, 2013]. The third class comprises of two methods that incorporate side information, including RUBIK [Wang *et al.*, 2015] and TFAI

Table 2: Performance of the compared methods on hyperspectral images.

		AltMin		FTC		GeomCG		gHOI		HalRTC		RUBIK		TFAI		CGSI	
data	OS	NRSE	Time(s)	NRSE	Time	NRSE	Time	NRSE	Time	NRSE	Time	NRSE	Time	NRSE	Time	NRSE	Time
S1	3	0.161	183	0.091	**52**	0.113	61	0.115	65	0.080	177	0.086	197	0.161	164	**0.062**	77
	5	0.156	307	0.067	**76**	0.077	93	0.103	109	0.078	177	0.085	194	0.159	273	**0.040**	100
	7	0.156	429	0.060	**100**	0.056	124	0.092	152	0.077	177	0.085	195	0.159	382	**0.039**	110
	9	0.156	550	0.046	**126**	0.044	151	0.078	195	0.077	178	0.085	198	0.156	479	**0.036**	126
S2	3	0.173	183	0.093	**50**	0.114	61	0.125	65	0.066	173	0.061	197	0.173	165	**0.048**	83
	5	0.166	306	0.082	**76**	0.076	92	0.100	103	0.066	171	0.061	196	0.171	203	**0.043**	96
	7	0.166	428	0.073	**101**	0.064	123	0.091	152	0.057	172	0.061	197	0.169	386	**0.040**	110
	9	0.166	578	0.062	**125**	0.056	154	0.073	197	0.057	171	0.060	197	0.169	433	**0.038**	130
S3	3	0.033	226	0.041	**68**	0.044	181	0.043	187	0.034	174	0.062	189	0.063	131	**0.025**	83
	5	0.033	346	0.030	**99**	0.029	251	0.037	308	0.033	177	0.061	185	0.062	209	**0.021**	108
	7	0.033	486	0.023	**124**	0.021	389	0.033	177	0.031	177	0.059	187	0.057	210	**0.018**	131
	9	0.033	587	0.019	**156**	0.021	386	0.031	491	0.029	172	0.034	189	0.033	229	**0.017**	143
S4	3	0.033	238	0.031	**78**	0.036	181	0.038	193	0.047	172	0.034	182	0.033	155	**0.012**	105
	5	0.033	359	0.015	**108**	0.015	254	0.031	293	0.032	171	0.037	183	0.033	247	**0.012**	118
	7	0.033	486	0.012	**128**	0.012	391	0.021	177	0.029	177	0.027	180	0.033	181	**0.011**	131
	9	0.033	600	0.012	170	0.012	398	0.018	492	0.026	177	0.024	192	0.033	231	**0.010**	**144**
S5	3	0.059	236	0.051	**75**	0.077	180	0.169	187	0.086	169	0.126	180	0.062	99	**0.024**	104
	5	0.059	362	0.041	**104**	0.051	254	0.113	289	0.076	171	0.059	183	0.061	128	**0.022**	114
	7	0.059	483	0.034	137	0.037	325	0.089	398	0.047	173	0.054	190	0.061	181	**0.021**	**128**
	9	0.059	603	0.028	166	0.029	400	0.065	494	0.042	173	0.058	192	0.061	229	**0.021**	**142**
S6	3	0.090	237	0.067	**76**	0.057	181	0.132	189	0.095	177	0.090	180	0.091	170	**0.036**	107
	5	0.090	356	0.039	**105**	0.040	251	0.095	298	0.083	177	0.081	180	0.091	213	**0.034**	119
	7	0.090	489	0.039	**130**	0.040	325	0.095	394	0.083	178	0.081	181	0.091	300	**0.034**	136
	9	0.090	600	0.039	165	0.040	396	0.095	501	0.083	178	0.081	183	0.091	383	**0.034**	**143**
S7	3	0.071	245	0.073	**82**	0.069	181	0.075	193	0.077	172	0.069	181	0.072	165	**0.031**	119
	5	0.072	377	0.034	**102**	0.032	225	0.064	293	0.069	172	0.067	180	0.072	203	**0.028**	158
	7	0.072	581	0.028	161	0.028	336	0.052	452	0.062	171	0.064	181	0.072	302	**0.026**	**157**
	9	0.072	603	0.027	**170**	0.027	400	0.041	494	0.057	173	0.058	183	0.072	183	**0.026**	189
S8	3	0.039	236	0.030	**74**	0.042	181	0.050	187	0.071	174	0.034	179	0.040	131	**0.013**	103
	5	0.039	354	0.018	**107**	0.019	247	0.038	293	0.061	174	0.040	182	0.045	213	**0.012**	114
	7	0.039	701	0.013	**102**	0.013	381	0.030	234	0.031	181	0.030	182	0.060	363	**0.011**	169
	9	0.039	853	0.012	**112**	0.012	502	0.026	369	0.027	175	0.031	183	0.039	502	**0.011**	180

Figure 2: Visual results of the recovered 27th frame of scene7 when OS is set to 3.

[Narita *et al.*, 2011]. All the experiments are performed in Matlab on the same machine with 3.0 GHz Intel E5-2690 CPU and 128GB RAM.

5.1 Hyperspectral Image Inpainting

A hyperspectral image is a tensor whose the slices are photographs of the same scene under different wavelets. We adopt the dataset provided in [Foster *et al.*, 2006] which contains images about eight different rural scenes taken under 33 various wavelets. To make all methods in our baseline applicable to the completion problem, we resize each hyperspectral images to a small dimension such that $n_1 = 306$, $n_2 = 402$, and $n_3 = 33$. Empirically, we treat these graphs as tensors of rank $r = (30, 30, 6)$. The observed pixels, or the training set, are sampled from the tensors uniformly at random. And the sample size is set to $|\Omega| = OS \times p$ in which OS is so-called Over-Sampling ratio and $p = \sum_{i=1}^{3}(n_i r_i - r_i^2) + r_1 r_2 r_3$ is the number of free parameters in a size n tensor with rank r. In addition to the observed entries, the mode-1 feature matrix is constructed by extracting the top-$(r_1 + 10)$ singular vectors from a matrix of size $n_1 \times 10r_1$ whose columns are sampled from the mode-1 fibers of the hyperspectral graphs. The recovery accuracy is measured by Normalized Root mean Square Error (NRSE) [Kressner *et al.*, 2014]. All the compared methods are terminated when the training NRSE is less than 0.003 or iterate more than 300 epochs. We report the NRSE and CPU time of the compared methods in Tab. 2. From the table, we can see that the proposed method has much higher accuracy than the other solvers in our baseline. The empirical results also indicate that the sparser the observed pixels are the higher CGSI's improvement is on the recovery accuracy. The visual results of the 27th slices of recovered hyperspectral images of scene 7 are illustrated in Fig. 2.

5.2 Recommender System

In recommendation tasks, two datasets are considered: MovieLens 10M (ML10M) and MovieLens 20M (ML20M). Both datasets contain the rating history of users for items at specific moments. For both datasets, we partition the samples into 731 slices in terms of time stamp. Those slices have the identical time intervals. Accordingly, the completion tasks for the two datasets are of sizes $71567 \times 10681 \times 731$ and $138493 \times 26744 \times 731$ respectively. In addition to the rating history, both datasets contain two extra files: one describes the genres of each movie, and the other contains tags of each

Table 3: Performance of the compared methods on Recommendation Tasks.

		AltMin		FTC		GeomCG		gHOI		TFAI		CGSI	
dataset	rank	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time	RMSE	Time
ML10M	(4,4,4)	0.982	924	0.824	236	0.835	307	1.076	467	1.011	426	**0.823**	**178**
	(6,6,6)	0.968	1830	**0.814**	535	0.826	679	1.262	1035	0.9948	942	**0.814**	**434**
	(8,8,8)	1.010	3123	0.822	928	0.833	1135	1.062	1734	0.993	1617	**0.810**	**754**
	(10,10,10)	1.147	4963	0.824	1631	0.843	2220	1.094	2788	0.992	2522	**0.807**	**1067**
ML20M	(4,4,4)	1.061	690	0.822	466	0.829	601	1.050	918	1.029	797	**0.818**	**363**
	(6,6,6)	1.089	3451	0.808	**982**	0.822	1309	1.057	1869	1.008	1644	**0.805**	1107
	(8,8,8)	1.092	5890	0.812	**1725**	0.828	2271	1.045	3363	1.004	3144	**0.804**	1739
	(10,10,10)	1.092	9418	0.818	3161	0.834	4308	1.054	5795	1.025	5394	**0.799**	**2813**

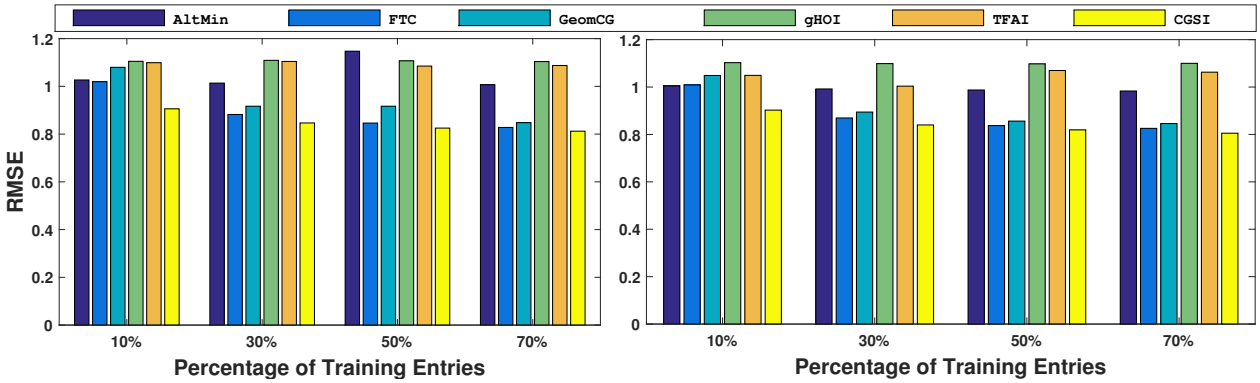

Figure 3: Accuracy of compared methods under different size of training set

movie. We construct a corpus that contains the text description of all movies from the genres descriptions and all the tags. The feature matrix is extracted from the above corpus by the latent semantic analysis (LSA) method. The processing is efficient since LSA is implemented via randomized SVD.

Various empirical studies are conducted to validate the performance of the proposed method. In the first scenario, we record the CPU time and the Root Mean Square Error (RMSE) outputted by the compared algorithms under different choices of multi-linear rank. In this scenario, for both datasets, 80% samples are chosen as training set, and the rest are left for testing. The results are listed in Tab. 3, which suggests that the proposed method outperforms all other solvers in terms of accuracy. For ML10M, our method uses significantly less CPU time than its competitors. In Fig. 3, we report another scenario, in which the percentage of training samples are varied from 10% to 70% and the rank parameter is fixed to $(10, 10, 10)$. Experimental results in this figure indicate that our method has the lowest RMSE.

To show the impact of parameter α on the performance of our method, we depict the relation between RMSE and α in Fig. 4, where the rank parameter is set to $(10, 10, 10)$, and percentage of training samples is set to 80%. From this Figure we can see that our method has higher accuracy than the vanilla Riemannian model's solver FTC for a wide range of parameter choices.

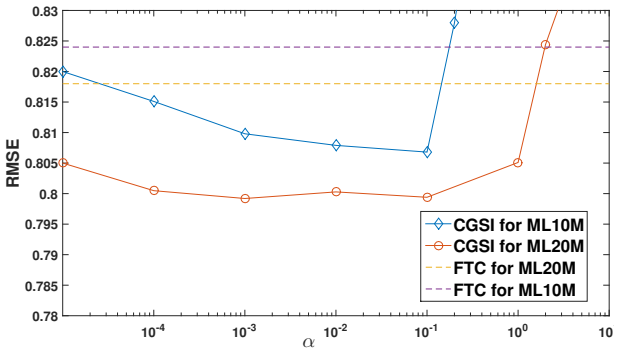

Figure 4: Effect of parameter α on the accuracy of CGSI.

6 Conclusion

In this paper, we exploit the side information to improve the accuracy of Riemannian tensor completion. A novel Riemannian model is proposed. To solve the model efficiently, we design a new Riemannian metric that implicitly induce an adaptive preconditioner for the solving procedure. Then, we devise a Riemannian conjugate gradient descent method using the well-tuned metric. Empirical results show that our solver outperforms the state-of-the-art.

Acknowledgments

This work is partially supported by National Natural Science Foundation of China (Grant No: 61472347, 61672376, and 61672449).

References

[Absil *et al.*, 2009] P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[Acar *et al.*, 2011] Evrim Acar, Tamara G. Kolda, and Daniel M. Dunlavy. All-at-once optimization for coupled matrix and tensor factorizations. In *MLG'11*, 2011.

[Bell and Koren, 2007] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *Acm Sigkdd Explorations Newsletter*, 9(2):75–79, 2007.

[Beutel *et al.*, 2014] Alex Beutel, Partha Pratim Talukdar, Abhimanu Kumar, Christos Faloutsos, Evangelos E Papalexakis, and Eric P Xing. Flexifact: Scalable flexible factorization of coupled tensors on hadoop. In *ICDM*, pages 109–117. SIAM, 2014.

[Chen *et al.*, 2013] Shouyuan Chen, Michael R Lyu, Irwin King, and Zenglin Xu. Exact and stable recovery of pairwise interaction tensors. In *NIPS*, 2013.

[Filipović and Jukić, 2015] Marko Filipović and Ante Jukić. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional systems and signal processing*, 26(3):677–692, 2015.

[Foster *et al.*, 2006] David H Foster, Kinjiro Amano, Sérgio MC Nascimento, and Michael J Foster. Frequency of metamerism in natural scenes. *Journal of the Optical Society ofAmerica A*, 23:2359–2372, 2006.

[Jain and Oh, 2014] Prateek Jain and Sewoong Oh. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*, pages 1431–1439, 2014.

[Kasai and Mishra, 2016] Hiroyuki Kasai and Bamdev Mishra. Low-rank tensor completion: a riemannian manifold preconditioning approach. In *ICML*, 2016.

[Kolda and Bader, 2009] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[Kressner *et al.*, 2014] Daniel Kressner, Michael Steinlechner, and Bart Vandereycken. Low-rank tensor completion by riemannian optimization. *BIT Numerical Mathematics*, 54(2):447–468, 2014.

[Liu *et al.*, 2013] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):208–220, 2013.

[Liu *et al.*, 2014] Yuanyuan Liu, Fanhua Shang, Hong Cheng, James Cheng, and Hanghang Tong. Factor matrix trace norm minimization for low-rank tensor completion. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 866–874. SIAM, 2014.

[Liu *et al.*, 2015] Qiang Liu, Shu Wu, and Liang Wang. Cot: Contextual operating tensor for context-aware recommender systems. In *AAAI*, pages 203–209, 2015.

[Liu *et al.*, 2016] Yuanyuan Liu, Fanhua Shang, Wei Fan, James Cheng, and Hong Cheng. Generalized higher order orthogonal iteration for tensor learning and decomposition. *IEEE transactions on neural networks and learning systems*, 27(12):2551–2563, 2016.

[Mishra and Sepulchre, 2014] Bamdev Mishra and Rodolphe Sepulchre. R3mc: A riemannian three-factor algorithm for low-rank matrix completion. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 1137–1142. IEEE, 2014.

[Mishra and Sepulchre, 2016] Bamdev Mishra and Rodolphe Sepulchre. Riemannian preconditioning. *SIAM Journal on Optimization*, 26(1):635–660, 2016.

[Mishra, 2014] Bamdev Mishra. *A Riemannian approach to large-scale constrained least-squares with symmetries*. PhD thesis, Universite de Liege, Liege, Belgique, 2014.

[Narita *et al.*, 2011] Atsuhiro Narita, Kohei Hayashi, Ryota Tomioka, and Hisashi Kashima. Tensor factorization using auxiliary information. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 501–516. Springer, 2011.

[Rai *et al.*, 2015] Piyush Rai, Yingjian Wang, and Lawrence Carin. Leveraging features and networks for probabilistic tensor decomposition. In *AAAI*, pages 2942–2948, 2015.

[Romera-Paredes and Pontil, 2013] Bernardino Romera-Paredes and Massimiliano Pontil. A new convex relaxation for tensor completion. In *Advances in Neural Information Processing Systems*, pages 2967–2975, 2013.

[Romera-Paredes *et al.*, 2013] Bernardino Romera-Paredes, Hane Aung, Nadia Bianchi-Berthouze, and Massimiliano Pontil. Multilinear multitask learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1444–1452, 2013.

[Sato and Iwai, 2015] Hiroyuki Sato and Toshihiro Iwai. A new, globally convergent riemannian conjugate gradient method. *Optimization*, 64(4):1011–1031, 2015.

[Smith *et al.*, 2016] Shaden Smith, Jongsoo Park, and George Karypis. An exploration of optimization algorithms for high performance tensor completion. In *Proceedings of the 2016 ACM/IEEE conference on Supercomputing*, 2016.

[Wang *et al.*, 2015] Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C Denny, Abel Kho, You Chen, Bradley A Malin, and Jimeng Sun. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1265–1274. ACM, 2015.

[Xu *et al.*, 2015] Yangyang Xu, Ruru Hao, Wotao Yin, and Zhixun Su. Parallel matrix factorization for low-rank tensor completion. *Inverse Problems & Imaging*, 9(2), 2015.

[Ye and Lim, 2014] Ke Ye and Lek-Heng Lim. Distance between subspaces of different dimensions. *arXiv preprint arXiv:1407.0900*, 2014.

[Yuan and Zhang, 2015] Ming Yuan and Cun-Hui Zhang. On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, pages 1–38, 2015.

[Zhang and Aeron, 2016] Zemin Zhang and Shuchin Aeron. Exact tensor completion using t-svd. *IEEE Transactions on Signal Processing*, 2016.

[Zhang *et al.*, 2014] Xiaoqin Zhang, Zhengyuan Zhou, Di Wang, and Yi Ma. Hybrid singular value thresholding for tensor completion. In *AAAI*, pages 1362–1368, 2014.