Low-rank tensor completion: a Riemannian manifold preconditioning approach

Hiroyuki Kasai KASAI@IS.UEC.AC.JP
The University of Electro-Communications,1-5-1, Chofu-gaoka, Chofu-shi, Tokyo, 182-8585, Japan

Bamdev Mishra BAMDEVM@AMAZON.COM
Amazon Development Centre India, Bengaluru 560055, Karnataka, India

Abstract

We propose a novel Riemannian manifold preconditioning approach for the tensor completion problem with rank constraint. A novel Riemannian metric or inner product is proposed that exploits the least-squares structure of the cost function and takes into account the structured symmetry that exists in Tucker decomposition. The specific metric allows to use the versatile framework of Riemannian optimization on quotient manifolds to develop preconditioned nonlinear conjugate gradient and stochastic gradient descent algorithms for batch and online setups, respectively. Concrete matrix representations of various optimization-related ingredients are listed. Numerical comparisons suggest that our proposed algorithms robustly outperform state-of-the-art algorithms across different synthetic and real-world datasets.

1. Introduction

This paper addresses the problem of low-rank tensor completion when the rank is a priori known or estimated. We focus on 3-order tensors in the paper, but the developments can be generalized to higher order tensors in a straightforward way. Given a tensor $\boldsymbol{\mathcal{X}}^{n_1 \times n_2 \times n_3}$, whose entries $\boldsymbol{\mathcal{X}}^{\star}_{i_1,i_2,i_3}$ are only known for some indices $(i_1, i_2, i_3) \in \Omega$, where Ω is a subset of the complete set of indices $\{(i_1, i_2, i_3) : i_d \in \{1, \ldots, n_d\}, d \in \{1, 2, 3\}\}$, the *fixed-rank tensor completion problem* is formulated as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}} & \frac{1}{|\Omega|} \|\boldsymbol{\mathcal{P}}_{\Omega}(\boldsymbol{\mathcal{X}}) - \boldsymbol{\mathcal{P}}_{\Omega}(\boldsymbol{\mathcal{X}}^{\star})\|_F^2 \\ \text{subject to} & \text{rank}(\boldsymbol{\mathcal{X}}) = \mathbf{r}, \end{array} \quad (1)$$

where the operator $\boldsymbol{\mathcal{P}}_{\Omega}(\boldsymbol{\mathcal{X}})_{i_1,i_2,i_3} = \boldsymbol{\mathcal{X}}_{i_1,i_2,i_3}$ if $(i_1, i_2, i_3) \in \Omega$ and $\boldsymbol{\mathcal{P}}_{\Omega}(\boldsymbol{\mathcal{X}})_{i_1,i_2,i_3} = 0$ otherwise and (with a slight abuse of notation) $\| \cdot \|_F$ is the Frobenius norm. $|\Omega|$ is the number of known entries. $\text{rank}(\boldsymbol{\mathcal{X}})$ $(= \mathbf{r} = (r_1, r_2, r_3))$, called the *multilinear rank* of $\boldsymbol{\mathcal{X}}$, is the set of the ranks of for each of mode-d unfolding matrices. $r_d \ll n_d$ enforces a low-rank structure. The *mode* is a matrix obtained by concatenating the mode-d fibers along columns, and mode-d *unfolding* of a D-order tensor $\boldsymbol{\mathcal{X}}$ is $\mathbf{X}_d \in \mathbb{R}^{n_d \times n_{d+1} \cdots n_D n_1 \cdots n_{d-1}}$ for $d = \{1, \ldots, D\}$.

Problem (1) has many variants, and one of those is extending the nuclear norm regularization approach from the matrix case (Candès & Recht, 2009) to the tensor case. This results in a summation of nuclear norm regularization terms, each one corresponds to each of the unfolding matrices of $\boldsymbol{\mathcal{X}}$. While this generalization leads to good results (Liu et al., 2013; Tomioka et al., 2011; Signoretto et al., 2014), its applicability to large-scale instances is not trivial, especially due to the necessity of high-dimensional singular value decomposition computations. A different approach exploits *Tucker decomposition* (Kolda & Bader, 2009, Section 4) of a low-rank tensor $\boldsymbol{\mathcal{X}}$ to develop large-scale algorithms for (1), e.g., in (Filipović & Jukić, 2013; Kressner et al., 2014).

The present paper exploits both the *symmetry* present in Tucker decomposition and the *least-squares* structure of the cost function of (1) to develop competitive algorithms. The multilinear rank constraint forms a smooth manifold (Kressner et al., 2014). To this end, we use the concept of *manifold preconditioning*. While preconditioning in unconstrained optimization is well studied (Nocedal & Wright, 2006, Chapter 5), preconditioning on constraints with *symmetries*, owing to non-uniqueness of Tucker decomposition (Kolda & Bader, 2009), is not straightforward. We build upon the recent work (Mishra & Sepulchre, 2016) that suggests to use *preconditioning* with a *tailored metric* (inner product) in the Riemannian optimization framework on quo-

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

tient manifolds (Absil et al., 2008; Edelman et al., 1998; Mishra & Sepulchre, 2016). The differences with respect to the work of Kressner et al. (2014), which also exploits the manifold structure, are twofold. (i) Kressner et al. (2014) exploit the search space as an *embedded submanifold* of the Euclidean space, whereas we view it as a product of simpler search spaces with symmetries. Consequently, certain computations have straightforward interpretation. (ii) Kressner et al. (2014) work with the standard Euclidean metric, whereas we use a metric that is tuned to the least-squares cost function, thereby inducing a preconditioning effect. This novel idea of using a tuned metric leads to a superior performance of our algorithms. They also connect to state-of-the-art algorithms proposed in (Ngo & Saad, 2012; Wen et al., 2012; Mishra & Sepulchre, 2014; Boumal & Absil, 2015).

The paper is organized as follows. Section 2 discusses the two fundamental structures of symmetry and least-squares associated with (1) and proposes a novel metric that captures the relevant second order information of the problem. The optimization-related ingredients on the Tucker manifold are developed in Section 3. The cost function specific ingredients are developed in Section 4. The final formulas are listed in Table 1, which allow to develop preconditioned conjugate gradient descent algorithm in the batch setup and stochastic gradient descent algorithm in the online setup. In Section 5, numerical comparisons with state-of-the-art algorithms on various synthetic and real-world benchmarks suggest a superior performance of our proposed algorithms. Our proposed algorithms are implemented in the Matlab toolbox Manopt (Boumal et al., 2014). The concrete proofs of propositions, development of optimization-related ingredients, and additional numerical experiments are shown in Sections **A** and **B**, respectively, of the supplementary material file. The Matlab codes for first and second order implementations, e.g., gradient descent and trust-region methods, are available at `https://bamdevmishra.com/codes/tensorcompletion/`.

2. Exploiting the problem structure

Construction of efficient algorithms depends on properly exploiting the problem structure. To this end, we focus on two fundamental structures in (1): *symmetry* in the constraints and the *least-squares structure* of the cost function. Finally, a novel metric is proposed.

The symmetry structure in Tucker decomposition. The Tucker decomposition of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ of rank $\mathbf{r}$ (=(r_1, r_2, r_3)) is

$$\boldsymbol{\mathcal{X}} \quad = \quad \boldsymbol{\mathcal{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \tag{2}$$

where $\mathbf{U}_d \in \mathrm{St}(r_d, n_d)$ for $d \in \{1, 2, 3\}$ belongs to the *Stiefel manifold* of matrices of size $n_d \times r_d$ with orthogonal columns and $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ (Kolda & Bader, 2009). Here, $\boldsymbol{\mathcal{W}} \times_d \mathbf{V} \in \mathbb{R}^{n_1 \times \cdots n_{d-1} \times m \times n_{d+1} \times \cdots n_D}$ computes the *d-mode product* of a tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{n_1 \times \cdots \times n_D}$ and a matrix $\mathbf{V} \in \mathbb{R}^{m \times n_d}$. Tucker decomposition (2) is *not unique* as $\boldsymbol{\mathcal{X}}$ remains unchanged under the transformation

$$\begin{array}{l}(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}}) \mapsto \\ \quad (\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T)\end{array} \tag{3}$$

for all $\mathbf{O}_d \in \mathcal{O}(r_d)$, which is the set of orthogonal matrices of size of $r_d \times r_d$. The classical remedy to remove this indeterminacy is to have additional structures on $\boldsymbol{\mathcal{G}}$ like sparsity or restricted orthogonal rotations (Kolda & Bader, 2009, Section 4.3). In contrast, we encode the transformation (3) in an abstract search space of *equivalence classes*, defined as,

$$\begin{array}{l}[(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})] := \{(\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \\ \qquad \boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T) : \mathbf{O}_d \in \mathcal{O}(r_d)\}.\end{array} \tag{4}$$

The set of equivalence classes is the *quotient manifold* (Lee, 2003)

$$\mathcal{M}/\sim := \mathcal{M}/(\mathcal{O}(r_1) \times \mathcal{O}(r_2) \times \mathcal{O}(r_3)), \tag{5}$$

where $\mathcal{M}$ is called the *total space* (computational space) that is the product space

$$\mathcal{M} := \mathrm{St}(r_1, n_1) \times \mathrm{St}(r_2, n_2) \times \mathrm{St}(r_3, n_3) \times \mathbb{R}^{r_1 \times r_2 \times r_3}. \tag{6}$$

Due to the invariance (3), the local minima of (1) in $\mathcal{M}$ are not isolated, but they become isolated on $\mathcal{M}/\sim$. Consequently, the problem (1) is an optimization problem on a quotient manifold for which systematic procedures are proposed in (Absil et al., 2008; Edelman et al., 1998). A requirement is to endow endow $\mathcal{M}/\sim$ with a Riemannian structure, which conceptually translates (1) into an unconstrained optimization problem over the search space $\mathcal{M}/\sim$. We call $\mathcal{M}/\sim$, defined in (5), the *Tucker manifold* as it results from Tucker decomposition.

The least-squares structure of the cost function. In unconstrained optimization, the Newton method is interpreted as a *scaled* steepest descent method, where the search space is endowed with a metric (inner product) induced by the Hessian of the cost function (Nocedal & Wright, 2006). This induced metric (or its approximation) resolves convergence issues of first order optimization algorithms. Analogously, finding a good inner product for (1) is of profound consequence. Specifically for the case of quadratic optimization with rank constraint (matrix case), Mishra and Sepulchre (Mishra & Sepulchre, 2016) propose a family of Riemannian metrics from the Hessian of the cost function. Applying this approach directly for the particular cost function of (1) is computationally costly. To circumvent the issue, we consider a simplified cost function by assuming that Ω contains the full set of indices, i.e., we focus on $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$ to propose a metric candidate. Applying the metric tuning approach of (Mishra & Sepulchre, 2016) to the simplified cost function leads to a family of Riemannian metrics. A good trade-off between computational cost and simplicity is by considering only the *block diagonal* elements of the Hessian of $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$. It should be noted that the cost function $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$ is *convex and quadratic* in $\boldsymbol{\mathcal{X}}$. Consequently, it is also convex and quadratic in the arguments $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ individually. Equivalently, the block diagonal approximation of the Hessian of $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{X}}^\star\|_F^2$ in $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ is

$$((\mathbf{G}_1\mathbf{G}_1^T) \otimes \mathbf{I}_{n_1}, (\mathbf{G}_2\mathbf{G}_2^T) \otimes \mathbf{I}_{n_2}, (\mathbf{G}_3\mathbf{G}_3^T) \otimes \mathbf{I}_{n_3}, \mathbf{I}_{r_1r_2r_3}), \tag{7}$$

where $\mathbf{G}_d$ is the mode-d unfolding of $\boldsymbol{\mathcal{G}}$ and is assumed to be full rank. $\otimes$ is the Kronecker product. The terms $\mathbf{G}_d\mathbf{G}_d^T$ for $d \in \{1, 2, 3\}$ are *positive definite* when $r_1 \leq r_2r_3$, $r_2 \leq r_1r_3$, and $r_3 \leq r_1r_2$, which is a reasonable assumption.

A novel Riemannian metric. An element x in the total space $\mathcal{M}$ has the matrix representation $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$. Consequently, the tangent space $T_x\mathcal{M}$ is the Cartesian product of the tangent spaces of the individual manifolds of (6), i.e., $T_x\mathcal{M}$ has the matrix characterization (Edelman et al., 1998)

$$\begin{array}{rcl} T_x\mathcal{M} & = & \{(\mathbf{Z}_{\mathbf{U}_1}, \mathbf{Z}_{\mathbf{U}_2}, \mathbf{Z}_{\mathbf{U}_3}, \mathbf{Z}_{\boldsymbol{\mathcal{G}}}) \\ & & \in \mathbb{R}^{n_1 \times r_1} \times \mathbb{R}^{n_2 \times r_2} \times \mathbb{R}^{n_3 \times r_3} \times \mathbb{R}^{r_1 \times r_2 \times r_3} : \\ & & \mathbf{U}_d^T\mathbf{Z}_{\mathbf{U}_d} + \mathbf{Z}_{\mathbf{U}_d}^T\mathbf{U}_d = 0, \text{ for } d \in \{1, 2, 3\}\}. \end{array} \tag{8}$$

From the earlier discussion on symmetry and least-squares structure, we propose the novel metric or inner product $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$

$$\begin{array}{rcl} g_x(\xi_x, \eta_x) & = & \langle \xi_{\mathbf{U}_1}, \eta_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)\rangle + \langle \xi_{\mathbf{U}_2}, \eta_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)\rangle \\ & & + \langle \xi_{\mathbf{U}_3}, \eta_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)\rangle + \langle \xi_{\boldsymbol{\mathcal{G}}}, \eta_{\boldsymbol{\mathcal{G}}}\rangle, \end{array} \tag{9}$$

where $\xi_x, \eta_x \in T_x\mathcal{M}$ are tangent vectors with matrix characterizations, shown in (8), $(\xi_{\mathbf{U}_1}, \xi_{\mathbf{U}_2}, \xi_{\mathbf{U}_3}, \xi_{\boldsymbol{\mathcal{G}}})$ and $(\eta_{\mathbf{U}_1}, \eta_{\mathbf{U}_2}, \eta_{\mathbf{U}_3}, \eta_{\boldsymbol{\mathcal{G}}})$, respectively and $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. It should be emphasized that the proposed metric (9) is induced from (7).

Proposition 1. *Let* $(\xi_{\mathbf{U}_1}, \xi_{\mathbf{U}_2}, \xi_{\mathbf{U}_3}, \xi_{\boldsymbol{\mathcal{G}}})$ *and* $(\eta_{\mathbf{U}_1}, \eta_{\mathbf{U}_2}, \eta_{\mathbf{U}_3}, \eta_{\boldsymbol{\mathcal{G}}})$ *be tangent vectors to the quotient manifold (5) at* $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$, *and* $(\xi_{\mathbf{U}_1\mathbf{O}_1}, \xi_{\mathbf{U}_2\mathbf{O}_2}, \xi_{\mathbf{U}_3\mathbf{O}_3}, \xi_{\boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T})$ *and* $(\eta_{\mathbf{U}_1\mathbf{O}_1}, \eta_{\mathbf{U}_2\mathbf{O}_2}, \eta_{\mathbf{U}_3\mathbf{O}_3}, \eta_{\boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T})$ *be tangent vectors to the quotient manifold (5) at* $(\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T)$. *The metric (9) is invariant along the equivalence class (4), i.e.,*

$$\begin{array}{l} g_{(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})}((\xi_{\mathbf{U}_1}, \xi_{\mathbf{U}_2}, \xi_{\mathbf{U}_3}, \xi_{\boldsymbol{\mathcal{G}}}), (\eta_{\mathbf{U}_1}, \eta_{\mathbf{U}_2}, \eta_{\mathbf{U}_3}, \eta_{\boldsymbol{\mathcal{G}}})) \\ \quad = g_{(\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T)} \\ \qquad ((\xi_{\mathbf{U}_1\mathbf{O}_1}, \xi_{\mathbf{U}_2\mathbf{O}_2}, \xi_{\mathbf{U}_3\mathbf{O}_3}, \xi_{\boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T}), \\ \qquad (\eta_{\mathbf{U}_1\mathbf{O}_1}, \eta_{\mathbf{U}_2\mathbf{O}_2}, \eta_{\mathbf{U}_3\mathbf{O}_3}, \eta_{\boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T})). \end{array}$$

3. Notions of manifold optimization

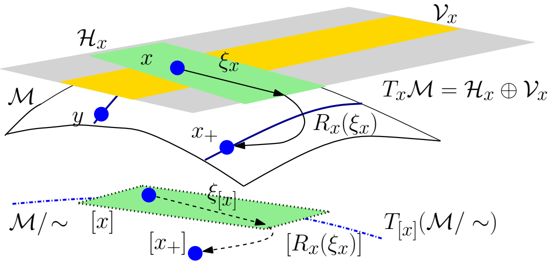


Figure 1. Riemannian optimization framework: geometric objects, shown in dotted lines, on quotient manifold $\mathcal{M}/\sim$ call for matrix representatives, shown in solid lines, in the total space $\mathcal{M}$.

Each point on a quotient manifold represents an entire equivalence class of matrices in the total space. Abstract geometric objects on the quotient manifold $\mathcal{M}/\sim$ call for matrix representatives in the total space $\mathcal{M}$. Similarly, algorithms are run in the total space $\mathcal{M}$, but under appropriate compatibility between the Riemannian structure of $\mathcal{M}$ and the Riemannian structure of the quotient manifold $\mathcal{M}/\sim$, they define algorithms on the quotient manifold. The key is endowing $\mathcal{M}/\sim$ with a Riemannian structure. Once this is the case, a constraint optimization problem, for example (1), is conceptually transformed into an unconstrained optimization over the Riemannian quotient manifold (5). Below we briefly show the development of various geometric objects that are required to optimize a smooth cost function on the quotient manifold (5) with first order methods, e.g., conjugate gradients.

Quotient manifold representation and horizontal lifts. Figure 1 illustrates a schematic view of optimization with equivalence classes, where the points x and y in $\mathcal{M}$ belong to the same equivalence class (shown in solid blue color) and they represent a single point $[x] := \{y \in \mathcal{M} : y \sim x\}$ on the quotient manifold $\mathcal{M}/\sim$. The abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ at $[x] \in \mathcal{M}/\sim$ has the matrix representation in $T_x\mathcal{M}$, but restricted to the directions that do not induce a displacement along the equivalence class $[x]$. This is realized by decomposing $T_x\mathcal{M}$ into two complementary subspaces, the vertical and horizontal subspaces. The vertical space $\mathcal{V}_x$ is the tangent space of the equivalence class $[x]$. On the other hand, the horizontal space $\mathcal{H}_x$ is the *orthogonal subspace* to $\mathcal{V}_x$ in the sense of the metric (9). Equivalently, $T_x\mathcal{M} = \mathcal{V}_x \oplus \mathcal{H}_x$. The horizontal subspace $\mathcal{H}_x$ provides a valid matrix representation to the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$. An abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ at $[x]$ has a unique element $\xi_x \in \mathcal{H}_x$ that is called its *horizontal lift*.

A Riemannian metric $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ at $x \in \mathcal{M}$ defines a Riemannian metric $g_{[x]} : T_{[x]}(\mathcal{M}/\sim) \times T_{[x]}(\mathcal{M}/\sim) \to \mathbb{R}$, i.e., $g_{[x]}(\xi_{[x]}, \eta_{[x]}) := g_x(\xi_x, \eta_x)$ on the quotient manifold $\mathcal{M}/\sim$, if $g_x(\xi_x, \eta_x)$ does not depend on a specific representation along the equivalence class $[x]$. Here, $\xi_{[x]}$ and $\eta_{[x]}$ are tangent vectors in $T_{[x]}(\mathcal{M}/\sim)$, and ξ_x and η_x are their horizontal lifts in $\mathcal{H}_x$ at x, respectively. Equivalently, the definition of the Riemannian metric is well posed when $g_x(\xi_x, \zeta_x) = g_x(\xi_y, \zeta_y)$ for all $x, y \in [x]$, where $\xi_x, \zeta_x \in \mathcal{H}_x$ and $\xi_y, \zeta_y \in \mathcal{H}_y$ are the horizontal lifts of $\xi_{[x]}, \zeta_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ along the same equivalence class $[x]$. This holds true for the proposed metric (9) as shown in Proposition 1. From (Absil et al., 2008), endowed with the Riemannian metric (9), the quotient manifold $\mathcal{M}/\sim$ is a *Riemannian submersion* of $\mathcal{M}$. The submersion principle allows to work out concrete matrix representations of abstract object on $\mathcal{M}/\sim$, e.g., the gradient of a smooth cost function (Absil et al., 2008).

Starting from an arbitrary matrix (with appropriate dimensions), two linear projections are needed: the first projection Ψ_x is onto the tangent space $T_x\mathcal{M}$, while the second projection Π_x is onto the horizontal subspace $\mathcal{H}_x$. The computation cost of these is $O(n_1r_1^2 + n_2r_2^2 + n_3r_3^2)$.

The tangent space $T_x\mathcal{M}$ projection is obtained by extracting the component normal to $T_x\mathcal{M}$ in the ambient space. The normal space $N_x\mathcal{M}$ has the matrix characterization $\{(\mathbf{U}_1\mathbf{S}_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)^{-1}, \mathbf{U}_2\mathbf{S}_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)^{-1}, \mathbf{U}_3\mathbf{S}_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)^{-1}, 0) : \mathbf{S}_{\mathbf{U}_d} \in \mathbb{R}^{r_d \times r_d}, \mathbf{S}_{\mathbf{U}_d}^T = \mathbf{S}_{\mathbf{U}_d}, \text{ for } d \in \{1,2,3\}\}$. Symmetric matrices $\mathbf{S}_{\mathbf{U}_d}$ for all $d \in \{1,2,3\}$ parameterize the normal space. Finally, the operator $\Psi_x : \mathbb{R}^{n_1 \times r_1} \times \mathbb{R}^{n_2 \times r_2} \times \mathbb{R}^{n_3 \times r_3} \times \mathbb{R}^{r_1 \times r_2 \times r_3} \to T_x\mathcal{M} : (\mathbf{Y}_{\mathbf{U}_1}, \mathbf{Y}_{\mathbf{U}_2}, \mathbf{Y}_{\mathbf{U}_3}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}}) \mapsto \Psi_x(\mathbf{Y}_{\mathbf{U}_1}, \mathbf{Y}_{\mathbf{U}_2}, \mathbf{Y}_{\mathbf{U}_3}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}})$ is given as follows.

Proposition 2. *The quotient manifold (5) endowed with the metric (9) admits the tangent space projector defined as*

$$\begin{aligned}\Psi_x(\mathbf{Y}_{\mathbf{U}_1}, \mathbf{Y}_{\mathbf{U}_2}, \mathbf{Y}_{\mathbf{U}_3}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}}) = (&\mathbf{Y}_{\mathbf{U}_1} - \mathbf{U}_1\mathbf{S}_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)^{-1},\\ &\mathbf{Y}_{\mathbf{U}_2} - \mathbf{U}_2\mathbf{S}_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)^{-1}, \mathbf{Y}_{\mathbf{U}_3} - \mathbf{U}_3\mathbf{S}_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)^{-1}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}}),\end{aligned} \tag{10}$$

where $\mathbf{S}_{\mathbf{U}_d}$ *is the solution to the Lyapunov equation* $\mathbf{S}_{\mathbf{U}_d}\mathbf{G}_d\mathbf{G}_d^T + \mathbf{G}_d\mathbf{G}_d^T\mathbf{S}_{\mathbf{U}_d} = \mathbf{G}_d\mathbf{G}_d^T(\mathbf{Y}_{\mathbf{U}_d}^T\mathbf{U}_d + \mathbf{U}_d^T\mathbf{Y}_{\mathbf{U}_d})\mathbf{G}_d\mathbf{G}_d^T$ *for* $d \in \{1,2,3\}$.

The Lyapunov equations in Proposition 2 are solved efficiently with the Matlab's `lyap` routine.

The horizontal space projection of a tangent vector is obtained by removing the component along the vertical space. The vertical space $\mathcal{V}_x$ has the matrix characterization $\{(\mathbf{U}_1\boldsymbol{\Omega}_1, \mathbf{U}_2\boldsymbol{\Omega}_2, \mathbf{U}_3\boldsymbol{\Omega}_3, -(\boldsymbol{\mathcal{G}}\times_1\boldsymbol{\Omega}_1 + \boldsymbol{\mathcal{G}}\times_2\boldsymbol{\Omega}_2 + \boldsymbol{\mathcal{G}}\times_3\boldsymbol{\Omega}_3)) : \boldsymbol{\Omega}_d \in \mathbb{R}^{r_d \times r_d}, \boldsymbol{\Omega}_d^T = -\boldsymbol{\Omega}_d \text{ for } d \in \{1,2,3\}\}$. Skew symmetric matrices $\boldsymbol{\Omega}_d$ for all $d \in \{1,2,3\}$ parameterize the vertical space. Finally, the horizontal projection operator $\Pi_x : T_x\mathcal{M} :\to \mathcal{H}_x : \eta_x \mapsto \Pi_x(\eta_x)$ is given as follows.

Proposition 3. *The quotient manifold (5) endowed with the metric (9) admits the horizontal projector defined as*

$$\begin{aligned}\Pi_x(\eta_x) \quad = \quad &(\eta_{\mathbf{U}_1} - \mathbf{U}_1\boldsymbol{\Omega}_1, \eta_{\mathbf{U}_2} - \mathbf{U}_2\boldsymbol{\Omega}_2, \eta_{\mathbf{U}_3} - \mathbf{U}_3\boldsymbol{\Omega}_3,\\ &\eta_{\boldsymbol{\mathcal{G}}} - (-(\boldsymbol{\mathcal{G}}\times_1\boldsymbol{\Omega}_1 + \boldsymbol{\mathcal{G}}\times_2\boldsymbol{\Omega}_2 + \boldsymbol{\mathcal{G}}\times_3\boldsymbol{\Omega}_3))),\end{aligned}$$

where $\eta_x = (\eta_{\mathbf{U}_1}, \eta_{\mathbf{U}_2}, \eta_{\mathbf{U}_3}, \eta_{\boldsymbol{\mathcal{G}}}) \in T_x\mathcal{M}$ *and* $\boldsymbol{\Omega}_d$ *is a skew-symmetric matrix of size* $r_d \times r_d$ *that is the solution to the coupled Lyapunov equations*

$$\left\{\begin{aligned}&\mathbf{G}_1\mathbf{G}_1^T\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1\mathbf{G}_1\mathbf{G}_1^T\\ &\quad -\mathbf{G}_1(\mathbf{I}_{r_3} \otimes \boldsymbol{\Omega}_2)\mathbf{G}_1^T - \mathbf{G}_1(\boldsymbol{\Omega}_3 \otimes \mathbf{I}_{r_2})\mathbf{G}_1^T\\ &\qquad = \mathrm{Skew}(\mathbf{U}_1^T\eta_{\mathbf{U}_1}\mathbf{G}_1\mathbf{G}_1^T) + \mathrm{Skew}(\mathbf{G}_1\eta_{\mathbf{G}_1}^T),\\ &\mathbf{G}_2\mathbf{G}_2^T\boldsymbol{\Omega}_2 + \boldsymbol{\Omega}_2\mathbf{G}_2\mathbf{G}_2^T\\ &\quad -\mathbf{G}_2(\mathbf{I}_{r_3} \otimes \boldsymbol{\Omega}_1)\mathbf{G}_2^T - \mathbf{G}_2(\boldsymbol{\Omega}_3 \otimes \mathbf{I}_{r_1})\mathbf{G}_2^T\\ &\qquad = \mathrm{Skew}(\mathbf{U}_2^T\eta_{\mathbf{U}_2}\mathbf{G}_2\mathbf{G}_2^T) + \mathrm{Skew}(\mathbf{G}_2\eta_{\mathbf{G}_2}^T),\\ &\mathbf{G}_3\mathbf{G}_3^T\boldsymbol{\Omega}_3 + \boldsymbol{\Omega}_3\mathbf{G}_3\mathbf{G}_3^T\\ &\quad -\mathbf{G}_3(\mathbf{I}_{r_2} \otimes \boldsymbol{\Omega}_1)\mathbf{G}_3^T - \mathbf{G}_3(\boldsymbol{\Omega}_2 \otimes \mathbf{I}_{r_1})\mathbf{G}_3^T\\ &\qquad = \mathrm{Skew}(\mathbf{U}_3^T\eta_{\mathbf{U}_3}\mathbf{G}_3\mathbf{G}_3^T) + \mathrm{Skew}(\mathbf{G}_3\eta_{\mathbf{G}_3}^T),\end{aligned}\right. \tag{11}$$

where $\mathrm{Skew}(\cdot)$ *extracts the skew-symmetric part of a square matrix, i.e.,* $\mathrm{Skew}(\mathbf{D}) = (\mathbf{D} - \mathbf{D}^T)/2$.

The coupled Lyapunov equations (11) are solved efficiently with the Matlab's `pcg` routine that is combined with a specific symmetric preconditioner resulting from the Gauss-Seidel approximation of (11). For the variable $\boldsymbol{\Omega}_1$, the preconditioner is of the form $\mathbf{G}_1\mathbf{G}_1^T\boldsymbol{\Omega}_1 + \boldsymbol{\Omega}_1\mathbf{G}_1\mathbf{G}_1^T$. Similarly, for the variables $\boldsymbol{\Omega}_2$ and $\boldsymbol{\Omega}_3$.

Retraction. A retraction is a mapping that maps vectors in the horizontal space to points on the search space $\mathcal{M}$ and satisfies the local rigidity condition (Absil et al., 2008). It provides a natural way to move on the manifold along a search direction. Because the total space $\mathcal{M}$ has the product nature, we can choose a retraction by combining retractions on the individual manifolds, i.e., $R_x(\xi_x) = (\mathrm{uf}(\mathbf{U}_1 + \xi_{\mathbf{U}_1}), \mathrm{uf}(\mathbf{U}_2 + \xi_{\mathbf{U}_2}), \mathrm{uf}(\mathbf{U}_3 + \xi_{\mathbf{U}_3}), \boldsymbol{\mathcal{G}} + \xi_{\boldsymbol{\mathcal{G}}})$, where $\xi_x \in \mathcal{H}_x$ and $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of a full column rank matrix, i.e., $\mathrm{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1/2}$. The retraction R_x defines a retraction $R_{[x]}(\xi_{[x]}) := [R_x(\xi_x)]$ on the quotient manifold $\mathcal{M}/\sim$, as the equivalence class $[R_x(\xi_x)]$ does not depend on specific matrix representations of $[x]$ and $\xi_{[x]}$, where ξ_x is the horizontal lift of the abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$.

Vector transport. A vector transport on a manifold $\mathcal{M}$ is a smooth mapping that transports a tangent vector $\xi_x \in T_x\mathcal{M}$ at $x \in \mathcal{M}$ to a vector in the tangent space at a point $R_x(\eta_x)$. It is defined by the symbol $\mathcal{T}_{\eta_x}\xi_x$. It generalizes the classical concept of translation of vectors in the Euclidean space to manifolds (Absil et al., 2008, Section 8.1.4). The horizontal lift of the abstract vector transport $\mathcal{T}_{\eta_{[x]}}\xi_{[x]}$ on $\mathcal{M}/\sim$ has the matrix characterization $\Pi_{R_x(\eta_x)}(\mathcal{T}_{\eta_x}\xi_x) = \Pi_{R_x(\eta_x)}(\Psi_{R_x(\eta_x)}(\xi_x))$, where ξ_x and η_x are the horizontal lifts in $\mathcal{H}_x$ of $\xi_{[x]}$ and $\eta_{[x]}$ that belong to $T_{[x]}(\mathcal{M}/\sim)$. $\Psi_x(\cdot)$ and $\Pi_x(\cdot)$ are projectors defined in Propositions 2 and 3. The computational cost of transporting a vector solely depends on the projection and retraction operations.

4. Riemannian algorithms for (1)

We propose two Riemannian preconditioned algorithms for the tensor completion problem (1) that are based on the developments in Section 3. The preconditioning effect follows from the specific choice of the metric (9). In the batch setting, we use the off-the-shelf conjugate gradient implementation of Manopt for any smooth cost function (Boumal et al., 2014). A complete description of the Riemannian nonlinear conjugate gradient method is in (Absil et al., 2008, Chapter 8). In the online setting, we use the stochastic gradient descent implementation (Bonnabel, 2013). For fixed rank, theoretical convergence of the Riemannian algorithms are to a stationary point, and the convergence analysis follows from (Sato & Iwai, 2015; Ring & Wirth, 2012; Bonnabel, 2013). However, as simulations show, convergence to global minima is observed in many challenging instances.

In addition to the manifold-related ingredients in Section 3, the ingredients needed are the cost function specific ones. To this end, we show the computation of the Riemannian gradient as well as a way to compute an initial guess for the step-size, which is used in the conjugate gradient method. The concrete formulas are shown in Table 1.

Riemannian gradient computation. Let $f(\boldsymbol{\mathcal{X}}) = \|\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star)\|_F^2/|\Omega|$ be the mean square error function of (1), and $\boldsymbol{\mathcal{S}} = 2(\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{G}}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star))/|\Omega|$ be an auxiliary sparse tensor variable that is interpreted as the Euclidean gradient of f in $\mathbb{R}^{n_1\times n_2\times n_3}$. The partial derivatives of f with respect to $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ are computed in terms of the unfolding matrices $\mathbf{S}_d$. Due to the specific scaled metric (9), the partial derivatives are further scaled by $((\mathbf{G}_1\mathbf{G}_1^T)^{-1}, (\mathbf{G}_2\mathbf{G}_2^T)^{-1}, (\mathbf{G}_3\mathbf{G}_3^T)^{-1}, \boldsymbol{\mathcal{I}})$, denoted as $\mathrm{egrad}_x f$ (after scaling). Finally, from the Riemannian submersion theory (Absil et al., 2008, Section 3.6.2), the horizontal lift of $\mathrm{grad}_{[x]} f$ is equal to $\mathrm{grad}_x f = \Psi(\mathrm{egrad}_x f)$. The total numerical cost of computing the Riemannian gradient depends on computing the partial derivatives, which is $O(|\Omega| r_1 r_2 r_3)$.

Proposition 4. *The cost function (1) at $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ under the quotient manifold (5) endowed with the Riemannian metric (9) admits the horizontal lift of the Riemannian gradient*

$$\begin{array}{l}(\mathbf{S}_1(\mathbf{U}_3\otimes\mathbf{U}_2)\mathbf{G}_1^T(\mathbf{G}_1\mathbf{G}_1^T)^{-1} - \mathbf{U}_1\mathbf{B}_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)^{-1},\\ \mathbf{S}_2(\mathbf{U}_3\otimes\mathbf{U}_1)\mathbf{G}_2^T(\mathbf{G}_2\mathbf{G}_2^T)^{-1} - \mathbf{U}_2\mathbf{B}_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)^{-1},\\ \mathbf{S}_3(\mathbf{U}_2\otimes\mathbf{U}_1)\mathbf{G}_3^T(\mathbf{G}_3\mathbf{G}_3^T)^{-1} - \mathbf{U}_3\mathbf{B}_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)^{-1},\\ \boldsymbol{\mathcal{S}}\times_1\mathbf{U}_1^T\times_2\mathbf{U}_2^T\times_3\mathbf{U}_3^T),\end{array} \tag{12}$$

where $\mathbf{B}_{\mathbf{U}_d}$ for $d \in \{1,2,3\}$ are the solutions to the Lyapunov equations

$$\begin{cases}\mathbf{B}_{\mathbf{U}_1}\mathbf{G}_1\mathbf{G}_1^T + \mathbf{G}_1\mathbf{G}_1^T\mathbf{B}_{\mathbf{U}_1}\\ \qquad = 2\mathrm{Sym}(\mathbf{G}_1\mathbf{G}_1^T\mathbf{U}_1^T(\mathbf{S}_1(\mathbf{U}_3\otimes\mathbf{U}_2)\mathbf{G}_2^T),\\ \mathbf{B}_{\mathbf{U}_2}\mathbf{G}_2\mathbf{G}_2^T + \mathbf{G}_2\mathbf{G}_2^T\mathbf{B}_{\mathbf{U}_2}\\ \qquad = 2\mathrm{Sym}(\mathbf{G}_2\mathbf{G}_2^T\mathbf{U}_2^T(\mathbf{S}_2(\mathbf{U}_3\otimes\mathbf{U}_1)\mathbf{G}_2^T),\\ \mathbf{B}_{\mathbf{U}_3}\mathbf{G}_3\mathbf{G}_3^T + \mathbf{G}_3\mathbf{G}_3^T\mathbf{B}_{\mathbf{U}_3}\\ \qquad = 2\mathrm{Sym}(\mathbf{G}_3\mathbf{G}_3^T\mathbf{U}_3^T(\mathbf{S}_3(\mathbf{U}_2\otimes\mathbf{U}_1)\mathbf{G}_3^T),\end{cases}$$

where $\mathrm{Sym}(\cdot)$ *extracts the symmetric part of a matrix.*

Initial guess for the step-size. Following (Mishra & Sepulchre, 2014; Vandereycken, 2013; Kressner et al., 2014), the least-squares structure of the cost function in (1) is exploited to compute a *linearized* step-size guess efficiently along a search direction by considering a polynomial approximation of degree 2 over the manifold. Given a search direction $\xi_x \in \mathcal{H}_x$, the step-size guess is $\arg\min_{s\in\mathbb{R}_+}\|\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{G}}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3 + s\boldsymbol{\mathcal{G}}\times_1\xi_{\mathbf{U}_1}\times_2\mathbf{U}_2\times_3\mathbf{U}_3 + s\boldsymbol{\mathcal{G}}\times_1\mathbf{U}_1\times_2\xi_{\mathbf{U}_2}\times_3\mathbf{U}_3 + s\boldsymbol{\mathcal{G}}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\xi_{\mathbf{U}_3} + s\xi_{\boldsymbol{\mathcal{G}}}\times_1\mathbf{U}_1\times_2\mathbf{U}_2\times_3\mathbf{U}_3) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star)\|_F^2$, which has a closed-form expression and the numerical cost of computing it is $O(|\Omega| r_1 r_2 r_3)$.

Stochastic gradient descent in online setting. In the online setting, we update $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ every time a frontal slice, i.e., a matrix $\in \mathbb{R}^{n_1\times n_2}$, is randomly sampled from $\boldsymbol{\mathcal{X}}^\star_{i_1,i_2,i_3}$. Equivalently, we assume that the tensor grows along the third dimension. More concretely, we calculate the *rank-one* Riemannian gradient (12) for the input slice. $(\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$ are updated by taking a step along the negative Riemannian gradient direction. Subsequently, we retract using R_x. A popular formula for the step-size γ_k at k-th update is $\gamma_k = \gamma_0/(1+\gamma_0\lambda k)$, where γ_0 is the initial step-size and λ is a fixed reduction factor. Following (Bottou, 2012), we select γ_0 in the *pre-training phase* using a *small sample size* of a training set. λ is fixed to 10^{-7}.

Computational cost. The total computational cost per iteration of our proposed conjugate gradient implementation is $O(|\Omega| r_1 r_2 r_3)$, where $|\Omega|$ is the number of known entries. It should be stressed that the computational cost of our conjugate gradient implementation is equal to that of (Kressner et al., 2014). In the online setting, each stochastic gradient descent update costs $O(|\Omega_{\mathrm{slice}}| r_1 r_2 + n_1 r_1^2 + n_2 r_2^2 + T r_3^2 + r_1 r_2 r_3)$, where $|\Omega_{\mathrm{slice}}|$ is the number of known entries of the current frontal slice of the incomplete tensor $\boldsymbol{\mathcal{X}}^\star_{i_1,i_2,i_3}$, and T is the number of slices that we have seen along n_3 direction.

Table 1. Tucker manifold related optimization ingredients for (1)

Matrix representation	$x = (\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \boldsymbol{\mathcal{G}})$
Computational space $\mathcal{M}$	$\mathrm{St}(r_1, n_1) \times \mathrm{St}(r_2, n_2) \times \mathrm{St}(r_3, n_3) \times \mathbb{R}^{r_1 \times r_2 \times r_3}$
Group action	$\{(\mathbf{U}_1\mathbf{O}_1, \mathbf{U}_2\mathbf{O}_2, \mathbf{U}_3\mathbf{O}_3, \boldsymbol{\mathcal{G}} \times_1 \mathbf{O}_1^T \times_2 \mathbf{O}_2^T \times_3 \mathbf{O}_3^T) : \mathbf{O}_d \in \mathcal{O}(r_d)$, for $d \in \{1, 2, 3\}\}$
Quotient space $\mathcal{M}/\sim$	$\mathrm{St}(r_1, n_1) \times \mathrm{St}(r_2, n_2) \times \mathrm{St}(r_3, n_3) \times \mathbb{R}^{r_1 \times r_2 \times r_3} /(\mathcal{O}(r_1) \times \mathcal{O}(r_2) \times \mathcal{O}(r_3))$
Ambient space	$\mathbb{R}^{n_1 \times r_1} \times \mathbb{R}^{n_2 \times r_2} \times \mathbb{R}^{n_3 \times r_3} \times \mathbb{R}^{r_1 \times r_2 \times r_3}$
Tangent vectors in $T_x\mathcal{M}$	$\{(\mathbf{Z}_{\mathbf{U}_1}, \mathbf{Z}_{\mathbf{U}_2}, \mathbf{Z}_{\mathbf{U}_3}, \mathbf{Z}_{\boldsymbol{\mathcal{G}}}) \in \mathbb{R}^{n_1 \times r_1} \times \mathbb{R}^{n_2 \times r_2} \times \mathbb{R}^{n_3 \times r_3} \times \mathbb{R}^{r_1 \times r_2 \times r_3}$ $: \mathbf{U}_d^T \mathbf{Z}_{\mathbf{U}_d} + \mathbf{Z}_{\mathbf{U}_d}^T \mathbf{U}_d = 0,$ for $d \in \{1, 2, 3\}\}$
Metric $g_x(\xi_x, \eta_x)$ for any $\xi_x, \eta_x \in T_x\mathcal{M}$	$\langle \xi_{\mathbf{U}_1}, \eta_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)\rangle + \langle \xi_{\mathbf{U}_2}, \eta_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)\rangle + \langle \xi_{\mathbf{U}_3}, \eta_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)\rangle + \langle \xi_{\boldsymbol{\mathcal{G}}}, \eta_{\boldsymbol{\mathcal{G}}}\rangle$
Vertical tangent vectors in $\mathcal{V}_x$	$\{(\mathbf{U}_1\boldsymbol{\Omega}_1, \mathbf{U}_2\boldsymbol{\Omega}_2, \mathbf{U}_3\boldsymbol{\Omega}_3, -(\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{\Omega}_1 + \boldsymbol{\mathcal{G}} \times_2 \boldsymbol{\Omega}_2 + \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{\Omega}_3)) :$ $\boldsymbol{\Omega}_d \in \mathbb{R}^{r_d \times r_d}, \boldsymbol{\Omega}_d^T = -\boldsymbol{\Omega}_d$, for $d \in \{1, 2, 3\}\}$
Horizontal tangent vectors in $\mathcal{H}_x$	$\{(\zeta_{\mathbf{U}_1}, \zeta_{\mathbf{U}_2}, \zeta_{\mathbf{U}_3}, \zeta_{\boldsymbol{\mathcal{G}}}) \in T_x\mathcal{M} : (\mathbf{G}_d\mathbf{G}_d^T)\zeta_{\mathbf{U}_d}^T \mathbf{U}_d + \zeta_{\mathbf{G}_d}\mathbf{G}_d^T$ is symmetric, for $d \in \{1, 2, 3\}\}$
$\Psi(\cdot)$ projects an ambient vector $(\mathbf{Y}_{\mathbf{U}_1}, \mathbf{Y}_{\mathbf{U}_2}, \mathbf{Y}_{\mathbf{U}_3}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}})$ onto $T_x\mathcal{M}$	$(\mathbf{Y}_{\mathbf{U}_1} - \mathbf{U}_1\mathbf{S}_{\mathbf{U}_1}(\mathbf{G}_1\mathbf{G}_1^T)^{-1}, \mathbf{Y}_{\mathbf{U}_2} - \mathbf{U}_2\mathbf{S}_{\mathbf{U}_2}(\mathbf{G}_2\mathbf{G}_2^T)^{-1},$ $\mathbf{Y}_{\mathbf{U}_3} - \mathbf{U}_3\mathbf{S}_{\mathbf{U}_3}(\mathbf{G}_3\mathbf{G}_3^T)^{-1}, \mathbf{Y}_{\boldsymbol{\mathcal{G}}})$, where $\mathbf{S}_{\mathbf{U}_d}$ for $d \in \{1, 2, 3\}$ are computed by solving Lyapunov equations as in (10).
$\Pi(\cdot)$ projects a tangent vector ξ onto $\mathcal{H}_x$	$(\xi_{\mathbf{U}_1} - \mathbf{U}_1\boldsymbol{\Omega}_1, \xi_{\mathbf{U}_2} - \mathbf{U}_2\boldsymbol{\Omega}_2, \xi_{\mathbf{U}_3} - \mathbf{U}_3\boldsymbol{\Omega}_3,$ $\xi_{\boldsymbol{\mathcal{G}}} - (-(\boldsymbol{\mathcal{G}} \times_1 \boldsymbol{\Omega}_1 + \boldsymbol{\mathcal{G}} \times_2 \boldsymbol{\Omega}_2 + \boldsymbol{\mathcal{G}} \times_3 \boldsymbol{\Omega}_3)))$, $\boldsymbol{\Omega}_d$ is computed in (11).
First order derivative of $f(x)$	$(\mathbf{S}_1(\mathbf{U}_3 \otimes \mathbf{U}_2)\mathbf{G}_1^T, \mathbf{S}_2(\mathbf{U}_3 \otimes \mathbf{U}_1)\mathbf{G}_2^T, \mathbf{S}_3(\mathbf{U}_2 \otimes \mathbf{U}_1)\mathbf{G}_3^T), \boldsymbol{\mathcal{S}} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \times_3 \mathbf{U}_3^T)$, where $\boldsymbol{\mathcal{S}} = \frac{2}{\lvert\Omega\rvert}(\boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{G}} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3) - \boldsymbol{\mathcal{P}}_\Omega(\boldsymbol{\mathcal{X}}^\star))$.
Retraction $R_x(\xi_x)$	$(\mathrm{uf}(\mathbf{U}_1 + \xi_{\mathbf{U}_1}), \mathrm{uf}(\mathbf{U}_2 + \xi_{\mathbf{U}_2}), \mathrm{uf}(\mathbf{U}_3 + \xi_{\mathbf{U}_3}), \boldsymbol{\mathcal{G}} + \xi_{\boldsymbol{\mathcal{G}}})$
Horizontal lift of the vector transport $\mathcal{T}_{\eta_{[x]}}\xi_{[x]}$	$\Pi_{R_x(\eta_x)}(\Psi_{R_x(\eta_x)}(\xi_x))$

5. Numerical comparisons

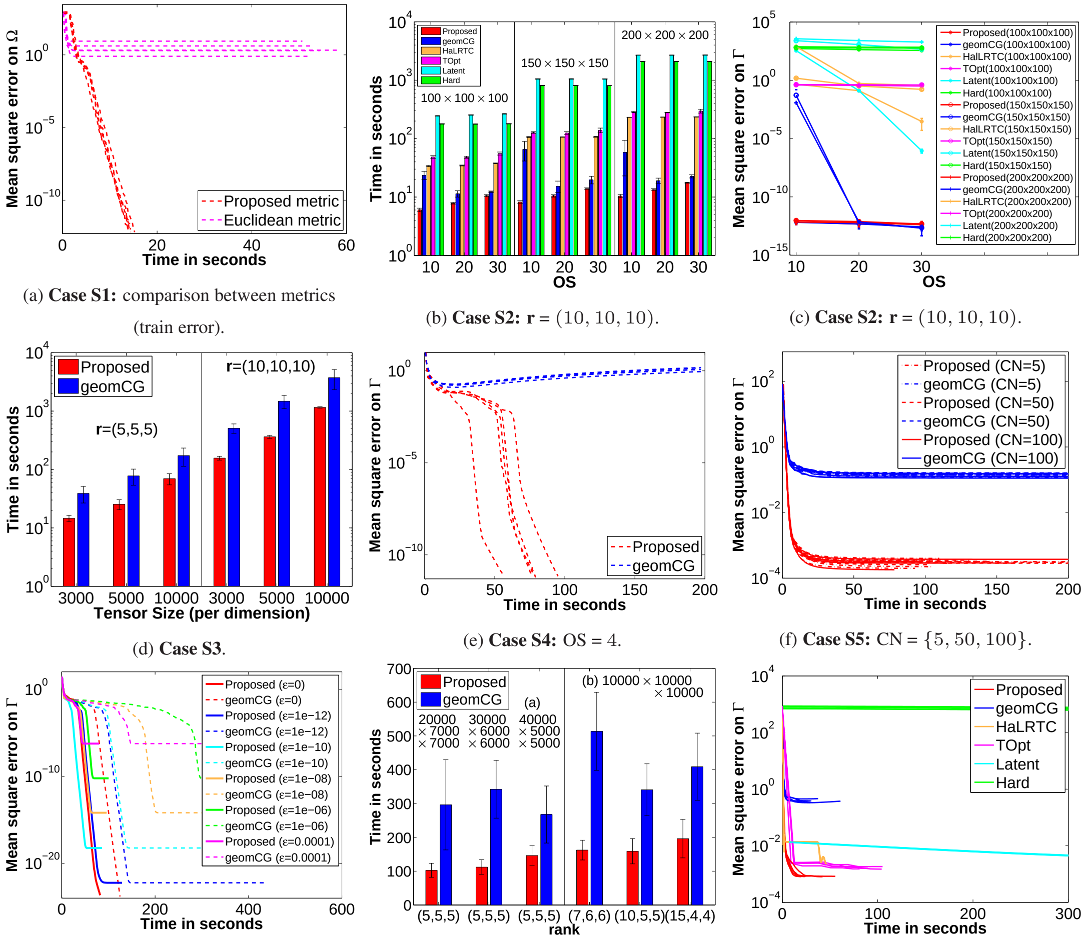
In the batch setting, we show a number of numerical comparisons of our proposed conjugate gradient algorithm with state-of-the-art algorithms that include TOpt (Filipović & Jukić, 2013) and geomCG (Kressner et al., 2014), for comparisons with Tucker decomposition based algorithms, and HaLRTC (Liu et al., 2013), Latent (Tomioka et al., 2011), and Hard (Signoretto et al., 2014) as nuclear norm minimization algorithms. In the online setting, we compare our proposed stochastic gradient descent algorithm with CANDECOMP/PARAFAC based TeCPSGD (Mardani et al., 2015) and OLSTEC (Kasai, 2016). All simulations are performed in Matlab on a 2.6 GHz Intel Core i7 machine with 16 GB RAM. For specific operations with unfoldings of $\boldsymbol{\mathcal{S}}$, we use the mex interfaces for Matlab that are provided by the authors of geomCG. For large-scale instances, our algorithm is only compared with geomCG as others cannot handle them. Cases S and R are for batch instances, whereas Case O is for online instances.

Since the dimension of the space of a tensor $\in \mathbb{R}^{n_1 \times n_2 \times n_3}$ of rank $\mathbf{r} = (r_1, r_2, r_3)$ is $\dim(\mathcal{M}/\sim) = \sum_{d=1}^3 (n_d r_d - r_d^2) + r_1 r_2 r_3$, we randomly and uniformly select known entries based on a multiple of the dimension, called the *over-sampling* (OS) ratio, to create the train set Ω. Algorithms are initialized randomly, as suggested in (Kressner et al., 2014), and are stopped when either the mean square error (MSE) on the train set Ω is below 10^{-12} or the number of iterations exceeds 250. We also evaluate the mean square error on a test set Γ, which is different from Ω. Five runs are performed in each scenario and the plots show all of them. The time plots are shown with standard deviations. It should be noted that we show most numerical comparisons on the *test set* Γ as it allows to compare with nuclear norm minimization algorithms, which optimize a different (training) cost function. Additional plots are provided as supplementary material.

Case S1: comparison with the Euclidean metric. We first show the benefit of the proposed metric (9) over the conventional choice of the Euclidean metric that exploits the product structure of $\mathcal{M}$ and symmetry (3). This is defined by combining the individual natural metrics for $\mathrm{St}(r_d, n_d)$ and $\mathbb{R}^{r_1 \times r_2 \times r_3}$. For simulations, we randomly generate a tensor of size $200 \times 200 \times 200$ and rank $\mathbf{r} = (10, 10, 10)$. OS is 10. For simplicity, we compare *gradient descent* algorithms with Armijo backtracking linesearch for both the metric choices. Figure 2(a) shows that the algorithm with the metric (9) gives a superior performance in *train error* than that of the conventional metric choice.

Case S2: small-scale instances. Small-scale tensors of size $100 \times 100 \times 100$, $150 \times 150 \times 150$, and $200 \times 200 \times 200$ and rank $\mathbf{r} = (10, 10, 10)$ are considered. OS is $\{10, 20, 30\}$. Figure 2(b) shows that our proposed algorithm has faster convergence than others. In Figure 2(c), the lowest test errors are obtained by our proposed algorithm and geomCG.

Case S3: large-scale instances. We consider large-scale tensors of size $3000 \times 3000 \times 3000$, $5000 \times 5000 \times 5000$, and $10000 \times 10000 \times 10000$ and ranks $\mathbf{r} = (5, 5, 5)$ and



(a) **Case S1:** comparison between metrics (train error).

(b) **Case S2:** $\mathbf{r} = (10, 10, 10)$.

(c) **Case S2:** $\mathbf{r} = (10, 10, 10)$.

(d) **Case S3**.

(e) **Case S4:** OS = 4.

(f) **Case S5:** CN = $\{5, 50, 100\}$.

(g) **Case S6:** noisy data.

(h) **Case S7:** rectangular tensors.

(i) **Case R1:** Ribeira, OS = 11.

Figure 2. Experiments on synthetic and real datasets.

$(10, 10, 10)$. OS is 10. Our proposed algorithm outperforms geomCG in Figure 2(d).

Case S4: influence of low sampling. We look into problem instances from scarcely sampled data, e.g., OS is 4. The test requires completing a tensor of size $10000 \times 10000 \times 10000$ and rank $\mathbf{r} = (5, 5, 5)$. Figure 2(e) shows the superior performance of the proposed algorithm against geomCG. Whereas the test error increases for geomCG, it decreases for the proposed algorithm.

Case S5: influence of ill-conditioning and low sampling. We consider the problem instance of **Case S4** with OS $= 5$. Additionally, for generating the instance, we impose a diagonal core $\boldsymbol{\mathcal{G}}$ with exponentially decaying *positive* values of condition numbers (CN) 5, 50, and 100. Figure 2(f) shows that the proposed algorithm outperforms geomCG for all the considered CN values.

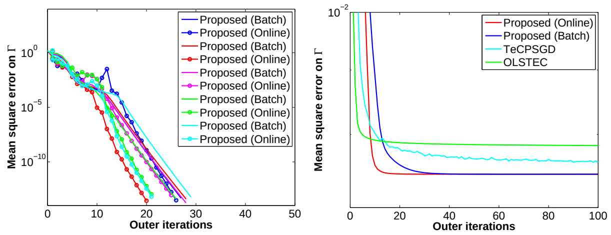
Case S6: influence of noise. We evaluate the convergence properties of algorithms under the presence of noise by adding *scaled* Gaussian noise $\mathcal{P}_\Omega(\boldsymbol{\mathcal{E}})$ to $\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}^\star)$ as in (Kressner et al., 2014). The different noise levels are $\epsilon = \{10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}\}$. In order to evaluate for $\epsilon = 10^{-12}$, the stopping threshold on the MSE of the train set is lowered to 10^{-24}. The tensor size and rank are same as in **Case S4** and OS is 10. Figure 2(g) shows that the test error for each ϵ is almost identical to the $\epsilon^2 \|\mathcal{P}_\Omega(\boldsymbol{\mathcal{X}}^\star)\|_F^2$ (Kressner et al., 2014), but our proposed algorithm converges faster than geomCG.

Case S7: rectangular instances. We consider instances where the dimensions and ranks along certain modes are different than others. Two cases are considered. Case (7.a) considers tensors size $20000 \times 7000 \times 7000$, $30000 \times 6000 \times 6000$, and $40000 \times 5000 \times 5000$ with rank $\mathbf{r} = (5, 5, 5)$. Case (7.b) considers a tensor of size $10000 \times 10000 \times 10000$ with ranks $(7, 6, 6)$, $(10, 5, 5)$, and $(15, 4, 4)$. In all the cases, the proposed algorithm converges faster than geomCG as shown in Figure 2(h).

Case R1: hyperspectral image. We consider the hyper-

Table 2. **Cases R1 and R2:** test MSE on Γ and time in seconds

Ribeira	OS = 11		OS = 22	
Algorithm	Time	MSE on Γ	Time	MSE on Γ
Proposed	$\mathbf{33 \pm 13}$	$\mathbf{8.2095 \cdot 10^{-4} \pm 1.7 \cdot 10^{-5}}$	67 ± 43	$\mathbf{6.9516 \cdot 10^{-4} \pm 1.1 \cdot 10^{-5}}$
geomCG	36 ± 14	$3.8342 \cdot 10^{-1} \pm 4.2 \cdot 10^{-2}$	150 ± 48	$6.2590 \cdot 10^{-3} \pm 4.5 \cdot 10^{-3}$
HaLRTC	46 ± 0	$2.2671 \cdot 10^{-3} \pm 3.6 \cdot 10^{-5}$	48 ± 0	$1.3880 \cdot 10^{-3} \pm 2.7 \cdot 10^{-5}$
TOpt	80 ± 32	$1.7854 \cdot 10^{-3} \pm 3.8 \cdot 10^{-4}$	$\mathbf{27 \pm 21}$	$2.1259 \cdot 10^{-3} \pm 3.8 \cdot 10^{-4}$
Latent	553 ± 3	$2.9296 \cdot 10^{-3} \pm 6.4 \cdot 10^{-5}$	558 ± 3	$1.6339 \cdot 10^{-3} \pm 2.3 \cdot 10^{-5}$
Hard	400 ± 5	$6.5090 \cdot 10^{2} \pm 6.1 \cdot 10^{1}$	402 ± 4	$6.5989 \cdot 10^{2} \pm 9.8 \cdot 10^{1}$
MovieLens-10M	Proposed		geomCG	
r	Time	MSE on Γ	Time	MSE on Γ
$(4, 4, 4)$	$\mathbf{1748 \pm 441}$	$\mathbf{0.6762 \pm 1.5 \cdot 10^{-3}}$	2981 ± 40	$0.6956 \pm 2.8 \cdot 10^{-3}$
$(6, 6, 6)$	$\mathbf{6058 \pm 47}$	$\mathbf{0.6913 \pm 3.3 \cdot 10^{-3}}$	6554 ± 655	$0.7398 \pm 7.1 \cdot 10^{-3}$
$(8, 8, 8)$	$\mathbf{11370 \pm 103}$	$\mathbf{0.7589 \pm 7.1 \cdot 10^{-3}}$	13853 ± 118	$0.8955 \pm 3.3 \cdot 10^{-2}$
$(10, 10, 10)$	$\mathbf{32802 \pm 52}$	$\mathbf{1.0107 \pm 2.7 \cdot 10^{-2}}$	38145 ± 36	$1.6550 \pm 8.7 \cdot 10^{-2}$



(a) **Case O:** synthetic dataset.

(b) **Case O:** Airport Hall dataset.

Figure 3. Experiments on online instances.

spectral image "Ribeira" (Foster et al., 2007) discussed in (Signoretto et al., 2011; Kressner et al., 2014). The tensor size is $1017 \times 1340 \times 33$, where each slice corresponds to a particular image measured at a different wavelength. As suggested in (Signoretto et al., 2011; Kressner et al., 2014), we resize it to $203 \times 268 \times 33$. We perform five random samplings of the pixels based on the OS values 11 and 22, corresponding to the rank **r**=$(15, 15, 6)$ adopted in (Kressner et al., 2014). This set is further randomly split into 80/10/10–train/validation/test partitions. The algorithms are stopped when the MSE on the validation set starts to increase. While OS $= 22$ corresponds to the observation ratio of 10% studied in (Kressner et al., 2014), OS $= 11$ considers a challenging scenario with the observation ratio of 5%. Figures 2(i) shows the good performance of our algorithm. Table 2 compiles the results.

Case R2: MovieLens-10M[1]. This dataset contains 10000054 ratings corresponding to 71567 users and 10681 movies. We split the time into 7-days wide bins results, and finally, get a tensor of size $71567 \times 10681 \times 731$. The fraction of known entries is less than 0.002%. The completion task on this dataset reveals *periodicity* of the *latent* genres. We perform five random 80/10/10–train/validation/test partitions. The maximum iteration threshold is set to 500. In Table 2, our proposed algorithm consistently gives lower test errors than geomCG across different ranks.

Case O: online instances. We compare the proposed stochastic gradient descent algorithm with its batch counterpart gradient descent algorithm and with TeCPSGD (Mardani et al., 2015) and OLSTEC (Kasai, 2016). As the implementations of TeCPSGD and OLSTEC are computationally more intensive than ours, our plots only show test MSE against the number of *outer iterations*, i.e., the number of the passes through the data.

Figure 3(a) shows comparisons on a synthetic instance of tensor size $100 \times 100 \times 10000$ with rank $\mathbf{r} = (5, 5, 5)$. γ_0 is selected from the step-size list $\{8, 9, 10, 11, 12\}$ in the pre-training phase. 10% entries are randomly observed. The pre-training uses 10% frontal slices of all the slices. The maximum number of outer loops is set to 100. Figure 3(a) shows five different runs, where the online algorithm has the same asymptotic convergence behavior as the batch counterpart on a test dataset Γ. Figure 3(b) shows comparisons on the Airport Hall surveillance video sequence dataset[2] of size 176×144 with 1000 frames. γ_0 is selected from $\{30, 40, 50, 60, 70\}$ and 10% frontal slices are selected for pre-training. 2% of the entries are observed. In Figure 3(b), both the proposed online and batch algorithms achieve lower test errors than TeCPSGD and OLSTEC.

6. Conclusion

We have proposed preconditioned batch (conjugate gradient) and online (stochastic gradient descent) algorithms for the tensor completion problem. The algorithms stem from the Riemannian preconditioning approach that exploits the fundamental structures of symmetry (due to non-uniqueness of Tucker decomposition) and least-squares of the cost function. A novel Riemannian metric (inner product) is proposed that enables to use the versatile Riemannian optimization framework. Numerical comparisons suggest that our proposed algorithms have a superior performance on different benchmarks.

[1] http://grouplens.org/datasets/movielens/.

[2] http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html

Acknowledgments

We thank Rodolphe Sepulchre, Paul Van Dooren, and Nicolas Boumal for useful discussions on the paper. This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. Hiroyuki Kasai is (partly) supported by the Ministry of Internal Affairs and Communications, Japan, as the SCOPE Project (150201002). This work was initiated while Bamdev Mishra was with the Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium and was visiting the Department of Engineering (Control Group), University of Cambridge, Cambridge, UK. He was supported as a research fellow (aspirant) of the Belgian National Fund for Scientific Research (FNRS).

References

Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Bonnabel, S. Stochastic gradient descent on Riemannian manifolds. *IEEE Trans. Autom. Control*, 58(9):2217–2229, 2013.

Bottou, L. Stochastic gradient descent tricks. *Neural Networks: Tricks of the Trade (2nd ed.)*, pp. 421–436, 2012.

Boumal, N. and Absil, P.-A. Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra Appl.*, 475:200–239, 2015.

Boumal, N., Mishra, B., Absil, P.-A., and Sepulchre, R. Manopt: a Matlab toolbox for optimization on manifolds. *J. Mach. Learn. Res.*, 15(1):1455–1459, 2014.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

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

Filipović, M. and Jukić, A. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidim. Syst. Sign. P.*, 2013. Doi: 10.1007/s11045-013-0269-9.

Foster, D. H., Nascimento, S. M. C., and Amano, K. Information limits on neural identification of colored surfaces in natural scenes. *Visual Neurosci.*, 21(3):331–336, 2007.

Kasai, H. Online low-rank tensor subspace tracking from incomplete data by CP decomposition using recursive least squares. In *IEEE ICASSP*, 2016.

Kolda, T. G. and Bader, B. W. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

Kressner, D., Steinlechner, M., and Vandereycken, B. Low-rank tensor completion by Riemannian optimization. *BIT Numer. Math.*, 54(2):447–468, 2014.

Lee, J. M. *Introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, second edition, 2003.

Liu, J., Musialski, P., Wonka, P., and Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):208–220, 2013.

Mardani, M., Mateos, G., and Giannakis, G.B. Subspace learning and imputation for streaming big data matrices and tensors. *IEEE Trans. Signal Process.*, 63(10):266 –2677, 2015.

Mishra, B. and Sepulchre, R. R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In *IEEE CDC*, pp. 1137–1142, 2014.

Mishra, B. and Sepulchre, R. Riemannian preconditioning. *SIAM J. Optim.*, 26(1):635–660, 2016.

Ngo, T. and Saad, Y. Scaled gradients on Grassmann manifolds for matrix completion. In *NIPS*, pp. 1421–1429, 2012.

Nocedal, J. and Wright, S. J. *Numerical Optimization*, volume Second Edition. Springer, 2006.

Ring, W. and Wirth, B. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM J. Optim.*, 22(2):596–627, 2012.

Sato, H. and Iwai, T. A new, globally convergent Riemannian conjugate gradient method. *Optimization*, 64(4): 1011–1031, 2015.

Signoretto, M., Plas, R. V. d., Moor, B. D., and Suykens, J. A. K. Tensor versus matrix completion: A comparison with application to spectral data. *IEEE Signal Process. Lett.*, 18(7):403–406, 2011.

Signoretto, M., Dinh, Q. T., Lathauwer, L. D., and Suykens, J. A. K. Learning with tensors: a framework based on convex optimization and spectral regularization. *Mach. Learn.*, 94(3):303–351, 2014.

Tomioka, R., Hayashi, K., and Kashima, H. Estimation of low-rank tensors via convex optimization. Technical report, arXiv preprint arXiv:1010.0789, 2011.

Vandereycken, B. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013.

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