53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

R3MC: A Riemannian three-factor algorithm for low-rank matrix completion*

Bamdev Mishra† and Rodolphe Sepulchre§†

***Abstract*— We exploit the versatile framework of Riemannian optimization on quotient manifolds to develop R3MC, a nonlinear conjugate-gradient method for low-rank matrix completion. The underlying search space of fixed-rank matrices is endowed with a novel Riemannian metric that is tailored to the least-squares cost. Numerical comparisons suggest that R3MC robustly outperforms state-of-the-art algorithms across different problem instances, especially those that combine scarcely sampled and ill-conditioned data.**

I. Introduction

We address the problem of low-rank matrix completion when the rank is a priori known or estimated. Given a rank-r matrix $\mathbf{X}^\star \in \mathbb{R}^{n\times m}$ whose entries $\mathbf{X}^\star_{ij}$ are only known for some indices $(i,j) \in \Omega$, where Ω is a subset of the complete set of indices $\{(i,j) : i \in \{1,\dots,n\} \text{ and } j \in \{1,\dots,m\}\}$, the *fixed-rank matrix completion problem* for rank r is formulated as

$$\begin{aligned} \min_{\mathbf{X}\in\mathbb{R}^{n\times n}} \quad & \tfrac{1}{|\Omega|}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 \\ \text{subject to} \quad & \operatorname{rank}(\mathbf{X}) = r, \end{aligned} \tag{1}$$

where the operator $\mathcal{P}_\Omega(\mathbf{X})_{ij} = \mathbf{X}_{ij}$ if $(i,j) \in \Omega$ and $\mathcal{P}_\Omega(\mathbf{X})_{ij} = 0$ otherwise, $|\Omega|$ is the cardinality of the Ω (equal to the number of known entries), and the norm $\|\cdot\|_F$ is the Frobenius norm. Problem (1) amounts to minimizing the data fitting error with the known entries in $\mathbf{X}^\star$ while constraining the rank to r. Not surprisingly, Problem (1) and its (many) variants find applications in system identification [1], control system [2], computer vision [3], machine learning [4], [5], to name a few.

The case of interest is when $r \ll \min(m,n)$, i.e., the rank is much smaller than the matrix dimensions. Solvability of (1) under the low-rank assumption has been studied in [6], [7], while a number of computationally efficient algorithms have been proposed in [4], [5], [7]–[15], among others.

The problem (1) has two fundamental structures. First, the *least-squares* structure of the cost function. Second, the fixed-rank constraint that has the structure of a *quotient matrix manifold* [13]. We denote the set of $n\times m$ matrices of rank r by $\mathbb{R}^{n\times m}_r$. A popular way to characterize fixed-rank matrices is through fixed-rank *matrix factorizations* [13]. However, most matrix factorizations have invariance properties that make the factorizations *non-unique*. And in many cases the set $\mathbb{R}^{n\times m}_r$ is identified with structured differentiable quotient manifolds [13]. Our focus in the present paper is on exploiting these particular structures in order to develop an efficient algorithm that scales to large-scale dimensional data by using manifold optimization techniques [16].

The manifold optimization framework of [16] naturally captures such *symmetries* in the search space and *submerses* them into an abstract quotient manifold, where non-uniqueness of the optimal solution disappears. In essence, the framework conceptually transforms the constrained optimization problem (1) into an *unconstrained* optimization problem on the quotient manifold of fixed-rank matrices [13], [16]. An essential step of the framework is to endow the search space with a *Riemannian metric*, a smoothly varying inner product. The metric structure plays a pivotal role in relating abstract notions on quotient manifolds to concrete matrix representations. Additionally, the performance of a gradient scheme on a manifold is *profoundly* dependent on the chosen metric [17, Chapter 3]. A potential limitation of the Riemannian framework is in identifying a suitable Riemannian metric [18]. We resolve this issue by proposing a Riemannian metric that is tailored to the cost function.

Out of many different works on low-rank matrix completion, we focus on the three recent algorithms [7], [14], [19] that have shown better performance in a number of challenging instances. Given a rank-r matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$, [7] exploits the fixed-rank factorization $\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T$ to embed the rank constraint, where $\mathbf{U}$ and $\mathbf{V}$ are column-orthonormal full-rank matrices of size $n\times r$ and $m\times r$, and $\mathbf{S} \in \mathbb{R}^{r\times r}$. At each iteration, [7] first updates $\mathbf{U}$ and $\mathbf{V}$ on the *bi-Grassmann* manifold $\mathrm{Gr}(r,n)\times\mathrm{Gr}(r,m)$, where $\mathrm{Gr}(r,n)$ is the set of r-dimensional subspaces in $\mathbb{R}^n$. Subsequently, a least-squares problem is solved to update $\mathbf{S}$. Building upon [7], Ngo and Saad [14] propose a *matrix scaling* on the bi-Grassmann manifold to *accelerate* the algorithm of [7]. In particular, Ngo and Saad [14] motivate the matrix scaling as an *adaptive preconditioner* for the optimization problem (1) and implement a conjugate-gradient algorithm. The same matrix scaling also appears in [19] where the authors motivate their *Gauss-Seidel* algorithm on the fixed-rank matrix factorization $\mathbf{X} = \mathbf{G}\mathbf{H}^T$, where $\mathbf{G}$ and $\mathbf{H}$ are full column-rank matrices of size $n\times r$ and $m\times r$, respectively. $\mathbf{G}$ and $\mathbf{H}$ are updated *alternatively*. A potential limitation of these algorithms is that they are alternating minimization and first-order algorithms, and extending them

*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. This work was also partly supported by the Belgian FRFC (Fonds de la Recherche Fondamentale Collective). Bamdev Mishra is a research fellow of the Belgian National Fund for Scientific Research (FNRS).

† Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liège, Belgium ({B.Mishra, R.Sepulchre}@ulg.ac.be).

§University of Cambridge, Department of Engineering, Trumpington Street, Cambridge CB2 1PZ, UK (R.Sepulchre@eng.cam.ac.uk).

to other classes of optimization methods is not trivial.

In order to get the best of these methods, we reinterpret the matrix scaling of [14], [19] as the outcome of a *specific* Riemannian metric construction and exploit it to propose a novel Riemannian geometry for $\mathbb{R}_r^{n\times r}$ and an algorithm for (1). The present work *follows and completes* the *unpublished* technical report [12], where we propose the general idea of tuning the Riemannian metric on a particular fixed-rank factorization. The metric so constructed provides a trade-off between second-order information of the cost function, symmetries in the search space, and simplicity of the computations involved. Building upon this work, we propose a three-factor *SVD-type* factorization, and exploit this for (1).

Our contribution is the nonlinear conjugate-gradient algorithm R3MC (Algorithm 1) that is based on a novel three-factor Riemannian geometry for matrix completion proposed in Section II. Although the new quotient geometry enables to propose second-order methods like the Riemannian trust-region method, here we specifically focus on conjugate-gradients as they offer an appropriate balance between convergence and computational cost. They have shown superior performance in our examples. The geometric notions to implement an off-the-shelf conjugate-gradient algorithm are listed in Section III. The concrete computations for the matrix completion problem are presented in Section IV. Finally in Section V, we illustrate the performance of R3MC across different problem instances, focusing in particular, on scarcely sampled and ill-conditioned problems. A Matlab implementation of R3MC is available on http://www.montefiore.ulg.ac.be/~mishra/codes/R3MC.html. The generic implementation of the algorithm for trust-regions is supplied with the Manopt package from http://manopt.org [20].

II. Three-factor fixed-rank matrix factorization

We consider a three-factor matrix factorization of rank-r matrix $\mathbf{X} \in \mathbb{R}_r^{n\times m}$, $\mathbf{X} = \mathbf{U}\mathbf{R}\mathbf{V}^T$, where $\mathbf{U} \in \mathrm{St}(r,n)$, $\mathbf{V} \in \mathrm{St}(r,m)$ and $\mathbf{R} \in \mathbb{R}_*^{r\times r}$. The Stiefel manifold $\mathrm{St}(r,n)$ is the set of matrices of size $n\times r$ with orthonormal columns and $\mathbb{R}_*^{r\times r}$ is the set of matrices of size $r\times r$ with non-zero determinant (full-rank square matrices). This factorization owes itself to the thin singular value decomposition (SVD). The difference with respect to SVD is that for the factor $\mathbf{R}$ we relax the diagonal constraint to accommodate any full-rank square matrix.

Consequently, the optimization problem (1) is a problem in $(\mathbf{U},\mathbf{R},\mathbf{V}) \in \mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$. However, the critical points in the space $\mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$ are not isolated as we have the symmetry

$$\mathbf{X} = \mathbf{U}\mathbf{R}\mathbf{V}^T = (\mathbf{U}\mathbf{O}_1)\mathbf{O}_1^T\mathbf{R}\mathbf{O}_2(\mathbf{V}\mathbf{O}_2), \tag{2}$$

for all $\mathbf{O}_1,\mathbf{O}_2 \in \mathcal{O}(r)$, the set of orthogonal matrices of size $r\times r$. In other words, the matrix $\mathbf{X} \in \mathbb{R}_r^{n\times m}$ (and the cost function) remains unchanged under the group action (2). Factoring out this symmetry, the problem (1) is *strictly* an optimization problem on the set of *equivalence classes* $[(\mathbf{U},\mathbf{R},\mathbf{V})] = \{(\mathbf{U}\mathbf{O}_1,\mathbf{O}_1^T\mathbf{R}\mathbf{O}_2,\mathbf{V}\mathbf{O}_2) : (\mathbf{O}_1,\mathbf{O}_2) \in \mathcal{O}(r)\times\mathcal{O}(r)\}$ rather than on $\mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$. Defining $\overline{\mathcal{M}} := \mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$, the set of equivalence classes is denoted as $\mathcal{M} := \overline{\mathcal{M}}/(\mathcal{O}(r)\times\mathcal{O}(r))$, where $\overline{\mathcal{M}}$ is called the *total space* (computational space) that is the product space $\mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$. The set $\mathcal{O}(r)\times\mathcal{O}(r)$ is the called the *fiber space*. The set of equivalence classes $\mathcal{M}$ has the structure of a quotient matrix manifold [16]. Equivalently, the set $\mathbb{R}_r^{n\times m}$ of $n\times m$ rank-r matrices is identified with $\mathcal{M}$, i.e., $\mathbb{R}_r^{n\times m} \simeq \mathcal{M}$.

Our main interest in studying the three-factor factorization is that it separates the *scaling* and the *subspace* information of a matrix as in the SVD factorization. The scaling information of $\mathbf{X}$ is in $\mathbf{R}$ whereas the subspace information is contained in $(\mathbf{U},\mathbf{V})$. This separation of scaling and subspaces leads to a robust behavior in numerical comparisons as shown later in Section V. Also since the fiber space is $\mathcal{O}(r)\times\mathcal{O}(r)$, by fixing $\mathbf{R}$ the search space is precisely the bi-Grassmann manifold $\mathrm{Gr}(r,n)\times\mathrm{Gr}(r,m)$ that comes into play in [7], where $\mathrm{Gr}(r,n) := \mathrm{St}(r,m)/\mathcal{O}(r)$ [16]. In other words, the three-factor geometry proposed in this paper generalizes the bi-Grassmann geometry of [7], [14].

A novel Riemannian metric. We represent an element of the quotient space $\mathcal{M}$ by x (each element is an equivalence class) and its matrix representation in the total space $\overline{\mathcal{M}} := \mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$ is given by $\bar{x} = (\mathbf{U},\mathbf{R},\mathbf{V})$. Equivalently, $x = [\bar{x}] = [(\mathbf{U},\mathbf{R},\mathbf{V})]$. The tangent space $T_{\bar{x}}\overline{\mathcal{M}}$ of the total space at $\bar{x} \in \overline{\mathcal{M}}$ is the product space of the tangent spaces of the individual manifolds. From [16, Example 3.5.2] we have the matrix characterization $T_{\bar{x}}\overline{\mathcal{M}} = \{(\mathbf{Z}_\mathbf{U},\mathbf{Z}_\mathbf{R},\mathbf{Z}_\mathbf{V}) \in \mathbb{R}^{n\times r}\times\mathbb{R}^{r\times r}\times\mathbb{R}^{m\times r} : \mathbf{U}^T\mathbf{Z}_\mathbf{U} + \mathbf{Z}_\mathbf{U}^T\mathbf{U} = \mathbf{0}, \mathbf{V}^T\mathbf{Z}_\mathbf{V} + \mathbf{Z}_\mathbf{V}^T\mathbf{V} = \mathbf{0}\}$.

The abstract quotient search space $\mathcal{M}$ is given the structure of a Riemannian quotient manifold by choosing a Riemannian metric in $\overline{\mathcal{M}}$ [16]. The metric defines an inner product between tangent vectors on the tangent space of $\overline{\mathcal{M}}$. Because the total space is a product space of well-studied manifolds, a typical metric on the total space is derived from choosing the natural metric of the product space, e.g., combining the natural metrics for $\mathrm{St}(r,n)$ and $\mathbb{R}_*^{r\times r}$ [16, Example 3.6.4]. However, this is not the only choice. Here we derive a different metric from the Hessian information of the cost function. The full Hessian information of (1) is computationally costly [21]. To circumvent the issue, we consider a simplified (but related) version of the cost function in (1). Specifically, consider the least-squares cost function $\|\mathbf{X}-\mathbf{X}^\star\|_F^2$ that is a simplification of the cost function in (1) by assuming that Ω contains the full set of indices. It should be stressed that the cost function $\|\mathbf{X}-\mathbf{X}^\star\|_F^2$ is *convex and quadratic* in $\mathbf{X}$, therefore, also convex and quadratic in each of the arguments $(\mathbf{U},\mathbf{R},\mathbf{V})$ *individually*. Additionally, the orthogonality constraint $\mathrm{St}(r,n)$ is also quadratic. This particular structure, *individually-convex quadratic optimization on quadratic constraints*, plays a critical role. For example, the second-order information of the cost function becomes a relevant source of second-order information of the optimization problem [17, Chapter 18]. Specifically, the *block approximation* of the Hessian of $\|\mathbf{U}\mathbf{R}\mathbf{V}^T-\mathbf{X}^\star\|_F^2$ has

the matrix characterization $((\mathbf{R}\mathbf{R}^T)\otimes\mathbf{I}_n, \mathbf{I}_r, (\mathbf{R}^T\mathbf{R})\otimes\mathbf{I}_m)$, where $\mathbf{I}_n$ is an $n\times n$ identity matrix, $\otimes$ is the Kronecker product of matrices, and $(\mathbf{U},\mathbf{R},\mathbf{V}) \in \overline{\mathcal{M}}$. The block approximation, equivalently, induces the metric

$$\begin{aligned}\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}},\bar{\eta}_{\bar{x}}) &= \mathrm{Trace}((\mathbf{R}\mathbf{R}^T)\bar{\xi}_{\mathbf{U}}^T\bar{\eta}_{\mathbf{U}}) + \mathrm{Trace}(\bar{\xi}_{\mathbf{R}}^T\bar{\eta}_{\mathbf{R}}) \\ &+ \mathrm{Trace}((\mathbf{R}^T\mathbf{R})\bar{\xi}_{\mathbf{V}}^T\bar{\eta}_{\mathbf{V}})\end{aligned} \tag{3}$$

on $T_{\bar{x}}\overline{\mathcal{M}}$, where $\bar{\xi}_{\bar{x}},\bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{M}}$ are tangent vectors with matrix characterizations $(\bar{\xi}_{\mathbf{U}},\bar{\xi}_{\mathbf{R}},\bar{\xi}_{\mathbf{V}})$ and $(\bar{\eta}_{\mathbf{U}},\bar{\eta}_{\mathbf{R}},\bar{\eta}_{\mathbf{V}})$, respectively. Since the function $\|\mathbf{U}\mathbf{R}\mathbf{V}^T-\mathbf{X}^\star\|_F^2$ is constant under the action (2), the $\bar{g}_{\bar{x}}$ induced from the block approximation of the Hessian of $\|\mathbf{U}\mathbf{R}\mathbf{V}^T-\mathbf{X}^\star\|_F^2$ respects the symmetry and is a valid Riemannian metric candidate on the total space $\overline{\mathcal{M}}$ [16, Section 3.6.2]. This is also equivalent to the metric proposed in [12] (for a different fixed-rank factorization).

III. Optimization-related ingredients

Once the metric is fixed on $\overline{\mathcal{M}} = \mathrm{St}(r,n)\times\mathbb{R}_*^{r\times r}\times\mathrm{St}(r,m)$, the concrete matrix representations necessary for implementing an off-the-shelf nonlinear conjugate-gradient algorithm for any smooth cost function on $\mathcal{M} = \overline{\mathcal{M}}/(\mathcal{O}(r)\times\mathcal{O}(r))$ are listed in this section. These are derived systematically using principles laid down in [16], the basis of which is the theory of *Riemannian submersion*.

Horizontal space. The matrix representation of the tangent space $T_x\mathcal{M}$ of the abstract quotient manifold $\mathcal{M} = \overline{\mathcal{M}}/(\mathcal{O}(r)\times\mathcal{O}(r))$ is identified with a subspace of the tangent space of the total space $T_{\bar{x}}\overline{\mathcal{M}}$ that does not produce a displacement along the equivalence classes. This subspace is called the *horizontal space* $\mathcal{H}_{\bar{x}}$ [16, Section 3.6.2]. In particular, we decompose $T_{\bar{x}}\overline{\mathcal{M}}$ into two *complementary subspaces* as $T_{\bar{x}}\overline{\mathcal{M}} = \mathcal{H}_{\bar{x}}\oplus\mathcal{V}_{\bar{x}}$. The *vertical space* $\mathcal{V}_{\bar{x}}$ is the tangent space to the equivalence class $[\bar{x}]$ and has the matrix expression $(\mathbf{U}\mathbf{\Omega}_1, \mathbf{R}\mathbf{\Omega}_2 - \mathbf{\Omega}_1\mathbf{R}, \mathbf{V}\mathbf{\Omega}_2)$, where $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are any skew-symmetric matrices of size $r\times r$ [16, Example 3.5.3]. Its complementary subspace with respect to the metric $\bar{g}_{\bar{x}}$ is chosen as the horizontal space $\mathcal{H}_{\bar{x}}$. The horizontal space provides a matrix characterization to the abstract tangent space $T_x\mathcal{M}$. After a routine computation, the subspace has the characterization $\mathcal{H}_{\bar{x}} = \{\bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{M}} : \mathbf{R}\mathbf{R}^T\mathbf{U}^T\bar{\eta}_{\mathbf{U}} + \bar{\eta}_{\mathbf{R}}\mathbf{R}^T \text{ and } \mathbf{R}^T\mathbf{R}\mathbf{V}^T\bar{\eta}_{\mathbf{V}} - \mathbf{R}^T\bar{\eta}_{\mathbf{R}} \text{ are symmetric}\}$. Any tangent vector $\xi_x \in T_x\mathcal{M}$ (on the quotient space) is uniquely represented by a vector $\bar{\xi}_{\bar{x}}$ in the horizontal space $\mathcal{H}_{\bar{x}}$, also called its *horizontal lift*.

Projection operators. We require two projection operators: one from the ambient space $\mathbb{R}^{n\times r}\times\mathbb{R}^{r\times r}\times\mathbb{R}^{m\times r}$ onto the tangent space $T_{\bar{x}}\overline{\mathcal{M}}$, and a second projection operator that further extracts the horizontal component of a tangent vector. Given a matrix in the space $\mathbb{R}^{n\times r}\times\mathbb{R}^{r\times r}\times\mathbb{R}^{m\times r}$, its projection onto the tangent space $T_{\bar{x}}\overline{\mathcal{M}}$ is obtained by extracting the component normal, in the metric sense, to the tangent space. The normal space $N_{\bar{x}}\overline{\mathcal{M}}$ is, thus, $\{(\mathbf{U}\mathbf{N}_1, \emptyset, \mathbf{V}\mathbf{N}_2) : \mathbf{N}_2\mathbf{R}\mathbf{R}^T \text{ and } \mathbf{N}_2\mathbf{R}^T\mathbf{R} \text{ are symmetric}\}$ for $\mathbf{N}_1,\mathbf{N}_2 \in \mathbb{R}^{r\times r}$. Extracting the tangential component is accomplished by using the linear operator $\Psi_{\bar{x}} : \mathbb{R}^{n\times r}\times\mathbb{R}^{r\times r}\times\mathbb{R}^{m\times r} \to T_{\bar{x}}\overline{\mathcal{M}} : (\mathbf{Z}_{\mathbf{U}},\mathbf{Z}_{\mathbf{R}},\mathbf{Z}_{\mathbf{V}}) \mapsto \Psi_{\bar{x}}(\mathbf{Z}_{\mathbf{U}},\mathbf{Z}_{\mathbf{R}},\mathbf{Z}_{\mathbf{V}})$, i.e.,

$$\begin{aligned}\Psi_{\bar{x}}(\mathbf{Z}_{\mathbf{U}},\mathbf{Z}_{\mathbf{R}},\mathbf{Z}_{\mathbf{V}}) &= (\mathbf{Z}_{\mathbf{U}} - \mathbf{U}\mathbf{B}_{\mathbf{U}}(\mathbf{R}\mathbf{R}^T)^{-1}, \mathbf{Z}_{\mathbf{R}}, \\ &\quad \mathbf{Z}_{\mathbf{V}} - \mathbf{V}\mathbf{B}_{\mathbf{V}}(\mathbf{R}^T\mathbf{R})^{-1}),\end{aligned} \tag{4}$$

where $\mathbf{B}_{\mathbf{U}}$ and $\mathbf{B}_{\mathbf{V}}$ are symmetric matrices of size $r\times r$ that are obtained by solving the Lyapunov equations $\mathbf{R}\mathbf{R}^T\mathbf{B}_{\mathbf{U}} + \mathbf{B}_{\mathbf{U}}\mathbf{R}\mathbf{R}^T = \mathbf{R}\mathbf{R}^T(\mathbf{U}^T\mathbf{Z}_{\mathbf{U}} + \mathbf{Z}_{\mathbf{U}}^T\mathbf{U})\mathbf{R}\mathbf{R}^T$ and $\mathbf{R}^T\mathbf{R}\mathbf{B}_{\mathbf{V}} + \mathbf{B}_{\mathbf{V}}\mathbf{R}^T\mathbf{R} = \mathbf{R}^T\mathbf{R}(\mathbf{V}^T\mathbf{Z}_{\mathbf{V}} + \mathbf{Z}_{\mathbf{V}}^T\mathbf{V})\mathbf{R}^T\mathbf{R}$ which can be solved efficiently with a total cost $O(r^3)$.

A subsequent projection onto the horizontal space $\mathcal{H}_{\bar{x}}$ is obtained by the operator $\Pi_{\bar{x}} : T_{\bar{x}}\overline{\mathcal{M}} \to \mathcal{H}_{\bar{x}} : \bar{\xi}_{\bar{x}} \mapsto \Pi_{\bar{x}}(\bar{\xi}_{\bar{x}})$

$$\Pi_{\bar{x}}(\bar{\xi}_{\bar{x}}) = (\bar{\xi}_{\mathbf{U}} - \mathbf{U}\mathbf{\Omega}_1, \bar{\xi}_{\mathbf{R}} + \mathbf{\Omega}_1\mathbf{R} - \mathbf{R}\mathbf{\Omega}_2, \bar{\xi}_{\mathbf{V}} - \mathbf{V}\mathbf{\Omega}_2), \tag{5}$$

where $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are skew-symmetric matrices of size $r\times r$ that are obtained by solving the coupled system of Lyapunov equations that has the form

$$\begin{aligned}\mathbf{R}\mathbf{R}^T\mathbf{\Omega}_1 + \;& \mathbf{\Omega}_1\mathbf{R}\mathbf{R}^T - \mathbf{R}\mathbf{\Omega}_2\mathbf{R}^T \\ & = \mathrm{Skew}(\mathbf{U}^T\bar{\xi}_{\mathbf{U}}\mathbf{R}\mathbf{R}^T) + \mathrm{Skew}(\mathbf{R}\bar{\xi}_{\mathbf{R}}^T), \\ \mathbf{R}^T\mathbf{R}\mathbf{\Omega}_2 + \;& \mathbf{\Omega}_2\mathbf{R}^T\mathbf{R} - \mathbf{R}^T\mathbf{\Omega}_1\mathbf{R} \\ & = \mathrm{Skew}(\mathbf{V}^T\bar{\xi}_{\mathbf{V}}\mathbf{R}^T\mathbf{R}) + \mathrm{Skew}(\mathbf{R}^T\bar{\xi}_{\mathbf{R}}),\end{aligned} \tag{6}$$

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 equations (6) can also be solved efficiently (and in closed form) by diagonalizing $\mathbf{R}$ and performing similarity transforms on $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$. The computational cost is $O(r^3)$.

Retraction. A retraction is a mapping that maps vectors in the horizontal space to points on the search space $\overline{\mathcal{M}}$. Precise formulation is in [16, Definition 4.1.1]). It provides a natural way to move on the manifold along a horizontal direction. The product nature of the total space $\overline{\mathcal{M}}$ again allows to choose a retraction by simply combining the retractions on the individual manifolds [16, Example 4.1.3]

$$R_{\bar{x}}(\bar{\xi}_{\bar{x}}) = (\mathrm{uf}(\mathbf{U}+\bar{\xi}_{\mathbf{U}}), \mathbf{R}+\bar{\xi}_{\mathbf{R}}, \mathrm{uf}(\mathbf{V}+\bar{\xi}_{\mathbf{V}})), \tag{7}$$

where $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ and $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of a full column-rank matrix, i.e., $\mathrm{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1/2}$ which is computed efficiently. The computational cost of a retraction operation is $O(nr^2 + mr^2 + r^3)$.

Vector transport. A vector transport $\mathcal{T}_{\eta_x}\xi_x$ 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 $R_x(\eta_x)$ satisfying the first-order approximation of transporting a vector along the geodesic [16, Definition 8.1.1]. With the projection operators, defined earlier, the horizontal lift of the vector transport $\mathcal{T}_{\eta_x}\xi_x$ is

$$\overline{\mathcal{T}_{\eta_x}\xi_x} = \Pi_{R_{\bar{x}}(\bar{\eta}_{\bar{x}})}(\Psi_{R_{\bar{x}}(\bar{\eta}_{\bar{x}})}(\bar{\xi}_{\bar{x}})), \tag{8}$$

where $\bar{\eta}_{\bar{x}}$ and $\bar{\xi}_{\bar{x}}$ are the horizontal lifts of ξ_x and η_x; and $\Pi(\cdot)$ and $\Psi(\cdot)$ are the projection operators defined in (5) and (4), respectively. The computational cost of transporting a vector solely depends on projection operations, defined earlier, which cost $O(nr^2 + mr^2 + r^3)$.

IV. R3MC: Algorithmic details

In Section III, the matrix representations of various notions on $\mathcal{M} = \overline{\mathcal{M}}/(\mathcal{O}(r) \times \mathcal{O}(r))$ are presented. Here we specifically deal with (1) that is, equivalently, the problem

$$\min_{x \in \mathcal{M}} f(x),$$

where $f : \mathcal{M} \to \mathbb{R}$ is the restriction of the function $\bar{f} : \overline{\mathcal{M}} \to \mathbb{R} : \bar{x} \mapsto \bar{f}(\bar{x}) = \|\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2/|\Omega|$, $\bar{x} \in \overline{\mathcal{M}}$ has the matrix representation $(\mathbf{U}, \mathbf{R}, \mathbf{V}) \in \mathrm{St}(r,n) \times \mathbb{R}_*^{r \times r} \times \mathrm{St}(r,m)$. The steps of our nonlinear conjugate-gradient algorithm R3MC are shown in Algorithm 1. The computational cost per iteration of R3MC is $O(|\Omega|r + nr^2 + mr^2 + r^3)$, where $|\Omega|$ is the number of known entries. The convergence of a Riemannian conjugate-gradient algorithm follows from the analysis in [22]. Step 3 of Algorithm 1, in particular, ensures that the sequence $\{\bar{\eta}_i\}$, $\bar{\eta}_i \in \mathcal{H}_{\bar{x}_i}$ is *gradient-related* [16, Definition 4.2.1].

Algorithm 1 R3MC: a conjugate-gradient algorithm for (1)

Input: Given $\bar{x}_0 = (\mathbf{U}_0, \mathbf{R}_0, \mathbf{V}_0) \in \overline{\mathcal{M}}$ and $\bar{\eta}_0 = 0$.

1: Compute the gradient $\bar{\xi}_i = \overline{\mathrm{grad}}_{\bar{x}_i} \bar{f}$. ▷ (9)
2: Compute the conjugate direction by Polak-Ribière (PR+) $\bar{\eta}_i = -\bar{\xi}_i + \beta_i \Pi_{\bar{x}_i}(\Psi_{\bar{x}_i}(\bar{\eta}_{i-1}))$. ▷ (8) and [16, (8.28)]
3: Check whether $\bar{\eta}_i$ is a descent direction. If not, then $\bar{\eta}_i = -\overline{\mathrm{grad}}_{\bar{x}_i} \bar{f}$.
4: Determine an initial step s_i. ▷ Section IV
5: Retract along $\bar{\eta}_i$ to compute the new iterate $\bar{x}_{i+1}$. ▷ (7)
6: Repeat until convergence.

The gradient computation. We define an auxiliary sparse variable $\mathbf{S} = 2(\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star))/|\Omega|$ that is interpreted as the Euclidean gradient of $\bar{f}$ in the space $\mathbb{R}^{n \times m}$. The horizontal lift (matrix representation) of the Riemannian gradient $\mathrm{grad}_x f$ is characterized using the projection operator (4) [16, Section 3.6] as

$$\begin{aligned}\overline{\mathrm{grad}_x f} = (&\mathbf{S}\mathbf{V}\mathbf{R}^T(\mathbf{R}\mathbf{R}^T)^{-1} - \mathbf{U}\mathbf{B}_\mathbf{U}(\mathbf{R}\mathbf{R}^T)^{-1}, \mathbf{U}^T\mathbf{S}\mathbf{V}, \\ &\mathbf{S}^T\mathbf{U}\mathbf{R}(\mathbf{R}^T\mathbf{R})^{-1} - \mathbf{V}\mathbf{B}_\mathbf{V}(\mathbf{R}^T\mathbf{R})^{-1}),\end{aligned} \tag{9}$$

where $\mathbf{B}_\mathbf{U}$ and $\mathbf{B}_\mathbf{V}$ are the solutions to the Lyapunov equations $\mathbf{R}\mathbf{R}^T\mathbf{B}_\mathbf{U} + \mathbf{B}_\mathbf{U}\mathbf{R}\mathbf{R}^T = 2\mathrm{Sym}(\mathbf{R}\mathbf{R}^T\mathbf{U}^T\mathbf{S}\mathbf{V}\mathbf{R}^T)$ and $\mathbf{R}^T\mathbf{R}\mathbf{B}_\mathbf{V} + \mathbf{B}_\mathbf{V}\mathbf{R}^T\mathbf{R} = 2\mathrm{Sym}(\mathbf{R}^T\mathbf{R}\mathbf{V}^T\mathbf{S}^T\mathbf{U}\mathbf{R})$. Here $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\mathbf{D}) = (\mathbf{D} + \mathbf{D}^T)/2$. The total numerical cost of computing the Riemannian gradient is $O(|\Omega|r)$.

Initial guess for the step-size. The least-squares nature of the cost function in (1) also allows to compute a *linearized* step-size guess efficiently [12], [15]. Given a search direction $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$, we solve the following optimization problem to obtain a linearized step-size guess by considering a degree 2 polynomial approximation, i.e., $s_* = \arg\min_{s \in \mathbb{R}_+} \|\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T + s\bar{\xi}_\mathbf{U}\mathbf{R}\mathbf{V}^T + s\mathbf{U}\bar{\xi}_\mathbf{R}\mathbf{V}^T + s\mathbf{U}\mathbf{R}\bar{\xi}_\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ that has a closed form solution. Computing s_* incurs a numerical cost $O(|\Omega|r)$.

Rank updating. In many problems, a good rank of the solution is either not known a priori or the notion of numerical rank is too vague to define it precisely, e.g., matrices with exponential decay of singular values. In such instances, it makes sense to traverse through a number of ranks, and not just one, in a systematic manner while ensuring that the cost function of (1) decreases with each such rank update. One way is to use fixed-rank optimization in conjunction with a rank-update strategy [11], [23]. The rank-one update is based on the idea of moving along the dominant rank-one projection of the negative gradient in the space $\mathbb{R}^{n \times m}$. If $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ is a rank-r matrix, then the rank-one update corresponds to $(\mathbf{U}_+, \mathbf{R}_+, \mathbf{V}_+)$ such that $\mathbf{U}_+\mathbf{R}_+\mathbf{V}_+^T = \mathbf{U}\mathbf{R}\mathbf{V}^T - \sigma uv^T$ where $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are the unit-norm dominant left and right singular vectors of $\mathbf{S}$ and $\sigma > 0$ is the dominant singular value. The total computational cost is $O(|\Omega| + nr^2 + mr^2 + r^3)$.

V. Numerical comparisons

We compare R3MC with the state-of-the-art algorithms RTRMC [10], LMaFit [19], R2MC (also known as qGeomMC) [12], ScGrassMC [14], LRGeom [15], and Polar Factorization [5], [13]. For the last four algorithms, we use with their conjugate-gradient implementations. R3MC and Polar Factorization exploit accelerated step-size computation in Section IV. The choice of these algorithms as state-of-the-art rests on recent publications [5], [10], [13]–[15], [19] and references therein. Other problems formulations, e.g., with nuclear norm regularization [8], have not been discussed here as the main motivation of the present paper is to look specifically look at algorithms that exploit the search space of fixed-rank matrices. In this section, we first show the connection of R3MC to ScGrassMC [14] and then provide a comparison of the considered algorithms across different problem instances, including one on a real dataset.

All simulations are performed in Matlab on a 2.53 GHz Intel Core i5 machine with 4 GB of RAM. We use Matlab codes for all the considered algorithms. For each example, an $n \times m$ random matrix of rank r is generated as in [8]. Two matrices $\mathbf{A} \in \mathbb{R}^{n \times r}$ and $\mathbf{B} \in \mathbb{R}^{m \times r}$ are generated according to a Gaussian distribution with zero mean and unit standard deviation. The matrix product $\mathbf{A}\mathbf{B}^T$ gives a random rank-r matrix. A fraction of the entries are randomly removed with uniform probability. We use the over-sampling ratio (OS), $\mathrm{OS} = |\Omega|/(nr + mr - r^2)$, to quantify the fraction of known entries. The maximum number of iterations of all is set to 500 (for RTRMC, it is 200). The algorithms are initialized similarly and stopped when the cost function is below 10^{-20}.

Connection to ScGrassMC. The connection with ScGrassMC is apparent from the fact that it scales the Riemannian gradient on $\mathrm{Gr}(r,n) \times \mathrm{Gr}(r,m)$ by $((\mathbf{R}\mathbf{R}^T)^{-1}, (\mathbf{R}^T\mathbf{R})^{-1})$. It should be noted that this is the same scaling that is obtained in the gradient computation (9). Additionally, we have linear projections to respect the quotient nature of the search space of fixed-rank matrices. The difference with respect to ScGrassMC is on two fronts. First, we perform a simultaneous update of the variables $(\mathbf{U}, \mathbf{R}, \mathbf{V})$, while ScGrassMC alternates between updating $(\mathbf{U}, \mathbf{V})$ and $\mathbf{R}$. Second, while preconditioning is motivated

in [14] as a way to accelerate the algorithm of [7], we view it as a particular choice of the Riemannian metric, enabling to develop arbitrary unconstrained optimization algorithms.

Case (a): influence of over-sampling. We consider a moderate scale matrix of size 10000×10000 of rank 10. Different instances with different over-sampling have been considered. For larger values of OS, most of the algorithms perform similarly and show a nice behavior. With smaller OS ratios, the algorithms, however, perform very differently. In fact in Figure 1(a), for the case of $\mathrm{OS} = 2.1$ only R3MC and Polar Factorization [13] algorithms converged, pointing towards a robustness of the three-factor matrix factorizations.

Case (b): influence of conditioning. We consider matrices of size 5000×5000 of rank 10, $\mathrm{CN} = 3$, and impose an exponential decay of singular values. The ratio of the largest to the lowest singular value is known as the condition number (CN) of the matrix. At rank 10 the singular values with condition number 100 is obtained using the Matlab function `logspace(-2,0,10)`. The over-sampling ratio for these instances is 3. The matrix completion problem becomes challenging as the CN increases. Figure 1(b) shows a particular instance. In general, for smaller CN, R2MC, R3MC, and LRGeom perform better than the others. In instances of larger CN, R3MC outperforms the others.

Case (c): influence of low sampling and ill-conditioning. In this test, we look into problem instances which result from both scarcely sampled and ill-conditioned data. The test requires completing relatively large matrices of size 25000×25000 of rank 10 with different condition numbers and OS ratios. Figure 1(c) shows the good performance of R3MC.

Case (d): ill-conditioning and rank-one updates. We generate a random matrix of size 5000×5000 of rank 20 with exponentially decaying singular values so that the condition number is 10^{10} and OS is 2 (computed for rank 10). Figure 1(d) shows the recovery of a set of entries in Γ, that is different from Ω on which we optimize, when R3MC is used with the rank-one updating procedure of Section IV. Most fixed-rank algorithms show a better performance when combined with a rank incrementing strategy.

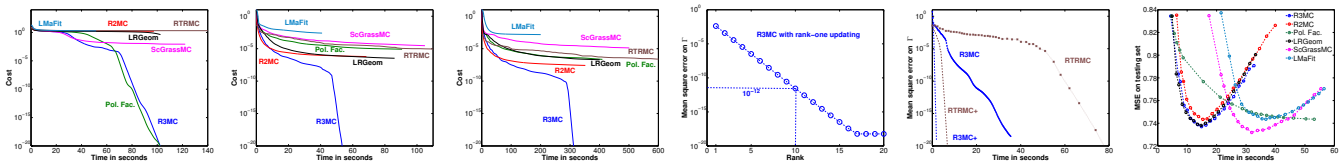
Case (e): rectangular matrices. Here we are particularly interested in instances with $n \ll m$, i.e., *rectangular* matrices. For these instances, most simulations suggest that the algorithm RTRMC of [10] performs numerically very efficiently. This is not surprising as the underlying geometry of RTRMC exploits the fact that the least-squares formulation (1) is solvable by fixing one of the fixed-rank factors.

To adapt the algorithms like R3MC (including RTRMC) to rectangular matrices under the standard assumptions for the matrix completion problem, we propose to deal with smaller size matrices with fewer columns. Consider a *truncated* submatrix of size $n \times p$ that picks all the rows of $\mathbf{X}^\star$ but picks only p (randomly chosen out of m) columns. A simple analysis shows that $\mathrm{OS}_{\mathrm{trunc.}} = \mathrm{OS}\alpha/(1+\alpha)$, where OS and $\mathrm{OS}_{\mathrm{trunc.}}$ are the over-sampling ratios for full and truncated problems, and $\alpha = p/n$. This means that the truncated problem is *challenging* for smaller values α. However for $\alpha > 1$, it is possible to have a competitive *trade-off* between difficulty and computational cost. It should, however, be noted that for a sufficiently large α both the original and the truncated matrices share the same *left subspace* [24]. Accordingly, we deal with the truncated problem of dimension $n \times p$ to compute the left subspace of the original matrix. Once the left subspace $\mathbf{U} \in \mathrm{St}(r, n)$ is identified, the weighting factor, e.g., the matrix $\mathbf{W} \in \mathbb{R}^{r \times n}$ of the factorization $\mathbf{X} = \mathbf{U}\mathbf{W}$ of [10] is obtained by solving a least-squares problem by fixing $\mathbf{U}$ [10]. A QR factorization of $\mathbf{W}$ results in the factors $\mathbf{R}$ and $\mathbf{V}$. The resulting $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ provides a good initialization to algorithms for the original problem.

We consider a rank-5 matrix of size 1000×50000 with $\mathrm{OS} = 5$ and $\mathrm{CN} = 10$. An incomplete submatrix of size 1000×2000 is formed by picking randomly 2000 columns. Consequently, $\alpha = 2$ and $\mathrm{OS}_{\mathrm{trunc.}} = 3.3$. The mean square errors on a set of entries Γ, different from Ω, are reported in Figure 1(e), where both R3MC and RTRMC with the proposed scheme (appended by the sign +) are *significantly* faster than their counterparts that deal with the full problem.

Case (f): MovieLens dataset. As a final test, we compare the algorithms on the MovieLens-1M dataset, downloaded from http://grouplens.org/datasets/movielens/. The dataset has a million ratings corresponding to 6040 users and 3952 movies. We perform 10 random 80/10/10-train/validation/test partitions of the ratings. The algorithms are run on the train set. The results are reported on the test set, averaged over 10 partitions. In RTRMC, the parameter λ is set to 10^{-6}, to avoid the error due to non-uniqueness of the least-squares solution that it uses. Finally, the maximum number of iterations is set to 1000 (200 for RTRMC). The algorithms are stopped before when the mean square error (MSE) on the validation set starts to increase. Table I (the second row) shows the MSEs on the test set with standard deviations $\pm 10^{-5}$ for different ranks. The best score of 0.7634 is obtained by R3MC at rank 6. Figure 1(f) shows the time taken by the algorithms, where R3MC, R2MC, and LRGeom are faster than the others. We also run all the algorithms with the rank-updating procedure of Section IV by traversing through all the ranks from 1 to 20. The rank is updated when the error on the validation set starts to increase. The last row of Table I compiles the *best* MSEs on the test set, where the optimal ranks so obtained are shown in brackets. RTRMC with rank-one updating did not give better results hence, is shown omitted. The best test score of 0.7323 is obtained by ScGrassMC at rank 10 followed by the score 0.7370 of R3MC at rank 9. However, R3MC is *twice* as fast as ScGrassMC.

Remarks. The case studies (a) to (f) are *challenging* instances of (1) as they combine ill-conditioning and low sampling in the data. Even though these studies are not fully exhaustive, they show a general trend of the performance of different algorithms. The conclusions drawn from each case study are based on a number of runs. Each figure, however, shows a *typical* instance. Similarly, even though convergence of the algorithms is shown to high accuracies, the conclusions drawn are equally valid for smaller accuracies. R3MC has shown faster and better convergence in all the examples.



(a) OS $= 2.1$ (b) CN $= 500$, OS $= 3$ (c) OS $= 4$, CN $= 500$ (d) Rank-one updating (e) Rectangular matrices (f) MovieLens dataset

Fig. 1. Performance of different algorithms under different scenarios. R3MC is particularly efficient in a number of instances.

TABLE I

MEAN SQUARE ERRORS OBTAINED ON THE TEST SET OF THE MOVIELENS-1M DATASET.

r	R3MC	R2MC	Pol. Fac.	ScGrassMC	LRGeom	LMaFit	RTRMC
3	0.7713	0.7771	0.7710	0.7967	0.7723	0.7762	0.7858
4	0.7677	0.7758	0.7675	0.7730	0.7689	0.7727	0.8022
5	0.7666	0.7781	0.7850	0.8280	0.7660	0.8224	0.8314
6	**0.7634**	0.7893	0.7651	0.7910	0.7698	0.8194	0.8802
7	0.7684	0.7996	0.7980	0.8368	0.7810	0.8074 (max. iters)	0.8241
With rank updates	0.7370 (9)	0.7434 (8)	0.7435 (20 max. rank)	**0.7323** (10)	0.7381 (9)	0.7435 (9)	-

VI. CONCLUSION

We have presented R3MC, an efficient Riemannian conjugate-gradient algorithm for the low-rank matrix completion problem. The algorithm stems from a novel Riemannian quotient geometry endowed with a tailored Riemannian metric on the set of fixed-rank matrices. Various numerical comparisons suggest a competitive performance of R3MC. At the conceptual level, the paper shows that the Riemannian optimization framework can take the advantage not only of the quotient structure of the search space of fixed-rank matrices, but also of the quadratic nature of the cost function. This viewpoint is further exploited in [25].

VII. ACKNOWLEDGMENT

We thank Paul Van Dooren, Bart Vandereycken, Nicolas Boumal, and Mariya Ishteva for useful discussions.

REFERENCES

[1] I. Markovsky, "Structured low-rank approximation and its applications," *Automatica*, vol. 44, no. 4, pp. 891–909, 2008.

[2] P. Benner and J. Saak, "Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: A state of the art survey," MPI Magdeburg, Tech. Rep., 2013.

[3] A. Joulin, F. Bach, and J. Ponce, "Discriminative clustering for image co-segmentation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1943–1950.

[4] J. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *International Conference on Machine learning (ICML)*, 2005, pp. 713–719.

[5] G. Meyer, S. Bonnabel, and R. Sepulchre, "Linear regression under fixed-rank constraints: a Riemannian approach," in *International Conference on Machine Learning (ICML)*, 2011, pp. 545–552.

[6] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[7] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2057–2078, 2010.

[8] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[9] P. Jain, R. Meka, and I. Dhillon, "Guaranteed rank minimization via singular value projection," in *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010, pp. 937–945.

[10] N. Boumal and P.-A. Absil, "RTRMC: A Riemannian trust-region method for low-rank matrix completion," in *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011, pp. 406–414.

[11] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre, "Low-rank optimization with trace norm penalty," *SIAM Journal on Optimization SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2124–2149, 2013.

[12] B. Mishra, K. Adithya Apuroop, and R. Sepulchre, "A Riemannian geometry for low-rank matrix completion," arXiv:1211.1550, Tech. Rep., 2012.

[13] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre, "Fixed-rank matrix factorizations and Riemannian low-rank optimization," *Computational Statistics*, vol. 29, no. 3–4, pp. 591–621, 2014.

[14] T. T. Ngo and Y. Saad, "Scaled gradients on Grassmann manifolds for matrix completion," in *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012, pp. 1421–1429.

[15] B. Vandereycken, "Low-rank matrix completion by Riemannian optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1214–1236, 2013.

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

[17] J. Nocedal and S. J. Wright, *Numerical Optimization, Second Edition*. New York, USA: Springer, 2006.

[18] J. H. Manton, "Optimization algorithms exploiting unitary constraints," *IEEE Transactions on Signal Processing*, vol. 20, no. 3, pp. 635–650, 2002.

[19] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012.

[20] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt: a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, no. Apr, pp. 1455–1459, 2014.

[21] A. Buchanan and A. Fitzgibbon, "Damped Newton algorithms for matrix factorization with missing data," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 316–322 vol. 2.

[22] W. Ring and B. Wirth, "Optimization methods on Riemannian manifolds and their application to shape space," *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 596–627, 2012.

[23] S. Burer and R. Monteiro, "A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization," *Mathematical Programming*, vol. 95, no. 2, pp. 329–357, 2003.

[24] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Annual Allerton Conference on Communication, Control, and Computing,*, 2010, pp. 704–711.

[25] B. Mishra and R. Sepulchre, "Riemannian preconditioning," arXiv:1405.6055, Tech. Rep., 2014.