Comput Stat (2014) 29:591–621
DOI 10.1007/s00180-013-0464-z

ORIGINAL PAPER

Fixed-rank matrix factorizations and Riemannian low-rank optimization

Bamdev Mishra · Gilles Meyer · Silvère Bonnabel · Rodolphe Sepulchre

Received: 31 August 2012 / Accepted: 28 October 2013 / Published online: 12 November 2013

Abstract Motivated by the problem of learning a linear regression model whose parameter is a large fixed-rank non-symmetric matrix, we consider the optimization of a smooth cost function defined on the set of fixed-rank matrices. We adopt the geometric framework of optimization on Riemannian quotient manifolds. We study the underlying geometries of several well-known fixed-rank matrix factorizations and then exploit the Riemannian quotient geometry of the search space in the design of a class of gradient descent and trust-region algorithms. The proposed algorithms generalize our previous results on fixed-rank symmetric positive semidefinite matrices, apply to a broad range of applications, scale to high-dimensional problems, and confer a geometric basis to recent contributions on the learning of fixed-rank non-symmetric matrices. We make connections with existing algorithms in the context of low-rank matrix completion and discuss the usefulness of the proposed framework. Numerical experiments suggest that the proposed algorithms compete with state-of-the-art algorithms and that manifold optimization offers an effective and versatile framework for the design of machine learning algorithms that learn a fixed-rank matrix.

B. Mishra (✉) · G. Meyer · R. Sepulchre
Department of Electrical Engineering and Computer Science, University of Liège, 4000 Liege, Belgium
e-mail: B.Mishra@ulg.ac.be

G. Meyer
e-mail: Gillesmy@gmail.com

R. Sepulchre
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK
e-mail: R.Sepulchre@eng.cam.ac.uk

S. Bonnabel
Robotics Center, Mines ParisTech, Boulevard Saint-Michel, 60, 75272 Paris, France
e-mail: Silvere.Bonnabel@mines-paristech.fr

Keywords Riemannian quotient geometry · Riemannian trust-region · Steepest descent · Low-rank matrix completion · Linear regression

1 Introduction

The problem of learning a low-rank matrix is a fundamental problem arising in many modern machine learning applications such as collaborative filtering (Rennie and Srebro 2005), classification with multiple classes (Amit et al. 2007), learning on pairs (Abernethy et al. 2009), dimensionality reduction (Cai et al. 2007), learning of low-rank distances (Kulis et al. 2009; Meyer et al. 2011b) and low-rank similarity measures (Shalit et al. 2010), multi-task learning (Evgeniou et al. 2005; Mishra et al. 2011a), to name a few. Parallel to the development of these new applications, the ever-growing size and number of large-scale datasets demands machine learning algorithms that can cope with large matrices. Scalability to high-dimensional problems is therefore a crucial issue in the design of algorithms that learn a low-rank matrix. Motivated by the above applications, the paper focuses on the following optimization problem

$$\min_{\mathbf{W}\in\mathbb{R}_r^{d_1\times d_2}} f(\mathbf{W}), \tag{1}$$

where $f : \mathbb{R}^{d_1\times d_2} \rightarrow \mathbb{R}$ is a smooth cost function and the search space is the set of fixed-rank non-symmetric real matrices,

$$\mathbb{R}_r^{d_1\times d_2} = \left\{\mathbf{W} \in \mathbb{R}^{d_1\times d_2} : \mathrm{rank}(\mathbf{W}) = r\right\}.$$

A particular case of interest is when $r \ll \min(d_1, d_2)$. In Sect. 2 we show that the considered optimization problem (1) encompasses various modern machine learning applications. We tackle problem (1) in a Riemannian framework, that is, by solving an unconstrained optimization on a Riemannian manifold in bijection with the nonlinear space $\mathbb{R}_r^{d_1\times d_2}$. This nonlinear space is an abstract space that is given the structure of a Riemannian quotient manifold in Sect. 4. The search space is motivated as a product space of well-studied manifolds which allows to derive the geometric notions in a straightforward and systematic way. Simultaneously, it ensures that we have *enough* flexibility in combining the different *pieces* together. One such flexibility is the choice of a Riemannian metric on the product space.

The paper follows and builds upon a number of recent contributions in that direction: the Ph.D. thesis Meyer (2011) and several papers by the authors, e.g., Journée (2009), Meyer et al. (2011a,b), Mishra et al. (2011a,b, 2012). The main contribution of this paper is to emphasize the common framework that underlines those contributions, with the aim of illustrating the versatile framework of Riemannian optimization for rank-constrained optimization. Necessary ingredients to perform both first-order and second-order optimization are listed for ready referencing. We discuss three popular fixed-rank matrix factorizations that embed the rank constraint. Two of these factorizations have been studied by Meyer et al. (2011a) and Mishra et al. (2011a). Exploiting the third factorization (the subspace-projection factorization in Sect. 3.3) in the Riemannian framework is a novel contribution of the present paper. An attempt is also made to classify the existing algorithms into various geometries and show the common structure that connects them all. The scalability of both first-order and second-order optimization algorithms to large dimensional problems is shown in Sect. 6.

The paper is organized as follows. Section 2 provides some concrete motivation for the proposed fixed-rank optimization problem. Section 3 reviews three classical fixed-rank matrix factorizations and introduces the quotient nature of the underlying search spaces. Section 4 develops the Riemannian quotient geometry of these three search spaces, providing all the concrete matrix operations required to code any first-order or second-order algorithm. Two basic algorithms are further detailed in Sect. 5. They underlie all numerical tests presented in Sect. 6.

2 Motivation and applications

In this section, a number of modern machine learning applications are cast as an optimization problem on the set of fixed-rank non-symmetric matrices.

2.1 Low-rank matrix completion

The problem of low-rank matrix completion amounts to estimating the missing entries of a matrix from a limited number of its entries. There has been a large number of research contributions on this subject over the last few years, addressing the problem both from a theoretical (Candès and Recht 2008; Gross 2011) and from an algorithmic point of view (Rennie and Srebro 2005; Meka et al. 2009; Cai et al. 2010; Lee and Bresler 2010; Keshavan et al. 2010; Simonsson and Eldén 2010; Boumal and Absil 2011; Jain et al. 2010; Mazumder et al. 2010; Ngo and Saad 2012). An important and popular application of the low-rank matrix completion problem is collaborative filtering (Rennie and Srebro 2005; Abernethy et al. 2009).

Let $\mathbf{W}^\star \in \mathbb{R}^{d_1 \times d_2}$ be a matrix whose entries $\mathbf{W}^\star_{ij}$ are only given for some indices $(i, j) \in \Omega$, where Ω is a subset of the complete set of indices $\{(i, j) : i \in \{1, \dots, d_1\}$ and $j \in \{1, \dots, d_2\}\}$. The fixed-rank matrix completion problem amounts to solving the following optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{W} \in \mathbb{R}^{d_1 \times d_2}} & \frac{1}{|\Omega|} \|\mathcal{P}_\Omega(\mathbf{W}) - \mathcal{P}_\Omega(\mathbf{W}^\star)\|_F^2 \\ \text{subject to} & \operatorname{rank}(\mathbf{W}) = r, \end{array} \tag{2}$$

where the function $\mathcal{P}_\Omega(\mathbf{W})_{ij} = \mathbf{W}_{ij}$ if $(i, j) \in \Omega$ and $\mathcal{P}_\Omega(\mathbf{W})_{ij} = 0$ otherwise and the norm $\|\cdot\|_F$ is *Frobenius* norm. $\mathcal{P}_\Omega$ is also called the *orthogonal sampling operator* and $|\Omega|$ is the cardinality of the set Ω (equal to the number of known entries).

The rank constraint captures redundant patterns in $\mathbf{W}^\star$ and ties the known and unknown entries together. The number of given entries $|\Omega|$ is of order $O(d_1 r + d_2 r - r^2)$ which is much smaller than $d_1 d_2$ (the total number of entries in $\mathbf{W}^*$) when $r \ll \min(d_1, d_2)$. Recent contributions provide conditions on $|\Omega|$ under which exact reconstruction is possible from entries sampled uniformly at random (Candès and Recht 2008; Keshavan et al. 2010). An application of this is in movie recommendations. The matrix to complete is a matrix of movie ratings of different users; a very sparse matrix with few ratings per user. The predictions of unknown ratings with a low-rank prior would have the interpretation that users' preferences only depend on few *genres* (Netflix 2006).

2.2 Learning on data pairs

The problem of learning on data pairs amounts to learning a predictive model $\hat{y} : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ from n training examples $\{(\mathbf{x}_i, \mathbf{z}_i, y_i)\}_{i=1}^n$ where data $\mathbf{x}_i$ and $\mathbf{z}_i$ are associated with two types of samples drawn from the set $\mathcal{X} \times \mathcal{Z}$ and $y_i \in \mathbb{R}$ is the associated scalar observation from the predictive model. If the predictive model is the bilinear form $\hat{y} = \mathbf{x}^T \mathbf{W} \mathbf{z}$ with $\mathbf{W} \in \mathbb{R}_r^{d_1 \times d_2}$, $\mathbf{x} \in \mathbb{R}^{d_1}$ and $\mathbf{z} \in \mathbb{R}^{d_2}$, then the problem boils down to the optimization problem,

$$\min_{\mathbf{W} \in \mathbb{R}_r^{d_1 \times d_2}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell \left(\mathbf{x}_i^T \mathbf{W} \mathbf{z}_i, y_i \right), \tag{3}$$

where the loss function ℓ penalizes the discrepancy between a scalar (experimental) observation y and the predicted value $\hat{y}$.

An application of this setup is the inference of edges in bipartite or directed graphs. Such problems arise in bioinformatics for the identification of interactions between drugs and target proteins, micro-RNA and genes or genes and diseases (Yamanishi et al. 2008; Bleakley and Yamanishi 2009). Another application is concerned with image domain adaptation (Kulis et al. 2011) where a transformation $\mathbf{x}^T \mathbf{W} \mathbf{z}$ is learned between labeled images $\mathbf{x}$ from a source domain $\mathcal{X}$ and labeled images $\mathbf{z}$ from a target domain $\mathcal{Z}$. The transformation $\mathbf{W}$ maps new input data from one domain to the other. A potential interest of the rank constraint in these applications is to address problems with a high-dimensional feature space and perform dimensionality reduction on the two data domains.

2.3 Multivariate linear regression

In multivariate linear regression, given the matrices $\mathbf{Y} \in \mathbb{R}^{n \times k}$ (output space) and $\mathbf{X} \in \mathbb{R}^{n \times q}$ (input space), we seek to learn a weight/coefficient matrix $\mathbf{W} \in \mathbb{R}_r^{q \times k}$ that minimizes the discrepancy between $\mathbf{Y}$ and $\mathbf{XW}$ (Yuan et al. 2007). Here n is the number of observations, q is the number of predictors and k is the number of responses.

One popular approach to multivariate linear regression problem is by minimizing a *quadratic loss* function. Note that in various applications *responses* are related and may therefore, be represented with much fewer coefficients (Yuan et al. 2007; Amit et al. 2007). This corresponds to finding the best low-rank matrix such that

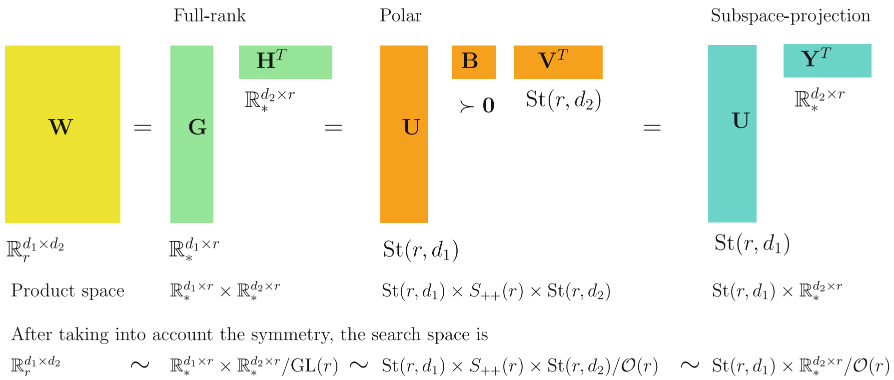


Fig. 1 Fixed-rank matrix factorizations lead to quotient search spaces due to their intrinsic *symmetries*. The pictures emphasize the situation of interest, i.e., the rank r is small compared to the matrix dimensions

$$\min_{\mathbf{W}\in\mathbb{R}_r^{q\times k}} \quad \|\mathbf{Y} - \mathbf{X}\mathbf{W}\|_F^2.$$

Though the quadratic loss function is shown here, the optimization setup extends to other smooth loss functions as well.

An application of this setup in financial econometrics is considered in Yuan et al. (2007) where the future returns of assets are estimated on the basis of their historical performance using the above formulation.

3 Matrix factorization and quotient spaces

A popular way to parameterize fixed-rank matrices is through matrix factorization. We review three popular matrix factorizations for fixed-rank non-symmetric matrices and study the underlying Riemannian geometries of the resulting search space (Fig. 1).

The three fixed-rank matrix factorizations of interest all arise from the thin singular value decomposition of a rank-r matrix $\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U}$ is a $d_1 \times r$ matrix with orthogonal columns, that is, an element of the Stiefel manifold $\mathrm{St}(r, d_1) = \{\mathbf{U} \in \mathbb{R}^{d_1\times r} : \mathbf{U}^T\mathbf{U} = \mathbf{I}\}$, $\Sigma \in \mathrm{Diag}_{++}(r)$ is a $r \times r$ diagonal matrix with positive entries and $\mathbf{V} \in \mathrm{St}(r, d_2)$. The singular value decomposition (SVD) exists for any matrix $\mathbf{W} \in \mathbb{R}_r^{d_1\times d_2}$ (Golub and Van Loan 1996, Sect. 2.5.3).

3.1 Full-rank factorization (beyond Cholesky-type decomposition)

The most popular low-rank factorization is obtained when the singular value decomposition (SVD) is rearranged as

$$\mathbf{W} = (\mathbf{U}\Sigma^{\frac{1}{2}})(\Sigma^{\frac{1}{2}}\mathbf{V}^T) = \mathbf{G}\mathbf{H}^T,$$

where $\mathbf{G} = \mathbf{U}\Sigma^{\frac{1}{2}} \in \mathbb{R}_*^{d_1 \times r}$, $\mathbf{H} = \mathbf{V}\Sigma^{\frac{1}{2}} \in \mathbb{R}_*^{d_2 \times r}$, and $\mathbb{R}_*^{d \times r}$ is the set of full column rank $d \times r$ matrices; also known as the *full-rank matrix factorization*. The resulting factorization is not unique because the transformation,

$$(\mathbf{G}, \mathbf{H}) \mapsto (\mathbf{G}\mathbf{M}^{-1}, \mathbf{H}\mathbf{M}^T), \tag{4}$$

where $\mathbf{M} \in \mathrm{GL}(r) = \{\mathbf{M} \in \mathbb{R}^{r \times r} : \det(\mathbf{M}) \neq 0\}$, leaves the original matrix $\mathbf{W}$ unchanged (Piziak and Odell 1999). This symmetry comes from the fact that the row and column spaces are invariant to change of coordinates. The classical remedy to remove this indeterminacy in the case of symmetric positive semidefinite matrices is the Cholesky factorization, which imposes further (triangular-like) structure in the factors (Golub and Van Loan 1996, Sect. 4.2). The LU decomposition plays a similar role for non-symmetric matrices (Jeffrey 2010). In a manifold setting, we instead encode the invariance map (4) in an abstract search space by optimizing over a set of equivalence classes defined as

$$[\mathbf{W}] = [(\mathbf{G}, \mathbf{H})] = \{(\mathbf{G}\mathbf{M}^{-1}, \mathbf{H}\mathbf{M}^T) : \mathbf{M} \in \mathrm{GL}(r)\}, \tag{5}$$

instead of the product space $\mathbb{R}_*^{d_1 \times r} \times \mathbb{R}_*^{d_2 \times r}$. The set of equivalence classes is denoted as

$$\mathcal{W} := \overline{\mathcal{W}}/\mathrm{GL}(r). \tag{6}$$

The product space $\mathbb{R}_*^{d_1 \times r} \times \mathbb{R}_*^{d_2 \times r}$ is called the *total space*, denoted by $\overline{\mathcal{W}}$. The set $\mathrm{GL}(r)$ is called the *fiber space*. The set of equivalence classes $\mathcal{W}$ is called the quotient space. In the next section it is given the structure of a Riemannian manifold over which optimization algorithms are developed.

3.2 Polar factorization (beyond SVD)

The second quotient structure for the set $\mathbb{R}_r^{d_1 \times d_2}$ is obtained by considering the following group action on the SVD (Bonnabel and Sepulchre 2009),

$$(\mathbf{U}, \Sigma, \mathbf{V}) \mapsto (\mathbf{U}\mathbf{O}, \mathbf{O}^T\Sigma\mathbf{O}, \mathbf{V}\mathbf{O}),$$

where $\mathbf{O}$ is any $r \times r$ orthogonal matrix, that is, any element of the set

$$\mathcal{O}(r) = \left\{\mathbf{O} \in \mathbb{R}^{r \times r} : \mathbf{O}^T\mathbf{O} = \mathbf{O}\mathbf{O}^T = \mathbf{I}\right\}.$$

This results in the *polar factorization*

$$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T,$$

where $\mathbf{U} \in \mathrm{St}(r, n)$ (the Stiefel manifold), $\mathbf{V} \in \mathrm{St}(r, m)$, and $\mathbf{B}$ is now a $r \times r$ symmetric positive definite matrix, that is, an element of

$$S_{++}(r) = \{\mathbf{B} \in \mathbb{R}^{r \times r} : \mathbf{B}^T = \mathbf{B} \succ 0\}. \tag{7}$$

The polar factorization reflects the original geometric purpose of singular value decomposition as representing an arbitrary linear transformation as the composition of two isometries and a scaling (Golub and Van Loan 1996, Sect. 2.5.3). Allowing the scaling $\mathbf{B}$ to be positive definite rather than diagonal gives more flexibility to optimization algorithms and removes the discrete symmetries induced by interchanging the order on the singular values. Empirical evidence to support the choice of $S_{++}(r)$ over $\mathrm{Diag}_{++}(r)$ (set of diagonal matrices with positive entries) for the middle factor $\mathbf{B}$ is shown in Sect. 6.3. The resulting search space is again the set of equivalence classes defined by

$$[\mathbf{W}] = [(\mathbf{U}, \mathbf{B}, \mathbf{V})] = \{(\mathbf{UO}, \mathbf{O}^T\mathbf{BO}, \mathbf{VO}) : \mathbf{O} \in \mathcal{O}(r)\}. \tag{8}$$

The total space is now $\overline{\mathcal{W}} = \mathrm{St}(r, d_1) \times S_{++}(r) \times \mathrm{St}(r, d_2)$. The fiber space is $\mathcal{O}(r)$ and the resulting quotient space is, thus, the set of equivalence classes

$$\mathcal{W} = \overline{\mathcal{W}}/\mathcal{O}(r). \tag{9}$$

3.3 Subspace-projection factorization (beyond QR decomposition)

The third low-rank factorization is obtained from the SVD when two factors are grouped together,

$$\mathbf{W} = \mathbf{U}(\Sigma\mathbf{V}^T) = \mathbf{UY}^T,$$

where $\mathbf{U} \in \mathrm{St}(r, d_1)$ and $\mathbf{Y} \in \mathbb{R}_*^{d_2 \times r}$ and is referred to as *subspace-projection* factorization. The column subspace of $\mathbf{W}$ matrix is represented by $\mathbf{U}$ while $\mathbf{Y}$ is the (left) *projection* or *coefficient* matrix of $\mathbf{W}$. The factorization is not unique as it is invariant with respect to the group action $(\mathbf{U}, \mathbf{Y}) \mapsto (\mathbf{UO}, \mathbf{YO})$, whenever $\mathbf{O} \in \mathcal{O}(r)$. The classical remedy to remove this indeterminacy is the QR factorization for which $\mathbf{Y}$ is chosen upper triangular (Golub and Van Loan 1996, Sect. 5.2). Here again we work with the set of equivalence classes

$$[\mathbf{W}] = [(\mathbf{U}, \mathbf{Y})] = \{(\mathbf{UO}, \mathbf{YO}) : \mathbf{O} \in \mathcal{O}(r)\}. \tag{10}$$

The search space is the quotient space

$$\mathcal{W} = \overline{\mathcal{W}}/\mathcal{O}(r), \tag{11}$$

where the total space is $\overline{\mathcal{W}} := \mathrm{St}(r, d_1) \times \mathbb{R}_*^{d_2 \times r}$ and the fiber space is $\mathcal{O}(r)$. Recent contributions exploiting this factorization include Simonsson and Eldén (2010), Dai et al. (2011), Boumal and Absil (2011).

Table 1 Fixed-rank matrix factorizations and their quotient manifold representations

	$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$
Matrix representation	$(\mathbf{G}, \mathbf{H})$	$(\mathbf{U}, \mathbf{B}, \mathbf{V})$	$(\mathbf{U}, \mathbf{Y})$
Total space $\overline{\mathcal{W}}$	$\mathbb{R}_*^{d_1 \times r} \times \mathbb{R}_*^{d_2 \times r}$	$\mathrm{St}(r, d_1) \times S_{++}(r) \times \mathrm{St}(r, d_2)$	$\mathrm{St}(r, d_1) \times \mathbb{R}_*^{d_2 \times r}$
Group action	$(\mathbf{G}\mathbf{M}^{-1}, \mathbf{H}\mathbf{M}^T)$	$(\mathbf{U}\mathbf{O}, \mathbf{O}^T\mathbf{B}\mathbf{O}, \mathbf{V}\mathbf{O})$	$(\mathbf{U}\mathbf{O}, \mathbf{Y}\mathbf{O})$
	$\mathbf{M} \in \mathrm{GL}(r)$	$\mathbf{O} \in \mathcal{O}(r)$	$\mathbf{O} \in \mathcal{O}(r)$
Quotient space $\mathcal{W}$	$\mathbb{R}_*^{d_1 \times r} \times \mathbb{R}_*^{d_2 \times r}/\mathrm{GL}(r)$	$\mathrm{St}(r, d_1) \times S_{++}(r) \times \mathrm{St}(r, d_2)/\mathcal{O}(r)$	$\mathrm{St}(r, d_1) \times \mathbb{R}_*^{d_2 \times r}/\mathcal{O}(r)$

The action of Lie groups $\mathrm{GL}(r)$ and $\mathcal{O}(r)$ make the quotient spaces, smooth quotient manifolds (Lee 2003, Theorem 9.16)

4 Fixed-rank matrix spaces as Riemannian submersions

The general philosophy of optimization on manifolds is to recast a constrained optimization problem in the Euclidean space $\mathbb{R}^n$ into an unconstrained optimization on a nonlinear search space that encodes the constraint. For special constraints that are sufficiently structured, the framework leads to an efficient computational framework (Absil et al. 2008). The three total spaces considered in the previous section all admit product structures of well-studied differentiable manifolds $\mathrm{St}(r, d_1)$, $\mathbb{R}_*^{d_1 \times r}$ and $S_{++}(r)$. Similarly, the fiber spaces are the Lie groups $\mathrm{GL}(r)$ and $\mathcal{O}(r)$. In this section, all the quotient spaces of the three fixed-rank factorizations are shown to have the differential structure of a Riemannian quotient manifold.

Each point on a quotient manifold represents an entire equivalence class of matrices in the total space. Abstract geometric objects on the quotient manifold can be defined by means of matrix representatives. Below we show the development of various geometric objects that are required to optimize a smooth cost function on the quotient manifold. Most of these notions follow directly from the work of Absil et al. (2008, Chapters 3 and 4). In the Tables 1, 2, 3, 4 and 5 we show the matrix representations of various geometric notions that are required to optimize a smooth cost function on a quotient manifold. More details about the full-rank factorization (Sect. 3.1) and the polar factorization (Sect. 3.2) may be found in Mishra et al. (2011a), Meyer (2011). The corresponding geometric notions for the subspace-projection factorization (Sect. 3.3) are new to the present paper but nevertheless, the development follows similar lines.

4.1 Quotient manifold representation

Consider a total space $\overline{\mathcal{W}}$ equipped with an equivalence relation $\sim$. The equivalence class of a given point $\bar{x} \in \overline{\mathcal{W}}$ is the set $[\bar{x}] = \{\bar{y} \in \overline{\mathcal{W}} : \bar{y} \sim \bar{x}\}$. The set $\mathcal{W}$ of all equivalence classes is the quotient manifold of $\overline{\mathcal{W}}$ by the equivalence relation $\sim$. The mapping $\pi : \overline{\mathcal{W}} \to \mathcal{W} : \bar{x} \mapsto [\bar{x}]$ is called the natural or canonical projection map. In Fig. 2, we have $\pi(\bar{x}) = \pi(\bar{y})$ if and only if $\bar{x} \sim \bar{y}$ and therefore, $[\bar{x}] = \pi^{-1}(\pi(\bar{x}))$. We represent an element of the quotient space $\mathcal{W}$ by $x = [\bar{x}]$ and its matrix representation in the total space $\overline{\mathcal{W}}$ by $\bar{x}$.

Table 2 Matrix representations of tangent vectors

	$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$
Tangent vectors in $\overline{\mathcal{W}}$	$(\bar{\xi}_{\mathbf{G}}, \bar{\xi}_{\mathbf{H}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$	$(\bar{\xi}_{\mathbf{U}}, \bar{\xi}_{\mathbf{B}}, \bar{\xi}_{\mathbf{V}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{d_2 \times r}$: $\mathbf{U}^T \bar{\xi}_{\mathbf{U}} + \bar{\xi}_{\mathbf{U}}^T \mathbf{U} = 0$, $\bar{\xi}_{\mathbf{B}}^T = \bar{\xi}_{\mathbf{B}}$, $\mathbf{V}^T \bar{\xi}_{\mathbf{V}} + \bar{\xi}_{\mathbf{V}}^T \mathbf{V} = 0$	$(\bar{\xi}_{\mathbf{U}}, \bar{\xi}_{\mathbf{Y}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$: $\mathbf{U}^T \bar{\xi}_{\mathbf{U}} + \bar{\xi}_{\mathbf{U}}^T \mathbf{U} = 0$
Metric $\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}})$	$\mathrm{Tr}((\mathbf{G}^T\mathbf{G})^{-1}\bar{\xi}_{\mathbf{G}}^T \bar{\eta}_{\mathbf{G}})$ $+\mathrm{Tr}((\mathbf{H}^T\mathbf{H})^{-1}\bar{\xi}_{\mathbf{H}}^T \bar{\eta}_{\mathbf{H}})$	$\mathrm{Tr}(\bar{\xi}_{\mathbf{U}}^T \bar{\eta}_{\mathbf{U}})$ $+\mathrm{Tr}(\mathbf{B}^{-1}\bar{\xi}_{\mathbf{B}}\mathbf{B}^{-1}\bar{\eta}_{\mathbf{B}})$ $+\mathrm{Tr}(\bar{\xi}_{\mathbf{V}}^T \bar{\eta}_{\mathbf{V}})$	$\mathrm{Tr}(\bar{\xi}_{\mathbf{U}}^T \bar{\eta}_{\mathbf{U}})$ $+\mathrm{Tr}((\mathbf{Y}^T\mathbf{Y})^{-1}\bar{\xi}_{\mathbf{Y}}^T \bar{\eta}_{\mathbf{Y}})$
Vertical tangent vectors	$(-\mathbf{G}\Lambda, \mathbf{H}\Lambda^T)$: $\mathbf{\Lambda} \in \mathbb{R}^{r \times r}$	$(\mathbf{U}\mathbf{\Omega}, \mathbf{B}\mathbf{\Omega} - \mathbf{\Omega}\mathbf{B}, \mathbf{V}\mathbf{\Omega})$: $\mathbf{\Omega} \in \mathbb{R}^{r \times r}$ $\mathbf{\Omega}^T = -\mathbf{\Omega}$	$(\mathbf{U}\mathbf{\Omega}, \mathbf{Y}\mathbf{\Omega})$: $\mathbf{\Omega} \in \mathbb{R}^{r \times r}$ $\mathbf{\Omega}^T = -\mathbf{\Omega}$
Horizontal tangent vectors	$(\bar{\zeta}_{\mathbf{G}}, \bar{\zeta}_{\mathbf{H}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$: $\bar{\zeta}_{\mathbf{G}}^T \mathbf{G}\mathbf{H}^T\mathbf{H} = \mathbf{G}^T\mathbf{G}\mathbf{H}^T \bar{\zeta}_{\mathbf{H}}$	$(\zeta_{\mathbf{U}}, \zeta_{\mathbf{B}}, \zeta_{\mathbf{V}}) \in T_{\bar{x}}\overline{\mathcal{W}}$: $(\zeta_{\mathbf{U}}^T\mathbf{U} + \mathbf{B}^{-1}\zeta_{\mathbf{B}} - \zeta_{\mathbf{B}}\mathbf{B}^{-1}$ $+\zeta_{\mathbf{V}}^T\mathbf{V})$ is symmetric	$(\zeta_{\mathbf{U}}, \zeta_{\mathbf{Y}}) \in T_{\bar{x}}\overline{\mathcal{W}}$: $\zeta_{\mathbf{U}}^T\mathbf{U} + (\mathbf{Y}^T\mathbf{Y})^{-1}\zeta_{\mathbf{Y}}^T\mathbf{Y}$ is symmetric

The tangent space $T_{\bar{x}}\overline{\mathcal{W}}$ in the total space is decomposed into orthogonal subspaces, the vertical space $\mathcal{V}_{\bar{x}}$ and the horizontal space $\mathcal{H}_{\bar{x}}$. The Riemannian metric is chosen by picking the natural metric for each of the spaces, $\mathbb{R}_*^{d_1 \times r}$ (Absil et al. 2008, Example 3.6.4), $\mathrm{St}(r, d_1)$ (Absil et al. 2008, Example 3.6.2), and $S_{++}(r)$ (Bhatia 2007, Sect. 6.1). The Riemannian metric $\bar{g}_{\bar{x}}$ makes the matrix representation of the abstract tangent space $T_x\mathcal{W}$ unique in terms of the horizontal space $\mathcal{H}_{\bar{x}}$

In Sect. 3 we see that the total spaces for the three fixed-rank matrix factorizations are in fact, different product spaces of the set of full column rank matrices $\mathbb{R}_*^{d_1 \times r}$, the set of matrices of size $d_1 \times r$ with orthonormal columns $\mathrm{St}(r, d_1)$ (Edelman et al. 1998), and the set of positive definite $r \times r$ matrices $S_{++}(r)$ (Bhatia 2007). Each of these manifolds is a smooth homogeneous space and their product structure preserves the smooth differentiability property (Absil et al. 2008, Sect. 3.1.6).

The quotient spaces of the three matrix factorizations are given by the equivalence relationships shown in (6), (9), and (11). The canonical projection π is, thus, obtained by the group action of Lie groups $\mathrm{GL}(r)$ and $\mathcal{O}(r)$, the fiber spaces of the fixed-rank matrix factorizations. Hence, by the direct application of Lee (2003, Theorem 9.16), the quotient spaces of the matrix factorizations have the structure of smooth quotient manifolds and the map π is a smooth submersion for each of the quotient spaces. Table 1 shows the matrix representations of different fixed-rank matrix factorizations considered earlier in Sect. 3. In the next section, each of the quotient spaces is given an additional structure of a Riemannian quotient manifold by choosing a proper *Riemannian metric*.

4.2 Tangent vector representation as horizontal lifts

Calculus on a manifold $\mathcal{W}$ is developed in the tangent space $T_x\mathcal{W}$, a vector space that can be considered as the linearization of the nonlinear space $\mathcal{W}$ at x. Since, the manifold $\mathcal{W}$ is an abstract space, the elements of its tangent space $T_x\mathcal{W}$ at $x \in \mathcal{W}$

Table 3 The matrix representations of the projection operators $\Psi_{\bar{x}}$ and $\Pi_{\bar{x}}$

	$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$
Matrix representation of the ambient space	$(\mathbf{Z_G}, \mathbf{Z_H}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$	$(\mathbf{Z_U}, \mathbf{Z_B}, \mathbf{Z_V}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{d_2 \times r}$	$(\mathbf{Z_U}, \mathbf{Z_Y}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$
		$\downarrow \Psi_{\bar{x}}$	
Projection onto $T_{\bar{x}}\overline{\mathcal{W}}$	$(\mathbf{Z_G}, \mathbf{Z_H})$	$(\mathbf{Z_U} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{Z_U}),$ $\mathrm{Sym}(\mathbf{Z_B}),$ $\mathbf{Z_V} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T\mathbf{Z_V}))$	$(\mathbf{Z_U} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{Z_U}),$ $\mathbf{Z_Y})$
		$\downarrow \Pi_{\bar{x}}$	
Projection of a tangent vector $\bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{W}}$ onto $\mathcal{H}_{\bar{x}}$	$(\bar{\eta}_{\mathbf{G}} + \mathbf{G}\boldsymbol{\Lambda},$ $\bar{\eta}_{\mathbf{H}} - \mathbf{H}\boldsymbol{\Lambda}^T)$ where $\boldsymbol{\Lambda}$ is the unique solution to the Lyapunov equation $\boldsymbol{\Lambda}^T(\mathbf{G}^T\mathbf{G})(\mathbf{H}^T\mathbf{H}) + (\mathbf{G}^T\mathbf{G})(\mathbf{H}^T\mathbf{H})\boldsymbol{\Lambda}^T = (\mathbf{G}^T\mathbf{G})\mathbf{H}^T\bar{\eta}_{\mathbf{H}} - \bar{\eta}_{\mathbf{G}}^T\mathbf{G}(\mathbf{H}^T\mathbf{H})$	$(\bar{\eta}_{\mathbf{U}} - \mathbf{U}\boldsymbol{\Omega},$ $\bar{\eta}_{\mathbf{B}} - (\mathbf{B}\boldsymbol{\Omega} - \boldsymbol{\Omega}\mathbf{B}),$ $\bar{\eta}_{\mathbf{V}} - \mathbf{V}\boldsymbol{\Omega})$ where $\boldsymbol{\Omega}$ is the unique solution to the Lyapunov equation $\boldsymbol{\Omega}\mathbf{B}^2 + \mathbf{B}^2\boldsymbol{\Omega} = \mathbf{B}(-\mathrm{Skew}(\mathbf{U}^T\bar{\eta}_{\mathbf{U}}) + 2\mathrm{Skew}(\mathbf{B}^{-1}\bar{\eta}_{\mathbf{B}}) - \mathrm{Skew}(\mathbf{V}^T\bar{\eta}_{\mathbf{V}}))\mathbf{B}$	$(\bar{\eta}_{\mathbf{U}} - \mathbf{U}\boldsymbol{\Omega},$ $\bar{\eta}_{\mathbf{Y}} - \mathbf{Y}\boldsymbol{\Omega})$ where $\boldsymbol{\Omega}$ is the unique solution to the Lyapunov equations $(\mathbf{Y}^T\mathbf{Y})\tilde{\boldsymbol{\Omega}} + \tilde{\boldsymbol{\Omega}}(\mathbf{Y}^T\mathbf{Y}) = +2\mathrm{Skew}((\bar{\eta}_{\mathbf{Y}}^T\mathbf{Y})(\mathbf{Y}^T\mathbf{Y})) - 2\mathrm{Skew}(\mathbf{Y}^T\mathbf{Y}\mathbf{U}^T\bar{\eta}_{\mathbf{U}}\mathbf{Y}^T\mathbf{Y})$ and $(\mathbf{Y}^T\mathbf{Y})\boldsymbol{\Omega} + \boldsymbol{\Omega}(\mathbf{Y}^T\mathbf{Y}) = \tilde{\boldsymbol{\Omega}}$

$\Psi_{\bar{x}}$ projects a matrix in the Euclidean space onto the tangent space $T_{\bar{x}}\overline{\mathcal{W}}$. $\Pi_{\bar{x}}$ extracts the horizontal component of a tangent vector $\bar{\xi}_{\bar{x}}$. Here the operators $\mathrm{Sym}(\cdot)$ and $\mathrm{Skew}(\cdot)$ extract the symmetric and skew-symmetric parts of a square matrix and are defined as $\mathrm{Sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^T)/2$ and $\mathrm{Skew}(\mathbf{A}) = (\mathbf{A} - \mathbf{A}^T)/2$ for any square matrix $\mathbf{A}$

Table 4 Retraction $R_{\bar{x}}(\cdot)$ maps a horizontal vector $\bar{\xi}_{\bar{x}}$ on the manifold $\overline{\mathcal{W}}$

	$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$
Retraction $R_{\bar{x}}(\bar{\xi}_{\bar{x}})$ that maps a horizontal vector $\bar{\xi}_{\bar{x}}$ onto $\overline{\mathcal{W}}$	$(\mathbf{G} + \bar{\xi}_{\mathbf{G}}, \mathbf{H} + \bar{\xi}_{\mathbf{H}})$	$(\mathrm{uf}(\mathbf{U} + \bar{\xi}_{\mathbf{U}}), \mathbf{B}^{\frac{1}{2}} \exp(\mathbf{B}^{-\frac{1}{2}} \bar{\xi}_{\mathbf{B}} \mathbf{B}^{-\frac{1}{2}}) \mathbf{B}^{\frac{1}{2}},$ $\mathrm{uf}(\mathbf{V} + \bar{\xi}_{\mathbf{V}}))$	$(\mathrm{uf}(\mathbf{U} + \bar{\xi}_{\mathbf{U}}), \mathbf{Y} + \bar{\xi}_{\mathbf{Y}})$

It provides a computationally efficient way to move on the manifold while approximating the geodesics. $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of a full column rank matrix $\mathbf{D}$, i.e., $\mathrm{uf}(\mathbf{D}) = \mathbf{D}(\mathbf{D}^T\mathbf{D})^{-1/2}$ and $\exp(\cdot)$ is the matrix exponential operator

call for a matrix representation in the total space $\overline{\mathcal{W}}$ at $\bar{x}$ that respects the equivalence relationship $\sim$. In other words, the matrix representation of $T_x\mathcal{W}$ should be restricted to the directions in the tangent space $T_{\bar{x}}\overline{\mathcal{W}}$ in the total space $\overline{\mathcal{W}}$ at $\bar{x}$ that do not induce a displacement along the equivalence class $[x]$.

On the other hand, the tangent space at $\bar{x}$ of the total space $\overline{\mathcal{W}}$ admits a product structure, similar to the product structure of the total space. Because the total space is

Table 5 The Riemannian gradient of the function $\bar{\phi}$ and the Riemannian connection at $\bar{x}$ in total space $\overline{\mathcal{W}}$

	$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$
Riemannian gradient $\mathrm{grad}_{\bar{x}}\bar{\phi}$	First compute the partial derivatives $(\bar{\phi}_{\mathbf{G}}, \bar{\phi}_{\mathbf{H}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$ and then perform the operation $(\bar{\phi}_{\mathbf{G}}\mathbf{G}^T\mathbf{G}, \bar{\phi}_{\mathbf{H}}\mathbf{H}^T\mathbf{H})$	First compute the partial derivatives $(\bar{\phi}_{\mathbf{U}}, \bar{\phi}_{\mathbf{B}}, \bar{\phi}_{\mathbf{V}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{d_2 \times r}$ and then perform the operation $(\bar{\phi}_{\mathbf{U}} - \mathbf{U}^T \mathrm{Sym}(\mathbf{U}^T\bar{\phi}_{\mathbf{U}}),$ $\mathbf{B}\mathrm{Sym}(\bar{\phi}_{\mathbf{B}})\mathbf{B}, \bar{\phi}_{\mathbf{V}} - \mathbf{V}^T$ $\mathrm{Sym}(\mathbf{V}^T\bar{\phi}_{\mathbf{V}}))$	First compute the partial derivatives $(\bar{\phi}_{\mathbf{U}}, \bar{\phi}_{\mathbf{Y}}) \in \mathbb{R}^{d_1 \times r} \times \mathbb{R}^{d_2 \times r}$ and then perform the operation $(\bar{\phi}_{\mathbf{U}} - \mathbf{U}^T \mathrm{Sym}(\mathbf{U}^T\bar{\phi}_{\mathbf{U}}),$ $\bar{\phi}_{\mathbf{Y}}\mathbf{Y}^T\mathbf{Y})$
Riemannian connection $\overline{\nabla}_{\bar{\xi}_{\bar{x}}}\bar{\eta}_{\bar{x}}$	$\Psi_{\bar{x}}(\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\xi}_{\bar{x}}] + (\mathbf{A}_{\mathbf{G}}, \mathbf{A}_{\mathbf{H}}))$ where $\mathbf{A}_{\mathbf{G}} =$ $-\bar{\eta}_{\mathbf{G}}(\mathbf{G}^T\mathbf{G})^{-1}\mathrm{Sym}(\mathbf{G}^T\bar{\xi}_{\mathbf{G}})$ $-\bar{\xi}_{\mathbf{G}}(\mathbf{G}^T\mathbf{G})^{-1}\mathrm{Sym}(\mathbf{G}^T\bar{\eta}_{\mathbf{G}})$ $+\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathrm{Sym}(\bar{\eta}_{\mathbf{G}}^T\bar{\xi}_{\mathbf{G}})$, $\mathbf{A}_{\mathbf{H}} =$ $-\bar{\eta}_{\mathbf{H}}(\mathbf{H}^T\mathbf{H})^{-1}\mathrm{Sym}(\mathbf{H}^T\bar{\xi}_{\mathbf{H}})$ $-\bar{\xi}_{\mathbf{H}}(\mathbf{H}^T\mathbf{H})^{-1}\mathrm{Sym}(\mathbf{H}^T\bar{\eta}_{\mathbf{H}})$ $+\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathrm{Sym}(\bar{\eta}_{\mathbf{H}}^T\bar{\xi}_{\mathbf{H}}))$	$\Psi_{\bar{x}}(\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\xi}_{\bar{x}}]$ $+ (\mathbf{A}_{\mathbf{U}}, \mathbf{A}_{\mathbf{B}}, \mathbf{A}_{\mathbf{V}}))$ where $\mathbf{A}_{\mathbf{U}} = -\bar{\xi}_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\bar{\eta}_{\mathbf{U}})$, $\mathbf{A}_{\mathbf{B}} = -\mathrm{Sym}(\xi_{\mathbf{B}}\mathbf{B}^{-1}\eta_{\mathbf{B}})$, $\mathbf{A}_{\mathbf{V}} = -\bar{\xi}_{\mathbf{V}}\mathrm{Sym}(\mathbf{V}^T\bar{\eta}_{\mathbf{V}})$	$\Psi_{\bar{x}}(\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\xi}_{\bar{x}}]$ $+ (\mathbf{A}_{\mathbf{U}}, \mathbf{A}_{\mathbf{Y}}))$ where $\mathbf{A}_{\mathbf{U}} = -\bar{\xi}_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\bar{\eta}_{\mathbf{U}})$, $\mathbf{A}_{\mathbf{Y}} =$ $-\bar{\eta}_{\mathbf{Y}}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathrm{Sym}(\mathbf{Y}^T\bar{\xi}_{\mathbf{Y}})$ $-\bar{\xi}_{\mathbf{Y}}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathrm{Sym}(\mathbf{Y}^T\bar{\eta}_{\mathbf{Y}})$ $+\mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathrm{Sym}(\bar{\eta}_{\mathbf{Y}}^T\bar{\xi}_{\mathbf{Y}})$

The matrix representations of their counterparts on the Riemannian quotient manifold $\mathcal{W}$ are given by (14) and (15). Here $\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\xi}_{\bar{x}}]$ is the standard Euclidean directional derivative of the vector field $\bar{\eta}_{\bar{x}}$ in the direction $\bar{\xi}_{\bar{x}}$, i.e., $\mathrm{D}\bar{\eta}_{\bar{x}}[\bar{\xi}_{\bar{x}}] = \lim_{t \to 0^+} \frac{\bar{\eta}_{\bar{x}+t\bar{\xi}_{\bar{x}}} - \bar{\eta}_{\bar{x}}}{t}$. The projection operator $\Psi_{\bar{x}}$ maps an arbitrary matrix in the Euclidean space to vectors in the tangent space $T_{\bar{x}}\overline{\mathcal{W}}$ and is defined in Table 3

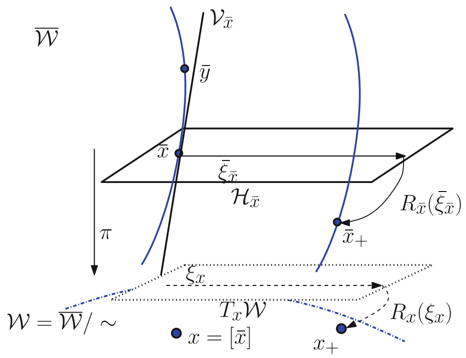


Fig. 2 Visualization of a Riemannian quotient manifold. The points $\bar{y}$ and $\bar{x}$ in the total space $\overline{\mathcal{W}}$ belong to the same equivalence class and they represent a single point $[x]$ in the quotient space $\mathcal{W}$. $\pi : \overline{\mathcal{W}} \rightarrow \mathcal{W}$ is a Riemannian submersion. The subspaces $\mathcal{V}_{\bar{x}}$ and $\mathcal{H}_{\bar{x}}$ are complementary spaces of $T_{\bar{x}}\overline{\mathcal{W}}$. The *horizontal space* $\mathcal{H}_{\bar{x}}$ provides a matrix representation to the abstract tangent space $T_x\mathcal{W}$ of the Riemannian quotient manifold. The mapping $R_{\bar{x}}$ maps a *horizontal vector* $\bar{\xi}_{\bar{x}}$ onto the total space. Here $\bar{\xi}_{\bar{x}}$ is the matrix representation of a tangent vector ξ_x on the abstract space

a product space of $\mathbb{R}_*^{d_1 \times r}$, $\mathrm{St}(r, d_1)$, and $S_{++}(r)$, its tangent space $T_{\bar{x}}\overline{\mathcal{W}}$ at $\bar{x}$ embodies the product space of the tangent spaces of $\mathbb{R}_*^{d_1 \times r}$, $\mathrm{St}(r, d_1)$, and $S_{++}(r)$, the characterizations of which are well-known. Refer Edelman et al. (1998, Sect. 2.2) or Absil et al. (2008, Example 3.5.2) for the characterization of the tangent space of $\mathrm{St}(r, d_1)$. Similarly, the tangent spaces of $\mathbb{R}_*^{d_1 \times r}$ and $S_{++}(r)$ are $\mathbb{R}^{d_1 \times r}$ (Euclidean space) and $S_{sym}(r)$ (the set of symmetric $r \times r$ matrices) respectively.

The matrix representation of a tangent vector at $x \in \mathcal{W}$ relies on the decomposition of $T_{\bar{x}}\overline{\mathcal{W}}$ into complementary subspaces, the *vertical* and *horizontal* subspaces such that $\mathcal{H}_{\bar{x}} \oplus \mathcal{V}_{\bar{x}} = T_{\bar{x}}\overline{\mathcal{W}}$. The vertical space $\mathcal{V}_{\bar{x}}$ is the tangent space of the equivalence class $T_{\bar{x}}\pi^{-1}(x)$. The horizontal space $\mathcal{H}_{\bar{x}}$, the complementary space of $\mathcal{V}_{\bar{x}}$, then provides a valid matrix representation of the abstract tangent space $T_x\mathcal{W}$ (Absil et al. 2008, Sect. 3.5.8). Once $T_{\bar{x}}\overline{\mathcal{W}}$ is endowed with a horizontal distribution $\mathcal{H}_{\bar{x}}$, a given tangent vector $\xi_x \in T_x\mathcal{W}$ at x on the quotient manifold $\mathcal{W}$ is uniquely represented by the tangent vector $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ in the total space $\overline{\mathcal{W}}$ that satisfies $\mathrm{D}\pi(\bar{x})[\bar{\xi}_{\bar{x}}] = \xi_x$ where $\mathrm{D}\pi(\bar{x})[\bar{\xi}_{\bar{x}}]$ is the directional derivative of $\pi(\bar{x})$ in the direction $\bar{\xi}_{\bar{x}}$. The tangent vector $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ is called the *horizontal lift* of ξ_x at $\bar{x}$. Refer to Fig. 2 for a graphical illustration.

A metric $\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\zeta}_{\bar{x}})$ on the total space defines a valid Riemannian metric g_x on the quotient manifold if

$$g_x(\xi_x, \zeta_x) := \bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\zeta}_{\bar{x}}) \tag{12}$$

where ξ_x and ζ_x are the tangent vectors in $T_x\mathcal{W}$ and $\bar{\xi}_{\bar{x}}$ and $\bar{\zeta}_{\bar{x}}$ are their horizontal lifts in $\mathcal{H}_{\bar{x}}$. The product structure of the total space $\overline{\mathcal{W}}$ again allows us to define a valid Riemannian metric by picking the natural metric for $\mathbb{R}_*^{d_1 \times r}$ (Absil et al. 2008, Example 3.6.4) $\mathrm{St}(r, d_1)$ (Absil et al. 2008, Example 3.6.2), and $S_{++}(r)$ (Bhatia 2007, Sect. 6.1). Endowed with this Riemannian metric, $\mathcal{W}$ is called a *Riemannian quotient manifold* of $\overline{\mathcal{W}}$ and the quotient map $\pi : \overline{\mathcal{W}} \rightarrow \mathcal{W} : \bar{x} \mapsto [\bar{x}]$ is a *Riemannian submersion*

(Absil et al. 2008, Sect. 3.6.2). It is customary to choose the horizontal space $\mathcal{H}_{\bar{x}}$ as the orthogonal complement of the vertical space $\mathcal{V}_{\bar{x}}$ according to the Riemannian metric. The matrix characterizations of $T_{\bar{x}}\overline{\mathcal{W}}$, $\mathcal{V}_{\bar{x}}$, and $\mathcal{H}_{\bar{x}}$, and the Riemannian metric $\bar{g}_{\bar{x}}$ for the three considered matrix factorizations are given in Table 2.

Table 3 summarizes the concrete matrix operations involved in computing horizontal vectors. Starting from an arbitrary matrix (with appropriate dimensions), two linear projections are needed: the first projection $\Psi_{\bar{x}}$ is onto the tangent space of the total space, while the second projection $\Pi_{\bar{x}}$ is onto the horizontal subspace. Note that the computational costs of all matrix operations are linear in the original matrix dimensions (d_1 or d_2). This is critical for the computational efficiency of the matrix algorithms.

4.3 Retractions from the tangent space to the manifold

An iterative optimization algorithm involves computing a (e.g., the gradient) search direction and then "moving in that direction". The default option on a Riemannian manifold is to move along geodesics, leading to the definition of the exponential map (see, e.g, Lee (2003, Chapter 20)). Because the calculation of the exponential map can be computationally demanding, it is customary in the context of manifold optimization to relax the constraint of moving along geodesics. The exponential map is then relaxed to a *retraction*, which is any map $R_{\bar{x}} : \mathcal{H}_{\bar{x}} \to \overline{\mathcal{W}}$ that locally approximates the exponential map on the manifold (Absil et al. 2008, Definition 4.1.1). A natural update on the manifold is, thus, based on the update formula

$$\bar{x}_+ = R_{\bar{x}}(\bar{\xi}_{\bar{x}}) \tag{13}$$

where $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ is a search direction and $\bar{x}_+ \in \overline{\mathcal{W}}$. See Fig. 2 for a graphical view. Due to the product structure of the total space, a retraction is obtained by combining the retraction updates on $\mathbb{R}_*^{d_1 \times r}$ (Absil et al. 2008, Example 4.1.5), $\mathrm{St}(r, d_1)$ (Absil et al. 2008, Example 4.1.3), and $S_{++}(r)$ (Bhatia 2007, Theorem 6.1.6). Note that the retraction on the positive definite cone is the exponential mapping with the natural metric (Bhatia 2007, Theorem 6.1.6). The cartesian product of the retractions also defines a valid retraction on the quotient manifold $\mathcal{W}$ (Absil et al. 2008, Proposition 4.1.3). The retractions for the fixed-rank matrix factorizations are presented in Table 4. The reader will notice that the matrix computations involved are again linear in the matrix dimensions d_1 and d_2.

4.4 Gradient and Hessian in Riemannian submersions

The choice of the metric (12), which is invariant along the equivalence class $[\bar{x}]$, and of the horizontal space (as the orthogonal complement of $\mathcal{V}_{\bar{x}}$ in the sense of the Riemannian metric) turns the quotient manifold $\mathcal{W}$ into a Riemannian submersion of $(\overline{\mathcal{W}}, \bar{g})$ (Absil et al. 2008, Sect. 3.6.2). As shown in Absil et al. (2008), this special construction allows for a convenient matrix representation of the gradient

(Absil et al. 2008, Sect. 3.6.2) and the Hessian (Absil et al. 2008, Proposition 5.3.3) on the abstract manifold $\mathcal{W}$.

Any smooth cost function $\bar{\phi} : \overline{\mathcal{W}} \to \mathbb{R}$ which is invariant along the fibers induces a corresponding smooth function ϕ on the quotient manifold $\mathcal{W}$. The Riemannian gradient of ϕ is uniquely represented by its horizontal lift in $\overline{\mathcal{W}}$ which has the matrix representation

$$\overline{\mathrm{grad}_x \phi} = \mathrm{grad}_{\bar{x}} \bar{\phi}. \tag{14}$$

It should be emphasized that $\mathrm{grad}_{\bar{x}} \bar{\phi}$ is in the tangent space $T_{\bar{x}}\overline{\mathcal{W}}$. However, due to invariance of the cost along the equivalence class $[\bar{x}]$, $\mathrm{grad}_{\bar{x}} \bar{\phi}$ also belongs to the horizontal space $\mathcal{H}_{\bar{x}}$ and hence, the equality in (14) (Absil et al. 2008, Sect. 3.6.2). The matrix expression of $\mathrm{grad}_{\bar{x}} \bar{\phi}$ in the total space $\overline{\mathcal{W}}$ at a point $\bar{x}$ is obtained from its definition: it is the unique element of $T_{\bar{x}}\overline{\mathcal{W}}$ that satisfies $\mathrm{D}\bar{\phi}[\eta_{\bar{x}}] = \bar{g}_{\bar{x}}(\mathrm{grad}_{\bar{x}} \bar{\phi}, \eta_{\bar{x}})$ for all $\eta_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{W}}$ (Absil et al. 2008, Eq. 3.31). $\mathrm{D}\bar{\phi}[\eta_{\bar{x}}]$ is the standard Euclidean directional derivative of $\bar{\phi}$ in the direction $\eta_{\bar{x}}$ and $\bar{g}_{\bar{x}}$ is the Riemannian metric. This definition leads to the matrix representations of the Riemannian gradient in Table 5.

In addition to the Riemannian gradient, any optimization algorithm that makes use of second-order information also requires the directional derivative of the gradient along a search direction. This involves the choice of an *affine connection* ∇ on the manifold. The affine connection provides a definition for the *covariant derivative* of vector field η_x with respect to the vector field ξ_x, denoted by $\nabla_{\xi_x} \eta_x$. Imposing an additional compatibility condition with the metric fixes the so-called *Riemannian* connection which is always unique (Absil et al. 2008, Theorem 5.3.1 and Sect. 5.2). The Riemannian connection $\nabla_{\xi_x} \eta_x$ on the quotient manifold $\mathcal{W}$ is uniquely represented in terms of the Riemannian connection in the total space $\overline{\mathcal{W}}$, $\overline{\nabla}_{\bar{\xi}_{\bar{x}}} \bar{\eta}_{\bar{x}}$ (Absil et al. 2008, Proposition 5.3.3) which is

$$\overline{\nabla_{\xi_x} \eta_x} = \Pi_{\bar{x}}(\overline{\nabla}_{\bar{\xi}_{\bar{x}}} \bar{\eta}_{\bar{x}}) \tag{15}$$

where ξ_x and η_x are vector fields in $\mathcal{W}$ and $\bar{\xi}_{\bar{x}}$ and $\bar{\eta}_{\bar{x}}$ are their horizontal lifts in $\overline{\mathcal{W}}$. Here $\Pi_{\bar{x}}$ is the projection operator that projects a tangent vector in $T_{\bar{x}}\overline{\mathcal{W}}$ onto the horizontal space $\mathcal{H}_{\bar{x}}$ as defined in Table 3. In this case as well, the Riemannian connection $\overline{\nabla}_{\bar{\xi}_{\bar{x}}} \bar{\eta}_{\bar{x}}$ on the total space $\overline{\mathcal{W}}$ has well-known expression owing to the product structure.

The Riemannian connection on the Stiefel manifold $\mathrm{St}(r, d_1)$ is derived in Journée (2009, Example 4.3.6). The Riemannian connection on $\mathbb{R}_*^{d_1 \times r}$ and on the set of positive definite matrices $S_{++}(r)$ with their natural metrics are derived in Meyer (2011, Appendix B). Finally, the Riemannian connection on the total space is given by the cartesian product of the individual connections. In Table 5 we give the final matrix expressions. The directional derivative of the Riemannian gradient in the direction ξ_x is called the Riemannian Hessian operator $\mathrm{Hess}_x \phi(x)[\xi_x]$ which is now directly given in terms of the Riemannian connection ∇. For the optimization algorithms in Sect. 5 we only require the Riemannian Hessian applied in a particular search direction (instead of the full Hessian) and this is given by $\mathrm{Hess}_x \phi(x)[\xi_x]$. The horizontal lift of the Riemannian Hessian in $\mathcal{W}$ has, thus, the following matrix expression

$$\overline{\mathrm{Hess}_x \phi(x)[\xi_x]} = \Pi_{\bar{x}}(\overline{\nabla}_{\bar{\xi}_{\bar{x}}} \overline{\mathrm{grad}_x \phi}) \tag{16}$$

for any $\xi_x \in T_x\mathcal{W}$ and its horizontal lift $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$.

5 Two optimization algorithms

For the sake of illustration, we consider two basic optimization schemes in this paper: the (steepest) gradient descent algorithm, as a representative of first-order algorithms, and the Riemannian trust-region scheme, as a representative of second-order algorithms. Both schemes can be easily implemented using the notions developed in the previous section. In particular, the Tables 3, 4 and 5 give all the necessary ingredients for optimizing a smooth cost function $\phi : \mathcal{W} \rightarrow \mathbb{R}$ on the Riemannian quotient manifold of fixed-rank matrix factorizations.

5.1 Gradient descent algorithm

For the gradient descent scheme we implement Absil et al. (2008, Algorithm 1) where at each iteration we move along the negative Riemannian gradient (see Table 5) direction by taking a step (13), and use the Armijo backtracking method of Nocedal and Wright (2006, Procedure 3.1) to compute an Armijo-optimal step-size satisfying the *sufficient decrease condition* (Nocedal and Wright 2006, Chapter 3). The Riemannian gradient is the gradient of the cost function in the sense of the Riemannian metric proposed in Table 2.

For computing an initial step-size, we use the information of the previous iteration by using the *adaptive step-size update* procedure proposed below. The adaptive step-size update procedure is different from the initial step-size procedure described in Nocedal and Wright (2006, Page 58). This procedure is independent of the cost function evaluation and can be considered as a zero-order *prediction heuristic*.

Let us assume that after the t^{th} iteration we know the initial step-size guess that was used $\hat{s}_t$, the Armijo-optimal step-size s_t and the number of backtracking line-searches j_t required to obtain the Armijo-optimal step-size s_t. The procedure is then,

$$\begin{array}{ll} \text{Given :} & \hat{s}_t \ \text{(initial step} - \text{size guess for iteration } t\text{)} \\ & j_t \ \text{(number of backtracking line} - \text{searches required at iteration } t\text{)} \\ & s_t \ \text{(Armijo} - \text{optimal step} - \text{size) at iteration } t \\ \text{Then :} & \text{the initial step} - \text{size guess at iteration } t+1 \\ & \text{is given by the update} \\ & \hat{s}_{t+1} = \begin{cases} 2\hat{s}_t, & j_t = 0 \\ 2s_t, & j_t = 1 \\ 2s_t, & j_t \geq 2. \end{cases} \end{array} \tag{17}$$

Here $s_0 (= \hat{s}_0)$ is the initial step-size guess provided by the user and $j_0 = 0$. This procedure keeps the number of line-searches close to 1 on average, that is, $\mathbb{E}_t(j_t) \approx 1$, assuming that the optimal step-size does not vary too much with iterations. An alternative is to choose any convex combination of the following updates:

$$\hat{s}_{t+1} = \begin{cases} \underline{\text{update 1}} & \underline{\text{update 2}} & \\ 2\hat{s}_t & 2\hat{s}_t, & j_t = 0 \\ 2s_t & 1s_t, & j_t = 1 \\ 1s_t & 2s_t, & j_t \geq 2. \end{cases}$$

5.2 Riemannian trust-region algorithm

The second optimization scheme we consider, is the Riemannian trust-region scheme. Analogous to trust-region algorithms in the Euclidean space (Nocedal and Wright 2006, Chapter 4), trust-region algorithms on a Riemannian quotient manifold with guaranteed quadratic rate convergence have been proposed in Absil et al. (2008, Chapter 7). Similar to the Euclidean case, at each iteration we solve the *trust-region sub-problem* on the quotient manifold $\mathcal{W}$. The trust-region sub-problem is formulated as the minimization of the locally-quadratic model of the cost function, say $\phi : \mathcal{W} \to \mathbb{R}$ at $x \in \mathcal{W}$

$$\begin{array}{ll} \min\limits_{\xi_x \in T_x \mathcal{W}} & \phi(x) + g_x(\xi_x, \mathrm{grad}_x \phi(x)) + \frac{1}{2} g_x(\xi_x, \mathrm{Hess}_x \phi(x)[\xi_x]) \\ \text{subject to} & g_x(\xi_x, \xi_x) \leq \Delta^2, \end{array} \tag{18}$$

where Δ is the trust-region radius, g_x is the Riemannian metric; and $\mathrm{grad}_x \phi$ and $\mathrm{Hess}_x \phi$ are the Riemannian gradient and Riemannian Hessian on the quotient manifold $\mathcal{W}$ (see Sect. 4.4 and Table 5). The Riemannian gradient is the gradient of the cost function in the sense of the Riemannian metric g_x and the Riemannian Hessian is given by the Riemannian connection. Computationally, the problem is horizontally lifted to the horizontal space $\mathcal{H}_{\bar{x}}$ (Absil et al. 2008, Sect. 7.2.2) where we have the matrix representations of the Riemannian gradient and Riemannian Hessian (Table 5). Solving the above trust-region sub-problem leads to a direction $\bar{\xi}$ that minimizes the quadratic model. Depending on whether the decrease of the cost function is sufficient or not, the potential iterate is accepted or rejected.

In particular, we implement the Riemannian trust-region algorithm (Absil et al., 2008, Algorithm 10) using the generic solver GenRTR (Baker et al. 2007). The trust-region sub-problem is solved using the *truncated conjugate gradient* method Absil et al. (2008, Algorithm 11) which does not require inverting the Hessian. The stopping criterion for the sub-problem is based on Absil et al. (2008, (7.10)), i.e.,

$$\|r_{t+1}\| \leq \|r_0\| \min(\|r_0\|^{\theta}, \kappa)$$

where r_t is the residual of the sub-problem at t^{th} iteration of the truncated conjugate gradient method. The parameters θ and κ are set to 1 and 0.1 as suggested in Absil et al. (2008, Sect. 7.5). The parameter $\theta = 1$ ensures that we seek a quadratic rate of convergence near the minimum.

5.3 Numerical complexity

The numerical complexity of manifold-based optimization methods depends on the computational cost of the components listed in the Tables 3, 4 and 5 and the Riemannian metric $\bar{g}_{\bar{x}}$ presented in Table 2. The computational cost of these ingredients are shown below.

1. Objective function $\bar{\phi}(\bar{x})$: Problem dependent.
2. Metric $\bar{g}_x$: The dominant computational cost comes from computing terms like $\mathbf{G}^T\mathbf{G}$, $\bar{\xi}_{\mathbf{G}}^T\bar{\eta}_{\mathbf{G}}$, and $\bar{\xi}_{\mathbf{U}}^T\bar{\eta}_{\mathbf{U}}$, each of these operations requires a numerical cost of $O(d_1r^2)$. Other matrix operations involve handling matrices of size $r \times r$ with total computational cost of $O(r^3)$.
3. Projecting on the tangent space $T_{\bar{x}}\overline{\mathcal{W}}$ with $\Psi_{\bar{x}}$: It involves multiplications between matrices of sizes $d_1 \times r$ and $r \times r$ which costs $O(d_1r^2)$. Other operations involve handling matrices of size $r \times r$.
4. Projecting on the horizontal space $\mathcal{H}_{\bar{x}}\overline{\mathcal{W}}$ with $\Pi_{\bar{x}}$:
   - Forming the Lyapunov equations: Dominant computational cost of $O(d_1r^2 + d_2r^2)$ with matrix multiplications that cost $O(r^3)$.
   - Solving the Lyapunov equation costs $O(r^3)$ (Bartels and Stewart 1972).
5. Retraction $R_{\bar{x}}$:
   - Computing the retraction on the $\mathrm{St}(r, d_1)$ (the set of matrices of size $d_1 \times r$ with orthonormal columns) costs $O(d_1r^2)$
   - Computing the retraction on $\mathbb{R}_*^{d_1\times r}$ costs $O(d_1r)$
   - Computing the retraction on the set of positive-definite matrices $S_{++}(r)$ costs $O(r^3)$.
6. Riemannian gradient $\overline{\mathrm{grad}}_{\bar{x}}\bar{\phi}$: First, it involves computing the partial derivatives of the cost function $\bar{\phi}$ which depend on the cost function $\bar{\phi}$. Second, the modifications to these partial derivatives involve matrix multiplications between matrices of sizes $d_1 \times r$ and $r \times r$ which costs $O(d_1r^2)$.
7. Riemannian Hessian $\overline{\nabla}_{\bar{\xi}_{\bar{x}}}\overline{\mathrm{grad}_x\phi}$ in the direction $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ on the total space: The Riemannian Hessian on each of the three manifolds, $\mathrm{St}(d_1, r)$, $\mathbb{R}_*^{d_1\times r}$, and $S_{++}(r)$, consists of two terms. The first term is the Euclidean directional derivative of the Riemannian gradient in the direction $\bar{\xi}_{\bar{x}}$, i.e., $\mathrm{D}\overline{\mathrm{grad}_x\phi}[\bar{\xi}_{\bar{x}}]$. The second term is the *correction term* corresponds to the manifold structure and the metric. The summation of these terms is projected on the tangent space $T_{\bar{x}}\overline{\mathcal{W}}$ using $\Psi_{\bar{x}}$.
   - $\mathrm{D}\overline{\mathrm{grad}_x\phi}[\bar{\xi}_{\bar{x}}]$: The computational cost depends on the cost function ϕ and its partial derivatives.
   - Correction term: It involves matrix multiplications with total cost of $O(d_1r^2 + r^3)$.

It is clear that all the geometry related operations are of linear complexity in d_1 and d_2; and cubic (or quadratic) in r. For the case of interest, $r \ll \min(d_1, d_2)$, these operations are therefore computationally very efficient. The ingredients that depend on the problem at hand are the evaluation of the cost function $\bar{\phi}$, computation of its partial derivatives and their directional derivatives along a search direction. In the next section, the computations of the partial derivatives and their directional derivatives are presented for the low-rank matrix completion problem.

6 Numerical comparisons

In this section, we show numerical comparisons with state-of-the-art algorithms. The application of choice is the low-rank matrix completion problem for which a number of algorithms with numerical codes are readily available. The competing algorithms are classified according to the way they view the set of fixed-rank matrices.

We show that our generic geometries connect closely with a number of competing methods. In addition to this, we bring out a few conceptual differences between the competing algorithms and our geometric algorithms. Finally, the numerical comparisons suggest that our geometric algorithms compete favorably with state-of-the-art algorithms.

6.1 Matrix completion as a benchmark for numerical comparisons

To illustrate the notions presented in the paper, we consider the problem of low-rank matrix completion (described in Sect. 2.1) as the benchmark application. The objective function is a smooth least square function and the search space is the space of fixed-rank matrices as shown in (2). It is an optimization problem that has attracted a lot of attention in recent years. Consequently, a large body of algorithms have been proposed. Hence, this provides a good benchmark to not only compare different algorithms including our Riemannian geometric algorithms but also bring out the salient features of different algorithms and geometries. Rewriting the optimization formulation of the low-rank matrix completion, we have

$$\min_{\mathbf{W}\in\mathbb{R}_r^{d_1\times d_2}} \ \frac{1}{|\Omega|}\|\mathcal{P}_\Omega(\mathbf{W}) - \mathcal{P}_\Omega(\mathbf{W}^\star)\|_F^2 \tag{19}$$

where $\mathbb{R}_r^{d_1\times d_2}$ is the set of rank-r matrices of size $d_1 \times d_2$ and $\mathbf{W}^*$ is a matrix of size $d_1 \times d_2$ whose entries are given for indices $(i, j) \in \Omega$. $|\Omega|$ denotes the cardinality of the set Ω ($|\Omega| \ll d_1 d_2$). $\mathcal{P}_\Omega$ is the orthogonal sampling operator, $\mathcal{P}_\Omega(\mathbf{W})_{ij} = \mathbf{W}_{ij}$ if $(i, j) \in \Omega$ and $\mathcal{P}_\Omega(\mathbf{W})_{ij} = 0$ otherwise. We seek to learn a rank-r matrix that best approximates the entries of $\mathbf{W}^*$ for the indices in Ω.

As mentioned before, the Tables 2, 3, 4 and 5 provide all the requisite information for implementing the (steepest) gradient descent and the Riemannian trust-region algorithms of Sect. 5. The only components still missing are the matrix formulas for the partial derivatives and their directional derivatives. These formulas are shown in Table 6. As regards the computational cost, the geometry related operations are linear in d_2 and d_2 (Sect. 5.3); and the evaluation of the cost function, the computations of the partial derivatives and their directional derivatives depend *primarily* on the computational cost of the auxiliary (sparse) variables $\mathbf{S}$ and $\mathbf{S}_*$ and the matrix multiplications of kind $\mathbf{SH}$ or $\mathbf{S}_*\mathbf{H}$ shown in Table 6. The variables $\mathbf{S}$ and $\mathbf{S}_*$ are respectively interpreted as the gradient of the cost function in the Euclidean space $\mathbb{R}^{d_1\times d_2}$ and its directional derivative in the direction $\bar{\xi}_{\bar{x}}$. Finally, we have the following additional computation cost.

- Cost of computing $\bar{\phi}(\bar{x})$: $O(|\Omega|r)$.
- Computational cost of forming the sparse matrix $\mathbf{S}$: Computing the non-zero entries of $\mathbf{S}$ costs $O(|\Omega|r)$ plus the cost of updating of a sparse matrix for specific indices in Ω. Both of these operations can be performed efficiently by the Matlab routines used in Cai et al. (2010), Boumal and Absil (2011), Wen et al. (2012).
- Computational cost of forming the sparse matrix $\mathbf{S}_*$: $O(|\Omega|r)$.
- Computing the matrix multiplication $\mathbf{SH}$ or $\mathbf{S}_*\mathbf{H}$: Each costs $O(|\Omega|r)$. One gradient evaluation ($\mathrm{grad}_{\bar{x}}\bar{\phi}$) *precisely* needs two such operations and a Hessian evaluation ($\overline{\nabla}_{\bar{\xi}_{\bar{x}}}\overline{\mathrm{grad}_x \phi}$) needs four such operations.
- Cost of computing all other matrix products: $O(d_1r^2 + d_2r^2 + r^3)$.

All simulations are performed in Matlab on a 2.53 GHz Intel Core i5 machine with 4 GB of RAM. We use the Matlab codes of all the competing algorithms supplied by their authors for our numerical studies. For each example, a $d_1 \times d_2$ random matrix of rank r is generated as in Cai et al. (2010). Two matrices $\mathbf{A} \in \mathbb{R}^{d_1 \times r}$ and $\mathbf{B} \in \mathbb{R}^{d_2 \times r}$ are generated according to a Gaussian distribution with zero mean and unit standard deviation. The matrix product $\mathbf{AB}^T$ then gives a random matrix of rank r. A fraction of the entries are randomly removed with uniform probability. Note that the dimension of the space of $d_1 \times d_2$ matrices of rank r is $(d_1+d_2-r)r$ and the number of known entries is a *multiple* of this dimension. This multiple or ratio is called the *over-sampling ratio* or simply, *over-sampling* (OS). The over-sampling ratio (OS) determines the number of entries that are known. A OS $= 6$ means that $6(d_1 + d_2 - r)r$ of randomly and uniformly selected entries are known a priori out of a total of d_1d_2 entries. We use an initialization that is based on the rank-r dominant singular value decomposition of $\mathcal{P}_\Omega(\mathbf{W}^*)$ (Boumal and Absil 2011). It should be stated that this procedure only provides a good initialization for the algorithms and we do not comment on the quality of this initialization procedure. The numerical codes of the proposed algorithms for the low-rank matrix completion problem are available from the website http://www.montefiore.ulg.ac.be/~mishra/pubs.html. The generic implementations of the three fixed-rank geometries can be found in the Manopt optimization toolbox (Boumal et al. 2013) which provides additional algorithmic implementations.

All the considered gradient descent schemes, except RTRMC-1 (Boumal and Absil 2011) and SVP (Jain et al. 2010), use the adaptive step-size guess procedure (17) and the maximum number of iterations set at 200. For the trust-region scheme, the maximum number of outer iterations is set at 100 (we expect a better rate of convergence in terms of the outer iterations) and the number of inner iterations (for solving the trust-regions sub-problem) is bounded by 100. Finally, the algorithms are stopped if the objective function value is below 10^{-20}.

In both the schemes we also set the initial step-size s_0 (for gradient descent) and the initial trust-region radius Δ_0 (for trust-region) including the upper bound on the radius, $\bar{\Delta}$. We do this by *linearizing* the search space. In particular, for the factorization $\mathbf{W} = \mathbf{UBV}^T$ (similarly for the other two factorizations) we solve the following optimization problem

$$s_0 = \arg\min_s \|\mathcal{P}_\Omega((\mathbf{U} - s\bar{\xi}_\mathbf{U})(\mathbf{B} - s\bar{\xi}_\mathbf{B})(\mathbf{V} - s\bar{\xi}_\mathbf{V})^T) - \mathcal{P}_\Omega(\mathbf{W}^\star)\|_F^2,$$

Table 6 Computation of the Riemannian gradient and its directional derivative in the direction $\bar{\xi}_{\bar{x}} \in \mathcal{H}_{\bar{x}}$ for the low-rank matrix completion problem (19)

	$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$
Cost function $\bar{\phi}(\bar{x})$	$\frac{1}{\lvert\Omega\rvert}\lVert\mathcal{P}_\Omega(\mathbf{G}\mathbf{H}^T) - \mathcal{P}_\Omega(\mathbf{W}^\star)\rVert_F^2$	$\frac{1}{\lvert\Omega\rvert}\lVert\mathcal{P}_\Omega(\mathbf{U}\mathbf{B}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{W}^\star)\rVert_F^2$	$\frac{1}{\lvert\Omega\rvert}\lVert\mathcal{P}_\Omega(\mathbf{U}\mathbf{Y}^T) - \mathcal{P}_\Omega(\mathbf{W}^\star)\rVert_F^2$
Partial derivatives of $\bar{\phi}$	$(\mathbf{S}\mathbf{H}, \mathbf{S}^T\mathbf{G}) \in \mathbb{R}^{d_1\times r} \times \mathbb{R}^{d_2\times r}$ where $\mathbf{S} = \frac{2}{\lvert\Omega\rvert}(\mathcal{P}_\Omega(\mathbf{G}\mathbf{H}^T) - \mathcal{P}_\Omega(\mathbf{W}^\star))$	$(\mathbf{S}\mathbf{V}\mathbf{B}, \mathbf{U}^T\mathbf{S}\mathbf{V}, \mathbf{S}^T\mathbf{U}\mathbf{B}) \in \mathbb{R}^{d_1\times r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{d_2\times r}$ where $\mathbf{S} = \frac{2}{\lvert\Omega\rvert}(\mathcal{P}_\Omega(\mathbf{U}\mathbf{B}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{W}^\star))$	$(\mathbf{S}\mathbf{Y}, \mathbf{S}^T\mathbf{U}) \in \mathbb{R}^{d_1\times r} \times \mathbb{R}^{d_2\times r}$ where $\mathbf{S} = \frac{2}{\lvert\Omega\rvert}(\mathcal{P}_\Omega(\mathbf{U}\mathbf{Y}^T) - \mathcal{P}_\Omega(\mathbf{W}^\star))$
Riemannian gradient $\mathrm{grad}_{\bar{x}}\bar{\phi}$ from Table 5	$(\mathbf{S}\mathbf{H}\mathbf{G}^T\mathbf{G}, \mathbf{S}^T\mathbf{G}\mathbf{H}^T\mathbf{H})$	$(\mathbf{S}\mathbf{V}\mathbf{B} - \mathbf{U}^T\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V}\mathbf{B}), \mathbf{B}\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V})\mathbf{B}, \mathbf{S}^T\mathbf{U}\mathbf{B} - \mathbf{V}^T\mathrm{Sym}(\mathbf{V}^T\mathbf{S}^T\mathbf{U}\mathbf{B}))$	$(\mathbf{S}\mathbf{Y} - \mathbf{U}^T\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{Y}), \mathbf{S}^T\mathbf{U}\mathbf{Y}^T\mathbf{Y})$
Directional derivative of the Riemannian gradient and its projection, i.e., $\Psi_{\bar{x}}(\mathrm{D}\overline{\mathrm{grad}_x\phi}[\bar{\xi}_{\bar{x}}])$			
$\mathbf{W} = \mathbf{G}\mathbf{H}^T$	$\Psi_{\bar{x}}(\mathbf{S}_*\mathbf{H}\mathbf{G}^T\mathbf{G} + \mathbf{S}\bar{\xi}_{\mathbf{H}}\mathbf{G}^T\mathbf{G} + 2\mathbf{S}\mathbf{H}\mathrm{Sym}(\mathbf{G}^T\bar{\xi}_{\mathbf{G}}), \mathbf{S}_*^T\mathbf{G}\mathbf{H}^T\mathbf{H} + \mathbf{S}^T\bar{\xi}_{\mathbf{G}}\mathbf{H}^T\mathbf{H} + 2\mathbf{S}^T\mathbf{G}\mathrm{Sym}(\mathbf{H}^T\bar{\xi}_{\mathbf{H}}))$ where $\mathbf{S}_* = \frac{2}{\lvert\Omega\rvert}\mathcal{P}_\Omega(\mathbf{G}\bar{\xi}_{\mathbf{H}}^T + \bar{\xi}_{\mathbf{G}}\mathbf{H}^T)$		
$\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$	$\Psi_{\bar{x}}(\mathbf{S}_*\mathbf{V}\mathbf{B} + \mathbf{S}\bar{\xi}_{\mathbf{V}}\mathbf{B} + \mathbf{S}\mathbf{V}\bar{\xi}_{\mathbf{B}} - \bar{\xi}_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V}\mathbf{B}), 2\mathrm{Sym}(\mathbf{B}\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V})\bar{\xi}_{\mathbf{B}}) + \mathbf{B}\mathrm{Sym}(\bar{\xi}_{\mathbf{U}}^T\mathbf{S}\mathbf{V} + \mathbf{U}^T\mathbf{S}_*\mathbf{V} + \mathbf{U}^T\mathbf{S}\bar{\xi}_{\mathbf{V}})\mathbf{B}, \mathbf{S}_*^T\mathbf{U}\mathbf{B} + \mathbf{S}\bar{\xi}_{\mathbf{U}}\mathbf{B} + \mathbf{S}^T\mathbf{U}\bar{\xi}_{\mathbf{B}} - \bar{\xi}_{\mathbf{V}}\mathrm{Sym}(\mathbf{V}^T\mathbf{S}^T\mathbf{U}\mathbf{B}))$ where $\mathbf{S}_* = \frac{2}{\lvert\Omega\rvert}\mathcal{P}_\Omega(\mathbf{U}\mathbf{B}\bar{\xi}_{\mathbf{V}}^T + \mathbf{U}\bar{\xi}_{\mathbf{B}}\mathbf{V}^T + \bar{\xi}_{\mathbf{U}}\mathbf{B}\mathbf{V}^T)$		
$\mathbf{W} = \mathbf{U}\mathbf{Y}^T$	$\Psi_{\bar{x}}(\mathbf{S}_*\mathbf{Y} + \mathbf{S}\bar{\xi}_{\mathbf{Y}} - \bar{\xi}_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{Y}), \mathbf{S}_*^T\mathbf{U}\mathbf{Y}^T\mathbf{Y} + \mathbf{S}^T\bar{\xi}_{\mathbf{U}}\mathbf{Y}^T\mathbf{Y} + 2\mathbf{S}^T\mathbf{U}\mathrm{Sym}(\mathbf{Y}^T\bar{\xi}_{\mathbf{Y}}))$ where $\mathbf{S}_* = \frac{2}{\lvert\Omega\rvert}\mathcal{P}_\Omega(\mathbf{U}\bar{\xi}_{\mathbf{Y}}^T + \bar{\xi}_{\mathbf{U}}\mathbf{Y}^T)$		

$\Psi_{\bar{x}}$ is the projection operator defined in Table 3 and $\mathrm{Sym}(\cdot)$ extracts the symmetric part, $\mathrm{Sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^T)/2$. The development of these formulas follows systematically using the *chain rule* of computing the derivatives. The auxiliary variables $\mathbf{S}$ and $\mathbf{S}_*$ are interpreted as the gradient of the cost function in the Euclidean space $\mathbb{R}^{d_1\times d_2}$ and its directional derivative in the direction $\bar{\xi}_{\bar{x}}$ respectively where $\bar{\xi}_{\bar{x}}$ is the Riemannian gradient. The above objective function is a degree 6 polynomial in s and thus, the global minimum s_0 can be obtained *numerically* (and computationally efficiently) by finding the roots of a degree 5 polynomial. Δ_0 is then set to $\frac{s_0}{4^3}\sqrt{\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\xi}_{\bar{x}})}$. The numerator of Δ_0 is the linearized trust-region radius and the reduction by 4^3 considers the fact that this linearization might lead to an over-ambitious radius. Overall, this promotes a few extra gradient descent steps during the initial phase of the trust-region algorithm. The radii are upper-bounded as $\bar{\Delta} = 2^{10}\delta_0$. The integers 4 and 2 are used in the context of trust-region radius where an increase is usually by a factor of 2 and a reduction is by a factor of 4 (Absil et al. 2008, Algorithm 10). The integers 3 and 10 have been chosen empirically.

We consider the problem instance of completing a 32000×32000 matrix $\mathbf{W}^\star$ of rank 5 as the running example in many comparisons. The over-sampling ratio OS is 8 implying that 0.25 % (2.56×10^6 out of 1.04×10^9) of entries are randomly and uniformly revealed. In all the comparisons we show 5 random instances to give a a more general comparative view. The over-sampling ratio of 8 does not necessarily make the problem instance very challenging but it provides a standard benchmark to compare numerical scalability and performance of different algorithms. Similarly, a smaller tolerance is needed to observe the asymptotic rate of convergence of the algorithms. A rigorous comparison between different algorithms across different over sampling ratios and scenarios is beyond the scope of the present paper.

6.2 Full-rank factorization $\mathbf{W} = \mathbf{G}\mathbf{H}^T$, MMMF, and LMaFit

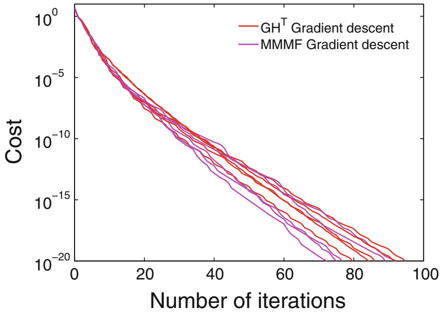
The gradient descent algorithm for the full-rank factorization $\mathbf{W} = \mathbf{G}\mathbf{H}^T$ is closely related to the gradient descent version of the Maximum Margin Matrix Factorization (MMMF) algorithm of Rennie and Srebro (2005). The gradient descent version of MMMF is a descent step in the product space $\mathbb{R}_*^{d_1\times r} \times \mathbb{R}_*^{d_2\times r}$ equipped with the Euclidean metric,

$$\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}}) = \mathrm{Tr}(\bar{\xi}_{\mathbf{G}}^T \bar{\eta}_{\mathbf{G}}) + \mathrm{Tr}(\bar{\xi}_{\mathbf{H}}^T \bar{\eta}_{\mathbf{H}}) \tag{20}$$

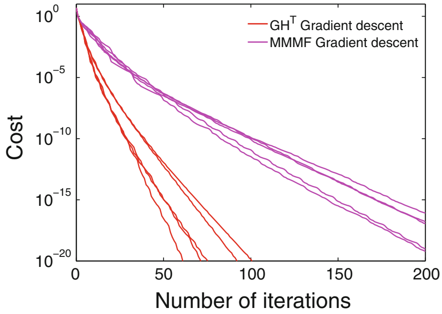
where $\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{W}}$. Note the difference with respect to the metric proposed in Table 2 which is

$$\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}}) = \mathrm{Tr}((\mathbf{G}^T\mathbf{G})^{-1}\bar{\xi}_{\mathbf{G}}^T \bar{\eta}_{\mathbf{G}}) + \mathrm{Tr}((\mathbf{H}^T\mathbf{H})^{-1}\bar{\xi}_{\mathbf{H}}^T \bar{\eta}_{\mathbf{H}}). \tag{21}$$

As a result, the invariance (with respect to $r \times r$ non-singular matrices) is not taken into account in MMMF. In contrast, the proposed retraction in Table 3 is invariant along the set of equivalence classes (5). This resolves the issue of choosing an appropriate step size when there is a discrepancy between $\|\mathbf{G}\|_F$ and $\|\mathbf{H}\|_F$. Indeed, this situation leads to a slower convergence of the MMMF algorithm, whereas the proposed algorithm is not affected (Fig. 3). To illustrate this effect, we consider a rank 5 matrix of size 4000×4000 with 2 % of entries (OS $= 8$) are revealed uniformly at random. The Riemannian gradient descent algorithm based on the Riemannian metric (21) is compared against MMMF. In the first case, the factors at initialization has comparable weights, $\|\mathbf{H}_0\|_F \approx$



(a) $\|\mathbf{H}_0\|_F \approx \|\mathbf{G}_0\|_F$.



(b) $\|\mathbf{H}_0\|_F \approx 2\|\mathbf{G}_0\|_F$.

Fig. 3 5 random instances of low-rank matrix completion problems under different weights of factors at initialization. The proposed metric (21) resolves the issue of choosing an appropriate step-size when there is a discrepancy between $\|\mathbf{G}\|_F$ and $\|\mathbf{H}\|_F$, a situation that leads to a slow convergence of the MMMF algorithm **a** $\|\mathbf{H}_0\|_F \approx \|\mathbf{G}_0\|_F$ **b** $\|\mathbf{H}_0\|_F \approx 2\|\mathbf{G}_0\|_F$

$\|\mathbf{G}_0\|_F$. In the second case, we make factors at initialization slightly unbalanced, $\|\mathbf{H}_0\|_F \approx 2\|\mathbf{G}_0\|_F$. This discrepancy of the weights of the factors is not handled properly with the Euclidean metric (20) and hence, the rate of convergence of MMMF is affected as the plots show in Fig. 3. The same also demonstrates that MMMF performs well when the factors are balanced. This understanding comes with notion of non-uniqueness of matrix factorization. In the previous example, though we force a bad balancing at initialization to show the relevance of scale-invariance, such a case might occur naturally for some particular cost functions and random initializations (e.g., when $d_1 \ll d_2$). Hence, a discussion of choosing an appropriate metric has its merits.

The algorithm LMaFit of Wen et al. (2012) for the low-rank matrix completion problem also relies on the factorization $\mathbf{W} = \mathbf{G}\mathbf{H}^T$ to alternatively learn the matrices $\mathbf{W}$, $\mathbf{G}$ and $\mathbf{H}$ so that the error $\|\mathbf{W} - \mathbf{G}\mathbf{H}^T\|_F^2$ is minimized while ensuring that the entries of $\mathbf{W}$ agree with the known entries, i.e., $\mathcal{P}_\Omega(\mathbf{W}) = \mathcal{P}_\Omega(\mathbf{W}^\star)$. The algorithm is a tuned version the block-coordinate descent algorithm that has a smaller computational cost per iteration and better convergence than the standard non-linear Gauss-Seidel scheme.

We compare our Riemannian algorithms for the factorization $\mathbf{W} = \mathbf{G}\mathbf{H}^T$ with LMaFit and MMMF in Fig. 4. Both MMMF and our gradient descent algorithm perform similarly. Asymptotically, the trust-region has a better rate of convergence both in terms of iterations and computational complexity. LMaFit reached 200 iterations. During the initial few iterations, the trust-region algorithm adapts itself to the problem structure and takes non-effective steps where as the gradient descent algorithms are effective during the initial phase. Once in the region of convergence, the trust-region shows a better behavior.

6.3 Polar factorization $\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ and SVP

Here, we first illustrate the empirical evidence that constraining $\mathbf{B}$ to be diagonal (as is the case with singular value decomposition) is detrimental to optimization. We consider the simplest implementation of a gradient descent algorithm for matrix

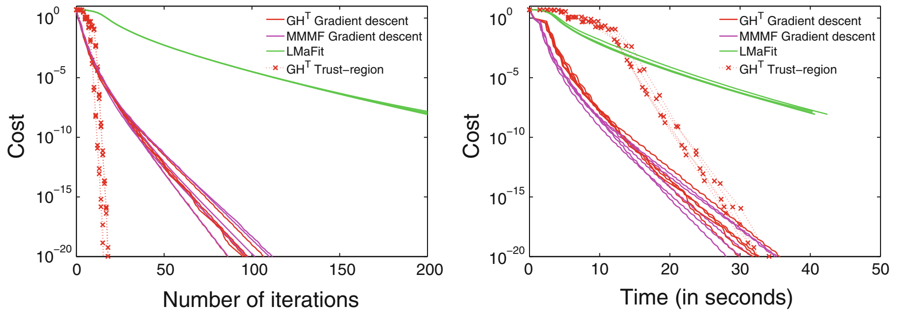


Fig. 4 5 random instances of rank 5 completion of 32000 × 32000 matrix with OS = 8. LMaFit has a smaller computational complexity per iteration but the convergence seems to suffer for large-scale matrices. MMMF and the gradient descent scheme perform similarly. After a slow start, the trust-region scheme shows a better rate of convergence

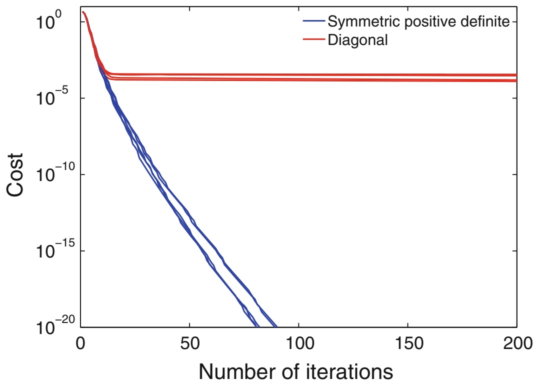


Fig. 5 Convergence of a gradient descent algorithm is affected by making $\mathbf{B}$ diagonal for the factorization $\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$. The retraction updates for both the algorithms are same. The only difference is in the computation of the Riemannian gradient on the search space of Diag_{++} versus $S_{++}(r)$. The *red curves* reached 200 iterations (color figure online)

completion problem (see below). The plots shown in Fig. 5 compare the behavior of the same algorithm in the search space $\mathrm{St}(r, d_1) \times S_{++}(r) \times \mathrm{St}(r, d_2)$ (Sect. 3.2) and $\mathrm{St}(r, d_1) \times \mathrm{Diag}_{++}(r) \times \mathrm{St}(r, d_2)$ (singular value decomposition). $\mathrm{Diag}_{++}(r)$ is the set of diagonal matrices of size $r \times r$ with positive entries. The metric and retraction updates are same for both the algorithms as shown in Table 3. The difference lies in constraining $\mathbf{B}$ to be diagonal which means that the Riemannian gradient for the later case is also diagonal and belongs to the space of $r \times r$ diagonal matrices, $\mathrm{Diag}(r)$ (the tangent space of the manifold $\mathrm{Diag}_{++}(r)$). The matrix formulas for the factor $\mathbf{B}$ of the Riemannian gradient are therefore,

$$\begin{array}{ll} \mathbf{B}\mathrm{Sym}(\mathbf{U}^T\mathbf{S}\mathbf{V})\mathbf{B} & \text{when } \mathbf{B} \in S_{++}(r), \text{ and} \\ \mathbf{B}\mathrm{diag}(\mathbf{U}^T\mathbf{S}\mathbf{V})\mathbf{B} & \text{when } \mathbf{B} \in \mathrm{Diag}_{++}(r) \end{array}$$

where the notations are same as in Table 6 and $\mathrm{diag}(\cdot)$ extracts the diagonal of a matrix, i.e., $\mathrm{diag}(\mathbf{A})$ is a diagonal matrix of size $r \times r$ with entries equal to the diagonal of $\mathbf{A}$. The empirical observation that convergence suffers from imposing diagonalization on $\mathbf{B}$ is a generic observation and has been noticed across various problem instances.

The problem here involves completing a 4000×4000 of rank 5 from 2 % of observed entries.

OptSpace (Keshavan et al. 2010) also relies on the factorization $\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$, but with $\mathbf{B} \in \mathbb{R}^{r \times r}$. At each iteration, the algorithm minimizes the cost function, say $\bar{\phi}$, by solving

$$\min_{\mathbf{U},\mathbf{V}} \quad \bar{\phi}(\mathbf{U}, \mathbf{B}, \mathbf{V})$$

over the *bi*-Grassmann manifold, $\mathrm{Gr}(r, d_1) \times \mathrm{Gr}(r, d_2)$ ($\mathrm{Gr}(r, d_1)$ denotes the set of r-dimensional subspaces in $\mathbb{R}^{d_1}$) obtained by fixing $\mathbf{B}$ and then solving the inner optimization problem

$$\min_{\mathbf{B}} \quad \bar{\phi}(\mathbf{U}, \mathbf{B}, \mathbf{V}) \tag{22}$$

for fixed $\mathbf{U}$ and $\mathbf{V}$. The algorithm thus alternates between a gradient descent step on the subspaces $\mathbf{U}$ and $\mathbf{V}$ for fixed $\mathbf{B}$, and a least-square estimation of $\mathbf{B}$ (matrix completion problem) for fixed $\mathbf{U}$ and $\mathbf{V}$. The proposed framework is different from OptSpace in the choice $\mathbf{B}$ positive definite versus $\mathbf{B} \in \mathbb{R}^{r \times r}$. As a consequence, each step of the algorithm retains the geometry of polar factorization. Our algorithm also differs from OptSpace in the simultaneous and progressive nature of the updates. A potential limitation of OptSpace comes from the fact that the inner optimization problem (22) may not be always solvable efficiently for other applications.

The singular value projection (SVP) algorithm of Jain et al. (2010) is based on the singular value decomposition (SVD) $\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ with $\mathbf{B} \in \mathrm{Diag}_{++}(r)$. It can also be interpreted in the considered framework as a gradient descent algorithm in the Euclidean space $\mathbb{R}^{d_1 \times d_2}$ (and hence, not the Riemannian gradient), along with an efficient SVD-projection based retraction exploiting the sparse structure of the gradient $\xi_{\mathrm{Euclidean}}$ (the gradient in the Euclidean space $\mathbb{R}^{d_1 \times d_2}$, same as $\mathbf{S}$ in Table 6) for the matrix completion problem. A general update for SVP can be written as

$$\mathbf{U}_{+}\mathbf{B}_{+}\mathbf{V}_{+}^T = \mathrm{SVD}_r(\mathbf{U}\mathbf{B}\mathbf{V}^T - \xi_{\mathrm{Euclidean}}),$$

where $\mathrm{SVD}_r(\cdot)$ extracts the dominant r singular values and singular vectors. An intrinsic limitation of the approach is that the computational cost of the algorithm is conditioned on the particular structure of the gradient. For instance, efficient routines exist for modifying the SVD with sparse (Larsen 1998) or low-rank updates (Brand 2006).

Both SVP and our gradient descent implementation use the Armijo backtracking method (Nocedal and Wright 2006, Procedure 3.1). The difference is that for computing an initial step-size guess at each iteration SVP uses $\frac{d_1 d_2}{|\Omega|(1+\delta)}$ with $\delta = 1/3$ as proposed in Jain et al. (2010) while our gradient descent implementation uses the adaptive step-size procedure (17). Figure 6 shows the competitiveness of the proposed framework of factorization model $\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ with the SVP algorithm. Again, the trust-region asymptotically shows a better performance. The test example is an incomplete rank-5 matrix of size 32000×32000 with $\mathrm{OS} = 8$. We could not compare the performance of OptSpace as some of the Matlab operations (in the code supplied

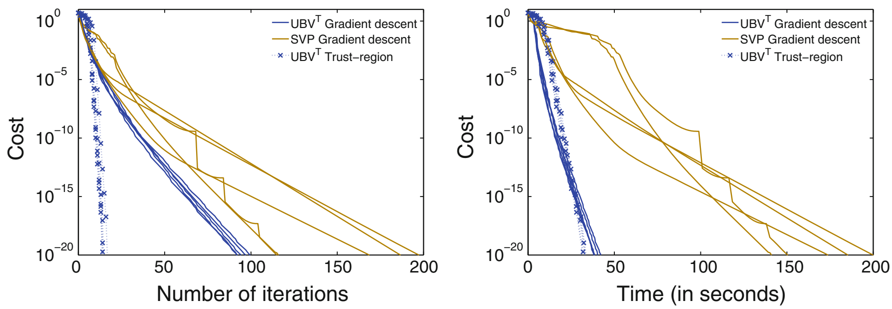


Fig. 6 Illustration of the Riemannian algorithms on low-rank matrix completion problem for the factorization $\mathbf{W} = \mathbf{UBV}^T$ on 5 random instances. Even though the number of iterations of SVP and our gradient descent are similar for some instances, the timings are very different. The main computational burden for SVP comes from computing the r dominant singular value decomposition which is absent in the quotient geometry. Except for a few sparse-matrix computations, most of our computations involve operations on dense matrices of sizes $d_1 \times r$ and $r \times r$ (Sect. 5.3)

by the authors) have not been optimized for large-scale matrices. We have, however, observed a good performance of OptSpace on smaller size instances.

6.4 Subspace-projection factorization $\mathbf{W} = \mathbf{UY}^T$ and RTRMC

The choice of metric for the subspace-projection factorization shown in Table 2, i.e.,

$$\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}}) = \mathrm{Tr}(\bar{\xi}_{\mathbf{U}}^T \bar{\eta}_{\mathbf{U}}) + \mathrm{Tr}((\mathbf{Y}^T\mathbf{Y})^{-1} \bar{\xi}_{\mathbf{Y}}^T \bar{\eta}_{\mathbf{Y}}) \tag{23}$$

is motivated by the fact that the total space $\mathrm{St}(r, d_1) \times \mathbb{R}_*^{d_2 \times r}$ equipped with the proposed metric is a complete Riemannian space and invariant to change of coordinates of the column space $\mathbf{Y}$. An alternative would be to consider the standard Euclidean metric for $\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}} \in T_{\bar{x}}\overline{\mathcal{W}}$,

$$\bar{g}_{\bar{x}}(\bar{\xi}_{\bar{x}}, \bar{\eta}_{\bar{x}}) = \mathrm{Tr}(\bar{\xi}_{\mathbf{U}}^T \bar{\eta}_{\mathbf{U}}) + \mathrm{Tr}(\bar{\xi}_{\mathbf{Y}}^T \bar{\eta}_{\mathbf{Y}}) \tag{24}$$

which is also invariant by the group action $\mathcal{O}(r)$ (the set of $r \times r$ matrices with orthonormal columns and rows) and thus, a valid Riemannian metric. This metric is for instance adopted in Simonsson and Eldén (2010), and recently also proposed in Absil et al. (2012) where the authors give a closed-form description of a Riemannian Newton method including a closed-form expression for the geodesic. Although this alternative choice is appealing for its numerical simplicity, Fig. 7 clearly illustrates the benefits of optimizing with a metric that considers the scaling invariance property. The algorithm with the Euclidean metric (24) flattens out due to a very slow rate of convergence. Under identical initializations and choice of step-size rule, our proposed metric (23) prevents the numerical ill-conditioning of the partial derivatives of the cost function (Table 6) that arises in the presence of unbalanced factors $\mathbf{U}$ and $\mathbf{Y}$, i.e., $\|\mathbf{U}\|_F \not\approx \|\mathbf{Y}\|_F$.

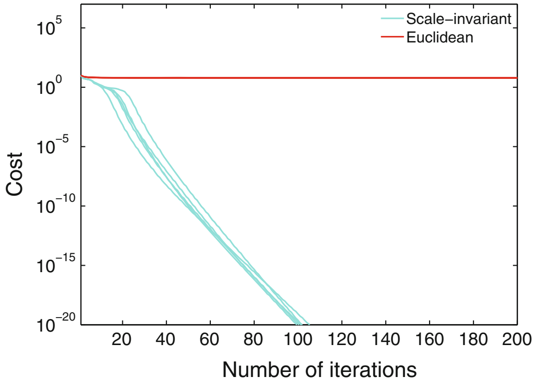


Fig. 7 The choice of a scale-invariant metric (23) for subspace-projection factorization algorithm dramatically affects the performance of the algorithm. The algorithm with the Euclidean metric (24) flattens out due to a very slow rate of convergence because of numerical ill-conditioning due to the presence of unbalanced factors, $\|\mathbf{U}\|_F \not\approx \|\mathbf{Y}\|_F$. The example shown involves completing a rank-5 completion of a 4000 × 4000 matrix with 98 % (OS = 8) entries missing but the observation is generic

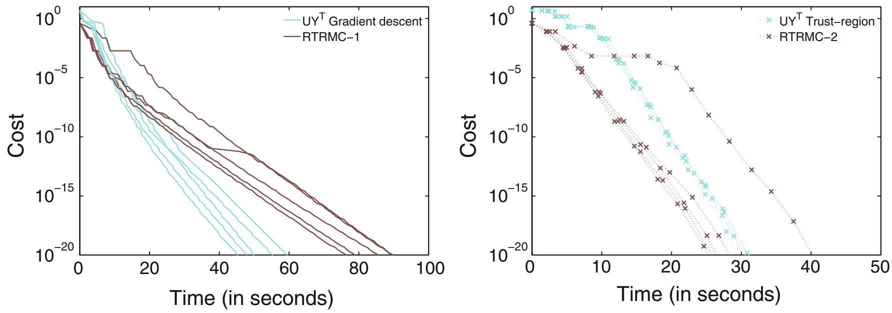


Fig. 8 5 random instances of rank-5 completion of 32000 × 32000 matrix with OS = 8. The framework proposed in this paper is competitive with RTRMC when $d_1 \approx d_2$. For the trust-region algorithms, during the initial few iterations RTRMC-2 shows a better performance owing to the efficient least-square estimation of $\mathbf{Y}$. Asymptotically, both the algorithms perform similarly. For the gradient descent algorithms, however, our implementation shows a better timing performance

The subspace-projection factorization is also exploited in the recent papers Boumal and Absil (2011), Dai et al. (2011, 2012) for the low-rank matrix completion problem. In RTRMC of Boumal and Absil (2011) the authors exploit the fact that in the variable $\mathbf{Y}$, $\min_{\mathbf{Y}} \quad \bar{\phi}(\mathbf{U}, \mathbf{Y})$ is a least square problem that has a closed-form solution. They are, thus, left with an optimization problem in the other variable $\mathbf{U}$ on the Grassmann manifold $\mathrm{Gr}(r, d_1)$.

The resulting geometry of RTRMC is efficient in situations where $d_1 \ll d_2$ where the least square is solved efficiently in the dimension $d_2 r$ and the optimization problem is on a smaller search space of dimension $d_1 r - r^2$. The advantage is reduced in square problems and the numerical experiments in Fig. 8 suggest that our generic algorithm compares favorably to the Grassmannian algorithm in Boumal and Absil (2011) in that case. Similar to our trust-region algorithm, RTRMC-2 is a trust-region implementation with the parameters $\theta = 1$ and $\kappa = 0.1$ (Sect. 5.2). The parameters Δ_0 and $\bar{\Delta}$ are chosen as suggested in Boumal and Absil (2012). Both RTRMC and our trust-region algorithm use the solver GenRTR (Baker et al. 2007) to solve the trust-region subproblem. RTRMC-1 is RTRMC-2 with the Hessian replaced by identity that yields the steepest descent algorithm. The number of iterations needed by both the algorithms are similar and hence, not shown in Fig. 8.

6.5 Quotient and embedded viewpoints

In Sect. 3 we have viewed the set of fixed-rank matrices as the product space of well-studied manifolds $\mathrm{St}(r, d_1)$ (the set of matrices of size $d_1 \times r$ with orthonormal columns), $\mathbb{R}_*^{d_1 \times r}$ (the set of matrices of size $d_1 \times r$ with full column rank) and $S_{++}(r)$ (the set of positive definite matrices of size $r \times r$) and consequently, the search space admitted a Riemannian quotient manifold structure. A different viewpoint is that of the *Riemannian embedded submanifold* approach. The search space $\mathbb{R}_r^{d_1 \times d_2}$ (the set of rank-r matrices of size $d_1 \times d_2$) admits a Riemannian submanifold of the Euclidean space $\mathbb{R}^{d_1 \times d_2}$ (Vandereycken 2013, Proposition 2.1). Recent papers Vandereycken (2013), Shalit et al. (2010) investigate the search space in detail and develop the notions of optimizing a smooth cost function. While conceptually the iterates move on the embedded submanifold, numerically the implementation is done using factorization models, the full-rank factorization is used in Shalit et al. (2010) and a compact singular value decomposition is used in Vandereycken (2013).

The characterization of the embedded geometry is tabulated in Table 7 using the factorization model $\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T$. Here $\Sigma \in \mathrm{Diag}_{++}$ is a diagonal matrix with positive entries, $\mathbf{U} \in \mathrm{St}(r, d_1)$ and $\mathbf{V} \in \mathrm{St}(r, d_2)$. The treatment is similar for the factorization $\mathbf{W} = \mathbf{G}\mathbf{H}^T$ as the underlying geometry is the same (Shalit et al. 2010).

The visualization of the search space as an embedded submanifold of $\mathbb{R}^{d_1 \times d_2}$ has some key advantages. For example, the notions of geometric objects can be interpreted in a straight forward way. In the matrix completion problem, this also allows us to compute the initial step-size guess (in a search direction) by linearizing the search space (Vandereycken 2013). On the other hand, the product space representation of fixed-rank matrices in Sect. 3 is naturally related to matrix factorization and provides additional flexibility in choosing the metric. It is only the horizontal space (Sect. 4.2) that couples the product spaces. From the optimization point of view this flexibility is also of interest. For instance, it allows us to regularize the matrix factors, say $\mathbf{G}$ and $\mathbf{H}$, differently.

In Fig. 9 we compare our algorithms with LRGeom (the algorithmic implementation of Vandereycken 2013) on 5 random instances. The timing plots for gradient descent and trust-region algorithms show that Riemannian quotient algorithms are competitive with LRGeom. The parameters s_0, Δ_0 and $\bar{\Delta}$ for all the algorithms are set by performing a linearized search as proposed in Sect. 6.1. The linearized step-size search for LRGeom is the one proposed in Vandereycken (2013). LRGeom RTR (the trust-region implementation) shows a better performance during the initial phase of the algorithm. The trust-region schemes based on the quotient geometries seem to spend more time *in transition* to the region of rapid convergence. However asymptotically,

Table 7 Optimization-related ingredients for using the embedded geometry of rank-r matrices at $\mathbf{W} \in \mathbb{R}_r^{d_1 \times d_2}$ (Vandereycken 2013)

	Embedded submanifold $\mathbb{R}_r^{d_1 \times d_2}$
Matrix representation	$\mathbf{W} = \mathbf{U}\Sigma\mathbf{V}^T$ where $\mathbf{U} \in \mathrm{St}(r, d_1)$, $\Sigma \in \mathrm{Diag}_{++}(r)$, and $\mathbf{V} \in \mathrm{St}(r, d_2)$
Tangent space $T_\mathbf{W}\mathbb{R}_r^{d_1 \times d_2}$	$\mathbf{UNV}^T + \mathbf{U}_p\mathbf{V}^T + \mathbf{UV}_p^T : \mathbf{N} \in \mathbb{R}^{r \times r}$, $\mathbf{U}_p \in \mathbb{R}^{d_1 \times r}, \mathbf{U}_p^T\mathbf{U} = \mathbf{0}$, $\mathbf{V}_p \in \mathbb{R}^{d_2 \times r}, \mathbf{V}_p^T\mathbf{V} = \mathbf{0}$
Metric $g_\mathbf{W}(\mathbf{Z}_1, \mathbf{Z}_2)$	$\mathrm{Tr}(\mathbf{Z}_1^T\mathbf{Z}_2)$
Projection of a matrix $\mathbf{Z} \in \mathbb{R}^{d_1 \times d_2}$ onto the tangent space $T_\mathbf{W}\mathbb{R}_r^{d_1 \times d_2}$	$\Pi_\mathbf{W}(\mathbf{Z}) = \{\mathbf{P_U ZP_V} + \mathbf{P_U^\perp ZP_V} + \mathbf{P_U ZP_V^\perp}$: $\mathbf{P_U} := \mathbf{UU}^T$ and $\mathbf{P_U^\perp} := \mathbf{I} - \mathbf{P_U}\}$
Riemannian gradient	$\mathrm{grad}_\mathbf{W} f = \Pi_\mathbf{W}(\mathrm{Grad}_\mathbf{W} \bar{f})$ where $\mathrm{Grad}_\mathbf{W} \bar{f}$ is the gradient of $\bar{f}$ in $\mathbb{R}^{d_1 \times d_2}$
Riemannian connection $\nabla_\xi \eta$ where $\xi, \eta \in T_\mathbf{W}\mathbb{R}_r^{d_1 \times d_2}$ (Absil et al. 2008, Proposition 5.3.2)	$\Pi_\mathbf{W}(\mathrm{D}\bar{\eta}[\bar{\xi}])$ where, $\mathrm{D}\bar{\eta}[\bar{\xi}]$ is the standard Euclidean directional derivative of $\bar{\eta}$ in the direction $\bar{\xi}$
Retraction	$R_\mathbf{W}(\xi) = \mathrm{SVD}(\mathbf{W} + \xi)$ where SVD involves the computation of a thin singular value decomposition of rank $2r$

The rank-r matrix $\mathbf{W}$ is stored in the factorized form $(\mathbf{U}, \Sigma, \mathbf{V})$ resulting form a compact singular value decomposition. As a consequence, it leads to computationally efficient calculations of all the above listed ingredients. The computation of the Riemannian gradient is shown for a smooth cost function $\bar{f} : \mathbb{R}^{d_1 \times d_2} \to \mathbb{R}$ and its restriction f on the manifold $\mathbb{R}_r^{d_1 \times d_2}$

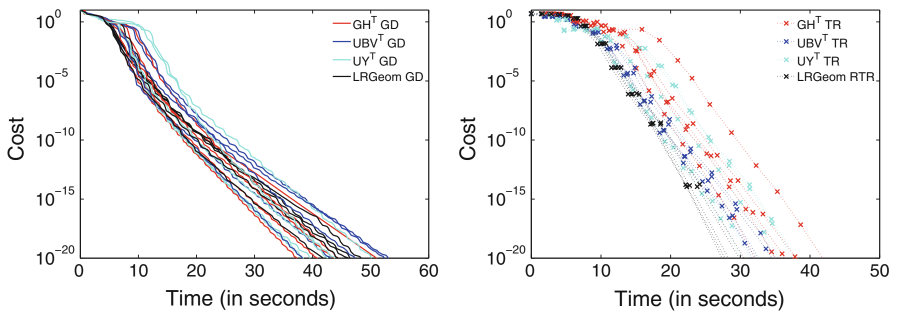


Fig. 9 Low-rank matrix completion of size 32000 × 32000 of rank 5 with OS = 8. Both quotient and embedded geometries behave similarly. The behaviors of the gradient descent (GD) algorithms of these geometries are indistinguishable. The trust-region (TR) schemes perform similarly with LRGeom RTR showing a better performance during the initial few iterations

we obtain the same performance as that of LRGeom RTR. The behaviors of all the gradient descent schemes are inseparable.

7 Conclusion

We have addressed the problem of rank-constrained optimization (1) and presented both first-order and second-order schemes. The proposed framework is general and encompasses recent advances in optimization algorithms. We have shown that classical fixed-rank matrix factorizations have a natural interpretation of classes of equivalences in well-studied manifolds. As a consequence, they lead to a matrix search space that has the geometric structure of a Riemannian quotient manifold, with convenient matrix expressions for all the geometric objects required for an optimization algorithm. The computational cost of involved matrix operations is always linear in the original dimensions of the matrix, which makes the proposed computational framework amenable to large-scale applications. The product structure of the considered total spaces provides some flexibility in choosing the proper metrics on the search space. The relevance of this flexibility was illustrated in the context of the subspace-projection factorization $\mathbf{W} = \mathbf{U}\mathbf{Y}^T$ in Sect. 6.4. The relevance of not fixing the matrix factorization beyond necessity has been illustrated in the context of the factorization $\mathbf{W} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ in Sect. 6.3 where the flexibility of $\mathbf{B}$ to be positive definite instead of diagonal (as is the case with singular value decomposition) results in good convergence properties. Similarly, the advantage of balancing an update for the factorization $\mathbf{W} = \mathbf{G}\mathbf{H}^T$ has been discussed in Sect. 6.2.

All numerical illustrations of the paper were provided on the low-rank matrix completion problem, that permitted a comparison with many existing fixed-rank optimization algorithms. It was shown that the proposed framework compares favorably with most state-of-the-art algorithms while maintaining a complete generality.

The three considered geometries show a comparable numerical performance in the simple examples considered in the paper. However, differences exist in the resulting metrics and related invariance properties, which may lead to a geometry being preferred for a particular problem. In the same way as different matrix factorizations exist and the preference for one factorization over the others is problem dependent, we view the three proposed geometries as three possible choices which the user should exploit as a source of flexibility in the design of a particular optimization algorithm tuned to a particular problem. They are all equivalent in terms of numerical complexity and convergence guarantees.

Optimizing the geometry and the metric to a particular problem such as matrix completion and to a particular dataset will be the topic of future research. Some steps in that direction are proposed in the recent papers Ngo and Saad (2012), Mishra et al. (2012).

Acknowledgments We thank the editor and the two anonymous reviewers for their helpful suggestions which led to various improvements in 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. 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).

References

Abernethy J, Bach F, Evgeniou T, Vert JP (2009) A new approach to collaborative filtering: operator estimation with spectral regularization. J Mach Learn Res 10:803–826

Absil PA, Amodei L, Meyer G (2012) Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. Tech. Rep. UCL-INMA-2012.05, U.C.Louvain

Absil PA, Mahony R, Sepulchre R (2008) Optimization algorithms on matrix manifolds. Princeton University Press, Princeton

Amit Y, Fink M, Srebro N, Ullman S (2007) Uncovering shared structures in multiclass classification. In: Ghahramani Z (ed) Proceedings of the 24th international conference on machine learning, pp 17–24

Baker CG, Absil PA, Gallivan KA (2007) GenRTR: the Generic Riemannian Trust-region package. http://www.math.fsu.edu/cbaker/genrtr/

Bartels RH, Stewart GW (1972) Solution of the matrix equation $ax + xb = c$ [f4] (algorithm 432). Commun ACM 15:820–826

Bhatia R (2007) Positive definite matrices. Princeton University Press, Princeton

Bleakley K, Yamanishi Y (2009) Supervised prediction of drug-target interactions using bipartite local models. Bioinformatics 25:2397–2403

Bonnabel S, Sepulchre R (2009) Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. SIAM J Matrix Anal Appl 31:1055–1070

Boumal N, Absil PA (2011) RTRMC: A Riemannian trust-region method for low-rank matrix completion. In: Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K (eds) Neural information processing systems conference, NIPS, pp 406–414

Boumal N, Absil PA (2012), Low-rank matrix completion via trust-regions on the Grassmann manifold. Tech. rep., UCL-INMA-2012.07

Boumal N, Mishra B, Absil PA, Sepulchre R (2013), Manopt: a Matlab toolbox for optimization on manifolds. arXiv, preprint arXiv:13085200 [csMS]

Brand M (2006) Fast low-rank modifications of the thin singular value decomposition. Linear Algebra Appl 415:20–30

Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM J Optim 20:1956–1982

Cai D, He X, Han J (2007) Efficient kernel discriminant analysis via spectral regression. In: Proceedings of the IEEE international conference on data mining, ICDM, pp 427–432

Candès EJ, Recht B (2008) Exact matrix completion via convex optimization. Found Comput Math 9:717–772

Dai W, Milenkovic O, Kerman E (2011) Subspace evolution and transfer (SET) for low-rank matrix completion. IEEE Trans Signal Process 59:3120–3132

Dai W, Kerman E, Milenkovic O (2012) A geometric approach to low-rank matrix completion. IEEE Trans Inf Theory 58:237–247

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

Evgeniou T, Micchelli C, Pontil M (2005) Learning multiple tasks with kernel methods. J Mach Learn Res 6:615–637

Golub GH, Van Loan CF (1996) Matrix computations, 3rd edn. The Johns Hopkins University Press, 2715 North Charles Street, Baltimore, Maryland 21218–4319

Gross D (2011) Recovering low-rank matrices from few coefficients in any basis. IEEE Trans Inf Theory 57:1548–1566

Jain P, Meka R, Dhillon I (2010) Guaranteed rank minimization via singular value projection. In: Lafferty J, Williams CKI, Shawe-Taylor J, Zemel R, Culotta A (eds) Advances in neural information processing systems. NIPS 23, pp 937–945

Jeffrey DJ (2010) LU factoring of non-invertible matrices. ACM Commun Comput Algebra 44:1–8

Journée M (2009) Geometric algorithms for component analysis with a view to gene expression data analysis. PhD thesis, University of Liège, Liège, Belgium

Keshavan RH, Montanari A, Oh S (2010) Matrix completion from noisy entries. J Mach Learn Res 11:2057–2078

Kulis B, Sustik M, Dhillon IS (2009) Low-rank kernel learning with Bregman matrix divergences. J Mach Learn Res 10:341–376

Kulis B, Saenko K, Darrell T (2011) What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In: Proceedings of the IEEE conference on computer vision and pattern recognition, CVPR, pp 1785–1792

Larsen R (1998) Lanczos bidiagonalization with partial reorthogonalization. Technical Report DAIMI PB-357, Department of Computer Science, Aarhus University

Lee JM (2003) Introduction to smooth manifolds, graduate texts in mathematics, vol 218, 2nd edn. Springer, New York

Lee K, Bresler Y (2010) Admira: atomic decomposition for minimum rank approximation. IEEE Trans Inf Theory 56:4402–4416

Mazumder R, Hastie T, Tibshirani R (2010) Spectral regularization algorithms for learning large incomplete matrices. J Mach Learn Res 11:2287–2322

Meka R, Jain P, Dhillon IS (2009) Matrix completion from power-law distributed samples. In: Bengio Y, Schuurmans D, Lafferty J, Williams CKI, Culotta A (eds) Advances in neural information processing systems 22, NIPS, pp 1258–1266

Meyer G (2011) Geometric optimization algorithms for linear regression on fixed-rank matrices. PhD thesis, University of Liège, Liège, Belgium

Meyer G, Bonnabel S, Sepulchre R (2011b) Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. J Mach Learn Res 11:593–625

Meyer G, Bonnabel S, Sepulchre R (2011a) Linear regression under fixed-rank constraints: a Riemannian approach. In: Proceedings of the 28th international conference on machine learning, ICML, pp 545–552

Mishra B, Adithya Apuroop K, Sepulchre R (2012) A Riemannian geometry for low-rank matrix completion. Tech. rep., arXiv:1211.1550

Mishra B, Meyer G, Bach F, Sepulchre R (2011a) Low-rank optimization with trace norm penalty. Tech. rep., arXiv:1112.2318

Mishra B, Meyer G, Sepulchre R (2011b) Low-rank optimization for distance matrix completion. In: Proceedings of the 50th IEEE conference on decision and control, Orlando (USA), pp 4455–4460

Netflix (2006) The Netflix prize. http://www.netflixprize.com/

Ngo TT, Saad Y (2012) Scaled gradients on Grassmann manifolds for matrix completion. In: Advances in neural information processing systems, NIPS, pp 1421–1429

Nocedal J, Wright SJ (2006) Numerical optimization, 2nd edn. Springer, New York

Piziak R, Odell PL (1999) Full rank factorization of matrices. Math Mag 72:193–201

Rennie J, Srebro N (2005), Fast maximum margin matrix factorization for collaborative prediction. In: Proceedings of the 22nd international conference on machine learning, pp 713–719

Shalit U, Weinshall D, Chechik G (2010) Online learning in the manifold of low-rank matrices. In: Lafferty J, Williams CKI, Shawe-Taylor J, Zemel R, Culotta A (eds) Advances in neural information processing systems 23, pp 2128–2136

Simonsson L, Eldén L (2010) Grassmann algorithms for low rank approximation of matrices with missing values. BIT Numer Math 50:173–191

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

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

Yamanishi Y, Araki M, Gutteridge A, Honda W, Kanehisa M (2008) Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. Bioinformatics 24:i232

Yuan M, Ekici A, Lu Z, Monteiro R (2007) Dimension reduction and coefficient estimation in multivariate linear regression. J R Stat Soc 69:329–346