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

Bamdev Mishra and Rodolphe Sepulchre∗
Department of Electrical Engineering and Computer Science
University of Liège
Emails: \{b.mishra, r.sepulchre\}@ulg.ac.be

May 7, 2014

Abstract
We exploit the versatile framework of Riemannian optimization on quotient manifolds to implement R3MC, a nonlinear conjugate gradient method optimized for matrix completion. The underlying geometry uses a Riemannian metric tailored to the quadratic cost. Numerical comparisons suggest that R3MC robustly outperforms state-of-the-art algorithms across different synthetic and real datasets especially in instances that combine scarcely sampled and ill-conditioned data.

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

$$\min_X \frac{1}{|\Omega|} \|P_\Omega(X) - P_\Omega(X^*)\|_F^2$$
subject to $X \in \mathbb{R}^{n \times m}_r$,  

(1)

where $\mathbb{R}^{n \times m}_r$ is the set of rank-$r$ $n \times m$ matrices, the function $P_\Omega(X)_{ij} = X_{ij}$ if $(i, j) \in \Omega$ and $P_\Omega(X)_{ij} = 0$ otherwise and the norm $\| \cdot \|_F$ is the Frobenius norm. $P_\Omega$ is also called the orthogonal sampling operator and $|\Omega|$ is the cardinality of the set $\Omega$ (equal to the number of known entries). The case of interest is when the rank is much smaller than the matrix dimensions, i.e., $r \ll \min(m, n)$. An important and popular application of the low-rank matrix completion problem is collaborative filtering [1]. Matrix completion results have been proved under various assumptions in [2, 3, 4] and a number of computationally efficient algorithms have been proposed in [1, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In parallel, a number of recent papers like [15, 16] have studied the geometry of the set of rank-$r$ matrices $\mathbb{R}^{n \times m}_r$ in detail.

∗Also affiliated with Orchestron, INRIA Lille Nord-Europe, France
1.1 Related work

Out of a lot of different works on matrix completion we focus on three recent algorithms. Keshavan et al. [3] use the fixed-rank factorization $X = U\Sigma V^T$ to embed the rank constraint where $U$ and $V$ are column-orthonormal matrices of size $n \times r$ and $m \times r$, and $\Sigma \in \mathbb{R}^{r \times r}$. At each iteration, they first learn $U$ and $V$ on the bi-Grassmann manifold $\text{Gr}(r, n) \times \text{Gr}(r, m)$ where $\text{Gr}(r, n)$ is the set of $r$-dimensional subspaces in $\mathbb{R}^n$. Subsequently, a least square problem is solved to learn $\Sigma$. Building upon [3], Ngo et al. [14] propose a scaling on the bi-Grassmann manifold to accelerate the algorithm. In a parallel effort, Mishra et al. [13] use the fixed-rank factorization $X = GH^T$ to embed the rank constraint where $G \in \mathbb{R}^{n \times r}$ and $H \in \mathbb{R}^{m \times r}$ and discuss a Riemannian metric and quotient geometry tailored to (1). $\mathbb{R}^{n \times r}$ is the set of full column rank matrices. Unlike [3, 14], $G$ and $H$ are updated simultaneously in the Riemannian framework [13]. The Riemannian metric in [13] takes into account the approximation of the Hessian of the cost function and is closely related to the metric proposed in [14]. In order to get the best of these methods, we use a three-factor factorization, similar to [3, 14], and exploit the generic Riemannian quotient framework, as in [13], to propose a novel geometry and a novel algorithm.

1.2 Our contribution

Our contribution is a conjugate gradient algorithm R3MC (Algorithm 1) that is based on a novel three-factor Riemannian geometry for matrix completion proposed in Section 2. Although the new quotient geometry enables us to propose second-order methods like the Riemannian trust-region method, we specifically focus on the conjugate-gradients as they offer an appropriate balance between convergence and computational cost. They have shown superior performance in our examples. A Matlab implementation of R3MC may be downloaded from http://www.montefiore.ulg.ac.be/~mishra/softwares/R3MC.html.

The geometric notions to implement an off-the-self conjugate gradient algorithm are listed in Section 3. The computations of the matrix completion related ingredient are presented in Section 4. Finally in Section 5 we illustrate the performance of R3MC across different problem instances, focusing in particular on scarcely sampled and ill-conditioned problems.

2 Three-factor fixed-rank matrix factorization

We consider the following rank-$r$ three-factor matrix factorization of $X \in \mathbb{R}^{n \times m}_r$,

$$X = URV^T$$

(2)

where $U \in \text{St}(r, n)$, $V \in \text{St}(r, m)$ and $R \in \mathbb{R}^{r \times r}_s$. $\text{St}(r, n)$ is the set of matrices of size $n \times r$ with orthonormal columns and $\mathbb{R}^{r \times r}_s$ is the set of matrices of size $r \times r$ with non-zero determinant (full rank square matrices). This factorization owes itself to the thin singular value decomposition (SVD) where $R$ is a diagonal matrix with positive entries [17]. The difference with respect to the thin SVD is that for the factor $R$ we relax the diagonal constraint
to accommodate any full rank square matrix. As a result we have the following geometrical
symmetry in the factorization
\[ X = U R V^T = (U O_1) O_1^T R O_2 (V O_2), \]  
(3)
for any \( O_1, O_2 \in O(r) \), the set of orthogonal matrices of size \( r \times r \). In other words, the
matrix \( X \in \mathbb{R}^{n \times m} \) remains unchanged under the group action defined by (3) with \((O_1, O_2) \in O(r) \times O(r)\). We, thus, optimize on the set of equivalence classes,
\[ [(U, R, V)] = \{(U O_1, O_1^T R O_2, V O_2) : (O_1, O_2) \in O(r) \times O(r)\} \]  
(4)
in the space \( \widehat{\mathcal{M}} = \text{St}(r, n) \times \mathbb{R}^{r \times r} \times \text{St}(r, m) \). The set of equivalence classes is denoted as
\[ \mathcal{M} := \widehat{\mathcal{M}} / (O(r) \times O(r)). \]  
(5)
\( \mathcal{M} \) is the total space (computational space) that is a product space \( \text{St}(r, n) \times \mathbb{R}^{r \times r} \times \text{St}(r, m) \).
The set \( O(r) \times O(r) \) is the called the fiber space. The set of equivalence classes \( \mathcal{M} \) is the
quotient space.

Following the quotient manifold optimization framework of [18], the abstract quotient
search space \( \mathcal{M} \) is given the structure of a Riemannian quotient manifold by choosing a
Riemannian metric, that respects the symmetry (3) on the quotient space. The metric defines
an inner product between tangent vectors on the tangent space of \( \mathcal{M} \). Because the total space
is a product space of well-studied manifolds, a valid metric on the total space is derived from
choosing the natural metric of the product space, e.g., in [19] and [18, Example 3.6.4]. Here we
consider more general metrics and exploit this flexibility to choose a metric that approximates
the Hessian of the cost function (1).

2.1 Motivation for the three-factor factorization

Our main interest in studying the three-factor factorization (2) with respect to two-factor
factorizations [13, 16, 20, 21] is that it separates the mass and the subspace information of the
matrix as in SVD factorization. The mass of \( X \) is concentrated in \( R \) whereas the subspace
information is contained in \( (U, V) \). This separation of mass and subspace leads to a robust
behavior in numerical comparisons. Also because the fiber space is \( O(r) \times O(r) \), by fixing \( R \)
the search space is a bi-Grassmann manifold [3] as \( \text{Gr}(r, n) = \text{St}(r, m) / O(r) \) [19].
Optimization on the bi-Grassmann space has shown good results in [3, 14]. The resulting
Riemannian geometry of (2) is, thus, closely related to the bi-Grassmann geometry.

2.2 Riemannian metric on the tangent space

We represent an element of the quotient space \( \mathcal{M} \) by \( x = [\hat{x}] \) (each element is an equivalence
class) and its matrix representation in the total space \( \widehat{\mathcal{M}} := \text{St}(r, n) \times \mathbb{R}^{r \times r} \times \text{St}(r, m) \) is given
by \( \hat{x} = (U, R, V) \). The tangent space of the total space at \( \hat{x} \in \widehat{\mathcal{M}} \) is the product space of the
tangent spaces of the individual manifolds. From [18, Example 3.5.2] we have the matrix
characterization
\[ T_{\hat{x}} \widehat{\mathcal{M}} = \{(Z_U, Z_R, Z_V) \in \mathbb{R}^{n \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{m \times r} : \ U^T Z_U + Z_U^T U = 0, \ V^T Z_V + Z_V^T V = 0\}. \]  
(6)
Similar to [13], the Riemannian metric on the tangent space $\mathcal{T}_x\overline{\mathcal{M}}$ is motivated by both the symmetry and the least square cost function of (1). The Riemannian metric on the tangent space $\mathcal{T}_x\overline{\mathcal{M}}$ we propose is

$$\bar{g}_x(\bar{\xi}_x, \bar{\eta}_x) = \text{Trace}(\bar{\xi}_x^T U \bar{\eta}_U) + \text{Trace}(\bar{\xi}_U \bar{\eta}_R) + \text{Trace}(\bar{\xi}_R \bar{\eta}_V), \quad (7)$$

where $\bar{\xi}_x, \bar{\eta}_x \in T_x\overline{\mathcal{M}}$ are any tangent vectors. From [18, Section 3.6.2], $\bar{g}_x$ is a valid Riemannian metric on the total space $\overline{\mathcal{M}}$. This metric captures the scaling required to make the level sets of the cost function more circular which leads to good preconditioning properties [13, 14, 22] while respecting the symmetry (3). The Riemannian metric $g_x$ on the quotient space is induced by the metric $\bar{g}_x$ [18, Section 3.6.2].

### 3 Optimization related ingredients

Here we briefly discuss various geometric notions on the Riemannian quotient manifold $\mathcal{M}$ and detail the computations that are necessary to implement an off-the-shelf nonlinear conjugate gradient algorithm. In Figure 1 we visualize our search space as a sequentially submersed space in $\mathcal{E} := \mathbb{R}^{n \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{m \times r}$. Further developments follow [18] Chapters 3, 5 and 8.

Figure 1: This schematic view of the search space allows us to derive various notions on the Riemannian quotient manifold $\mathcal{M}$ in a systematic way. Endowed with the Riemannian metric $\bar{g}$ [7], the quotient space is a Riemannian submersion. It is sufficient to describe the notions such as the Riemannian gradient in the space $\mathcal{E}$ endowed with the metric $\bar{g}$ [18]. The corresponding notions on the Riemannian submanifold and then afterwards on the Riemannian quotient manifold are obtained by orthogonal (in the metric sense) projections.

#### 3.1 Horizontal space

The matrix representation of the tangent space $\mathcal{T}_x\mathcal{M}$ of the abstract quotient manifold $\mathcal{M}$ is identified with a subspace of the tangent space of the total space $T_x\overline{\mathcal{M}}$ that does not produce a displacement along the equivalence classes. This subspace is called the horizontal space $\mathcal{H}_x\overline{\mathcal{M}}$ [18, Section 3.6.2]. In particular, we decompose the tangent space into two complementary subspaces as $T_x\overline{\mathcal{M}} = \mathcal{H}_x\overline{\mathcal{M}} \oplus \mathcal{V}_x\overline{\mathcal{M}}$. The vertical space $\mathcal{V}_x\overline{\mathcal{M}}$ is the tangent space to the equivalence class $[\bar{x}]$ that has the matrix expression $(U\Omega_1, R\Omega_2 - \Omega_1 R, V\Omega_2)$ where $\Omega_1$ and $\Omega_2$ are any skew-symmetric matrices of size $r \times r$ [18] Example 3.5.3]. Its complementary subspace according to the metric $\bar{g}_x$ is chosen as the horizontal space $\mathcal{H}_x\overline{\mathcal{M}}$. After a routine computation, the subspace has the following characterization,

$$\mathcal{H}_x\overline{\mathcal{M}} = \{\bar{\eta}_x \in T_x\overline{\mathcal{M}} : \begin{align*} &RR^TU^T\eta_U + \eta_R^T \text{ is symmetric,} \\ &R^T R V^T \eta_V - R^T \eta_R \text{ is symmetric.} \end{align*} \} \quad (8)$$
Once, the horizontal space is identified, any tangent vector \( \xi_x \in T_x \mathcal{M} \) on the quotient space is uniquely represented by a vector \( \bar{\xi}_x \) in the horizontal space \( \mathcal{H}_x \mathcal{M} \), also called its horizontal lift.

### 3.2 Projection operators

We require two projection operators: one from \( \mathcal{E} \) onto the tangent space of the total space \( T_x \mathcal{M} \), and a second projection operator that extracts the horizontal component of a tangent vector.

Given a matrix (with appropriate dimensions) in the space \( \mathcal{E} \), its projection onto the tangent space \( T_x \mathcal{M} \) is obtained by extracting the component normal (in the sense of \( \bar{g}_x \)) to the tangent space. The normal space \( N_x \mathcal{M} \) is, thus,

\[
\{(UN_1, \emptyset, VN_2) : N_2RR^T \text{ and } N_2R^TR \text{ are symmetric}\}
\]

for \( N_1, N_2 \in \mathbb{R}^{r \times r} \). Extracting the normal component is accomplished by using the linear operator \( \Psi_x : \mathcal{E} \rightarrow T_x \mathcal{M} \) that is shown below

\[
\Psi_x(Z_U, Z_R, Z_V) = (Z_U - UB(U(RR^T)^{-1})Z_R, Z_V - VB(V(R^TR)^{-1})Z_V)
\]

where \( B_U \) and \( B_V \) are symmetric matrices of size \( r \times r \) obtained by solving the Lyapunov equations,

\[
RR^TB_U + B_URR^T = RR^T(UZ_U + Z_U^TU)RR^T
\]

\[
R^TRB_V + B_VR^TR = R^TR(VZ_V + Z_V^TV)R^TR.
\]

(10)

\[
\text{Note that the Lyapunov equations can be solved computationally efficiently and in closed form by diagonalizing } R \text{ and performing similarity transforms on the variables } B_U \text{ and } B_V. \text{ If } R = \mathcal{P}\Lambda\mathcal{Q}^T \text{ is the singular value decomposition of } R \text{ then, } B_U \text{ and } B_V \text{ are linearly transformed to new variables } \tilde{B}_U \text{ and } \tilde{B}_V \text{ as } B_U = \mathcal{P}\tilde{B}_U \mathcal{Q}^T \text{ and } B_V = \mathcal{Q}\tilde{B}_V \mathcal{Q}^T. \text{ } \mathcal{P} \text{ and } \mathcal{Q} \text{ are orthogonal matrices of size } r \times r \text{ and } \Lambda \text{ is a diagonal matrix of size } r \times r \text{ with positive entries. Subsequently, (10) is written in terms of } \Lambda, \tilde{B}_U \text{ and } \tilde{B}_V \text{ which can be solved in closed form. The total cost of computing the solutions of (10) is } O(r^3). \text{ It should be emphasized that the projection (9) on } \text{St}(r, n) \text{ is not the one proposed in [18] Example 3.6.2}. \text{ A subsequent projection onto the horizontal space is obtained by the operator } \Pi_x : T_x \mathcal{M} \rightarrow \mathcal{H}_x \mathcal{M} \text{ defined as,}
\]

\[
\Pi_x(\bar{\xi}_x) = (\xi_U - U\Omega_1, \xi_R + \Omega_1R - R\Omega_2, \xi_V - V\Omega_2)
\]

(11)

where \( \Omega_1 \) and \( \Omega_2 \) are skew symmetric matrices of size \( r \times r \) obtained by solving the coupled system of Lyapunov equations that has the form

\[
RR^T\Omega_1 + \Omega_1RR^T - R\Omega_2R^T = \text{Skew}(U^T\xi_URR^T) + \text{Skew}(R\xi_R)
\]

\[
R^TR\Omega_2 + \Omega_2R^TR - R^T\Omega_1R = \text{Skew}(V^T\xi_VR^TR) + \text{Skew}(R^T\xi_R)
\]

(12)

where \( \text{Skew}(\cdot) \) extracts the skew-symmetric part of a square matrix, i.e., \( \text{Skew}(D) = (D - D^T)/2 \). The coupled equations (12) can also be solved efficiently and in closed form by diagonalizing \( R \) and performing similarity transforms on \( \Omega_1 \) and \( \Omega_2 \). The computational cost is \( O(r^3) \).
3.3 Retraction

A retraction is a mapping that maps vectors in the horizontal space to points on the search space $\mathbb{M}$ and satisfies the rigidity and first-order conditions [18, Definition 4.1.1]). It provides a natural way to move on the manifold in a search direction. The product nature of the total space $\mathbb{M}$ again allows us to choose a retraction by combining retractions on the individual manifolds [18, Example 4.1.3]

$$R_x(\xi_x) = (uf(U + \xi_U), R + \xi_R, uf(V + \xi_V))$$ (13)

where $\xi_x \in H_x$ and $uf(\cdot)$ extracts the orthogonal factor of a full column rank matrix, i.e., $uf(A) = A(A^T A)^{-1/2}$ which is computed efficiently by performing the singular value decomposition of $A$. The computational cost of a retraction operation is $O(nr^2 + nr^2 + r^3)$.

3.4 Vector transport

A vector transport $T_{\eta_x} \xi_x$ on a manifold $\mathcal{M}$ is a smooth mapping that transports a tangent vector $\xi_x \in T_x \mathcal{M}$ at $x \in \mathcal{M}$ to a vector in the tangent space at $R_x(\eta_x)$ satisfying the first-order approximation of transporting a vector along the geodesic [18, Definition 8.1.1]. Figure 1 also allows the vector transport to be given in terms of the projection operators [18, Sections 8.1.3 and 8.1.4]. The horizontal lift of the vector transport $T_{\eta_x} \xi_x$ is given as

$$\text{(}T_{\eta_x} \xi_x\text{)}_{R_x(\eta_x)} = \Pi_{R_x(\eta_x)}(\Psi_{R_x(\eta_x)}(\xi_x)),$$ (14)

where $\eta_x$ and $\xi_x$ are horizontal lifts of $\xi_x$ and $\eta_x$; and $\Pi(\cdot)$ and $\Psi(\cdot)$ are the projection operators defined in [13] and [12]. If $y = R_x(\eta_x)$ then, we could equivalently write $(T_{\eta_x} \xi_x)_{R_x(\eta_x)}$ as $(T_{y-x} \xi_x)y$. The computational cost of transporting a vector solely depends on forming and solving the Lyapunov equations [13] and [12] which costs $O(nr^2 + nr^2 + r^3)$.

4 R3MC: Algorithmic details

In the previous section, the matrix representations of various geometric notions on the quotient manifold $\mathcal{M}$ and their horizontal lifts have been presented. Here we give the Riemannian gradient formula and provide a way to compute an initial guess for the step-size. The concrete implementation of our nonlinear conjugate gradient algorithm R3MC is shown in Algorithm 1. The total computational cost per iteration of R3MC is $O(|\Omega|r + nr^2 + nr^2 + r^3)$ where $|\Omega|$ is the number of known entries. The convergence of R3MC to a critical point of (1) follows from the analysis in [18, Chapter 8].

4.1 Matrix representation of the gradient

Let $\tilde{\phi}(\bar{x}) = \|P_\Omega(URV^T) - P_\Omega(X^*)\|_F^2/|\Omega|$ be the mean least square cost function and $\phi$ be its induced function on the quotient manifold. We define an auxiliary sparse variable $S = 2(P_\Omega(URV^T) - P_\Omega(X^*))/|\Omega|$ that is interpreted as the Euclidean gradient of $\tilde{\phi}$ in the space $\mathbb{R}^n \times m$. The number of non-zero entries in $S$ is $|\Omega|$. The partial derivatives of the function
\section*{Algorithm 1 R3MC: Riemannian three-factor CG algorithm for matrix completion (1)}

\textbf{Input:} Initial iterate $\bar{x}_1 = (U_1, R_1, V_1) \in \bar{\mathcal{M}}$, tolerance $\tau > 0$, horizontal vector $\bar{\eta}_0 = 0$

\textbf{Output:} Sequence of iterates $\{\bar{x}_i\}$

\begin{algorithmic}[1]
  \REPEAT
    \STATE Compute the Riemannian gradient $\bar{\xi}_i = \text{grad} \bar{\phi}$ \hspace{1cm} \Comment{(15)}
    \STATE $\textbf{if}$ Check convergence, if $\bar{g}_{\bar{\xi}_i}(\bar{\xi}_i, \bar{\xi}_i) \leq \tau$ \textbf{then}
    \hspace{1cm} \textbf{break} \hspace{1cm} \Comment{(7)}
    \STATE Compute a conjugate direction by Polak-Ribiére (PR+)
    \hspace{1cm} $\bar{\eta}_i = -\bar{\xi}_i + \beta_i(\bar{T}_{\bar{x}_i,\bar{x}_i-\bar{x}_i-1}\bar{\eta}_i-1)\bar{x}_i$ \hspace{1cm} \Comment{(14) and (13) (8.28)}
    \STATE $\textbf{if} \ (\bar{g}_{\bar{\xi}_i}(\bar{\eta}_i, \bar{\xi}_i) > 0) \textbf{ then}$
    \hspace{1cm} $\bar{\eta}_i = -\bar{\xi}_i$ \hspace{1cm} \Comment{Reset}
    \STATE Determine an initial step $s_i$ from the linearized problem \hspace{1cm} \Comment{(17) or (18)}
    \STATE Find the smallest integer $p \geq 0$ such that
    \hspace{1cm} $\bar{\phi}(\bar{x}_i) - \bar{\phi}(R_{\bar{x}_i}(\frac{s_i}{\bar{p}}\bar{\eta}_i)) \geq 0.0001 s_i \bar{g}_{\bar{\xi}_i}(\bar{\eta}_i, \bar{\xi}_i)$ and
    \hspace{1cm} set $\bar{x}_{i+1} = R_{\bar{x}_i}(\frac{s_i}{\bar{p}}\bar{\eta}_i)$ \hspace{1cm} \Comment{(13)}
  \ENDREPEAT
\end{algorithmic}

$\bar{\phi}$ with respect to $U$, $R$ and $V$ are given in terms of $S$, i.e., $\frac{\partial \bar{\phi}(x)}{\partial x} = (SVR^T, U^TSV, S^TUR)$. Since the Euclidean space $\mathcal{E}$ is endowed with the scaled Riemannian metric $\tilde{\mathcal{E}}$, the expressions are further \textit{scaled} as $\frac{\partial \bar{\phi}(x)}{\partial U} = SVR^T(RR^T)^{-1}$, $\frac{\partial \bar{\phi}(x)}{\partial R} = U^TSV$ and $\frac{\partial \bar{\phi}(x)}{\partial V} = S^TUR(R^T\bar{R})^{-1}$. Finally, the horizontal lift of the Riemannian gradient $\text{grad}_x \bar{\phi}$ is given using the projection operator $\bar{\Psi} \bar{\mathcal{E}}$ [18] Section 3.6.1 and 3.6.2] and the (scaled) partial derivatives

\[
\text{grad}_x \bar{\phi} = \text{grad}_x \bar{\phi} = \bar{\Psi}_x (\frac{\partial \bar{\phi}(x)}{\partial x}) = (\frac{\partial \bar{\phi}(x)}{\partial U} - UB_U(RR^T)^{-1}, \frac{\partial \bar{\phi}(x)}{\partial R}, \frac{\partial \bar{\phi}(x)}{\partial V} - VB_V(R^T\bar{R})^{-1})
\]

\hspace{1cm} $(15)$

where $B_U$ and $B_V$ are solutions to the Lyapunov equations

\[
RR^T B_U + B_U RR^T = 2\text{Sym}(RR^T U^TSV^T),
\]

\[
R^T R B_V + B_V R^T R = 2\text{Sym}(R^T R V^T S^T U R).
\]

\hspace{1cm} $(16)$

Here $\text{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\text{Sym}(D) = (D + D^T)/2$. Solving $(15)$ costs $O(r^3)$. The total numerical cost of computing the Riemannian gradient depends on computing the partial derivatives $\partial \bar{\phi}(\bar{x})/\partial x$, i.e., on matrix products such as $SV$ which costs $O(|\Omega| r)$.

\subsection*{4.2 Initial guess for the step-size}

The least square nature of the matrix completion cost function also allows us to compute a linearized step-size guess efficiently [13] [13]. Given a search direction $\bar{\xi}_x \in \mathcal{H}_x \bar{\mathcal{M}}$ we solve the
following optimization problem to obtain a linearized step-size guess

$$s_* = \arg \min_s \|P_{\Omega}((U - s\xi)(R - s\xi)(V - s\xi)^T) - P_{\Omega}(X^*)\|_F^2.$$ \hspace{1cm} (17)

The above objective function is a degree 6 polynomial in $s$ and thus, the global minimum $s_*$ can be obtained numerically (and computationally efficiently) by finding the roots of its degree 5 derivative polynomial. However, this can be further approximated by considering a further degree 2 polynomial approximation, i.e.,

$$s_{acc} = \arg \min_s \|P_{\Omega}(URV^T + s\xi)(R + s\xi) + sUR\xi(V^T) - P_{\Omega}(X^*)\|_F^2$$ \hspace{1cm} (18)

that has a closed form solution. Computing $s_{acc}$ is about three times faster than computing $s_*$ \hspace{1cm} (17) with a numerical cost of $O(|\Omega|r)$.

### 4.3 Incrementing rank strategy

In problems where the rank is not known a priori, we employ the rank updating procedure of [12] that alternates between fixed-rank optimization and rank-one updates. The rank-one update is based on the idea of moving along the dominant rank-one projection of the negative gradient in the space $\mathbb{R}^{n \times m}$. If $(U, R, V)$ is a rank-$r$ matrix then, the rank-one update corresponds to $(U_+, R_+, V_+)$ such that $U_+R_+V_+^T = URV^T - \sigma uv^T$ where $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are the dominant left and right singular vectors of $S$ and $\sigma > 0$ is the dominant singular value. Total computational cost is $O(|\Omega| + nr^2 + mr^2 + r^3)$.

### 5 Numerical comparisons

We compare R3MC with the following state-of-the-art algorithms. For the first four algorithms we use their conjugate gradient implementations with linearized step-size guesses. For R3MC we use the accelerated step-size computation (18).

1. qGeomMC [13]: It is based on the factorization $X = GH^T$ and optimizes on the computational space $\mathbb{R}^{n \times r} \times \mathbb{R}^{n \times r}$ (full column rank matrices). It proposes a Riemannian quotient geometry with a scaled metric.
2. Polar Factorization [10, 16]: It considers a Riemannian quotient geometry based on the factorization model similar to ours but with the constraint that the factor $R$ is symmetric and positive definite.
3. ScGrassMC [14]: It reformulates the matrix completion problem on the bi-Grassmann search space $\text{Gr}(r, n) \times \text{Gr}(r, m)$ and proposes a scaled metric.
4. LRGeom [15]: It considers an embedded Riemannian submanifold geometry for the set of rank-$r$ matrices $\mathbb{R}^{n \times m}$. 
5. RTRMC [11]: It is based on the factorization model $X = UY^T$ where $U \in \text{St}(r, n)$ and $Y \in \mathbb{R}^{n \times r}$ and reformulates (1) as an optimization problem on the Grassmann manifold $\text{Gr}(r, n)$. It considers a second-order Riemannian trust-region method.
6. LMaFit [20]: It also relies on the factorization $X = GH^T$ to alternatively learn the matrices $X$, $G$ and $H$. It implements a tuned nonlinear successive over-relaxation algorithm.

All simulations are performed in Matlab on a 2.53 GHz Intel Core i5 machine with 4 GB of RAM. For all algorithms, we use the Matlab codes supplied by their authors. For each example considered here, a $n \times m$ random matrix of rank $r$ is generated as in [5]. Two matrices $A \in \mathbb{R}^{n \times r}$ and $B \in \mathbb{R}^{m \times r}$ are generated according to a Gaussian distribution with zero mean and unit standard deviation. The matrix product $AB^T$ then gives a random matrix of rank $r$. A fraction of the entries are randomly removed with uniform probability. The over-sampling ratio (OS) is ratio of the number of known entries to the matrix dimension, i.e., $OS = |\Omega|/(nr + mr - r^2)$. Without loss of generality, we consider only square matrices. For very rectangular matrices, say when $n \ll m$, one could truncate the number of columns to $\tilde{m}$ ($\approx n$) and accordingly solve the truncated matrix completion problem to learn the subspace $U$ of (2). Once $U$ is learned, the right subspace $V$ and the factor $R$ are learned by solving a least square problem, similar to the idea in [11]. The resulting $(U, R, V)$ provides a good initialization to the original rectangular problem.

![Figure 2](image1.png)  
(a) OS = 3  
(b) OS = 2.5  
(c) OS = 2.1

Figure 2: Case 1. The Riemannian geometries based on three-factor matrix factorization, including Polar Factorization, perform better under low sampling. R3MC is particularly efficient in a number of instances.

Stopping criteria. The maximum number of iterations of all except RTRMC is set to 500. For RTRMC, the maximum number of outer iterations is set at 200 (we expect a better rate of convergence) and the number of inner iterations (for the trust-regions sub-problem) is set to 100. Finally, all the algorithms are stopped if the objective function value is below
$10^{-20}$. The conclusions drawn are equally valid for lower accuracy. We initialize all the algorithms randomly.

**Case 1: low sampling.** We consider a moderate scale matrix of size $10000 \times 10000$ of rank 10. Three instances with different over-sampling have been considered. For larger values of OS, most of the algorithms perform similarly and show a nice behavior but with smaller OS ratios, the algorithms perform differently. In fact in Figure 2, for the case of OS = 2.1 only R3MC and Polar Factorization algorithms converged. It should be added, however, that second-order schemes like RTRMC exhibit a better behavior with non-random initialization, e.g., by taking $r$-dominant singular value decomposition of $P\Omega(X^*)$.

**Case 2: ill-conditioning.** We consider matrices of size $5000 \times 5000$ of rank 10 and impose an exponential decay of singular values. The ratio of the largest to the lowest singular value is known as the condition number (CN) of the matrix. At rank 10 the singular values with condition number 100 is obtained using the Matlab function `logspace(-2,0,10)`. The over-sampling ratio of these instances is 3. The matrix completion problem becomes challenging as the CN increases. We consider three different ill-conditioned matrices with different condition numbers. For CN = 100, only R3MC, LRGeom and RTRMC converge with RTRMC taking a much longer time. In Figure 3, R3MC outperforms both LRGeom and RTRMC.

![Figure 3](a) CN = 100  
(b) CN = 300  
(c) CN = 500

Figure 3: Case 2. Our proposed geometry is robust to different instances of ill-conditioned data.

**Case 3: low sampling + ill-conditioning.** In this test, we look into problem instances which result from both scarcely sampled and ill-conditioned data. The test requires completing relatively large matrices of size $25000 \times 25000$ of rank 10 with different condition numbers and OS ratios. In Figure 4 R3MC outperforms all other algorithms.
Case 4: ill-conditioning + rank-one updates. We create a random matrix of size $5000 \times 5000$ of rank 20 with exponentially decaying singular values so that the condition number is $10^{10}$ and OS 2 (computed for rank 10). In Figure 5 we show the mean square error obtained on a set of entries $\Gamma$ that is different from the set $\Omega$ (on which we optimize). First, in Figure 5(a) we compare the algorithms for rank 10 directly where R3MC shows a significantly better performance than others. The performance of RTRMC improves modestly with regularization. Second, in Figure 5(b) we use R3MC with the rank-one updating procedure (Section 4.3) starting from rank 1. It results in a better recovery at rank 10 and almost complete recovery at rank 17.

Figure 5: Case 4. R3MC along with the rank-one updates (Section 4.3) shows a much better performance for highly ill-conditioned data.
Case 5: MovieLens dataset. As a final test, we compare the algorithms on the MovieLens-1M dataset [23]. The dataset has a million ratings corresponding to 6040 users and 3952 movies. We perform 10 random 90/10 train/test partitions (i.e., validation on the 10% of the entries) of the ratings and average the results. In RTRMC we set the parameter $\lambda$ to $10^{-6}$ to avoid the error due to non uniqueness of the least square solution. Finally, the algorithms are run for 1000 iterations (200 for RTRMC). Table 1 shows the least mean square error (MSE) with standard deviations $\pm 10^{-5}$ for different ranks and with rank-one updates (ranks shown in brackets). For rank-one updating (Section 4.3), rank is updated when the test error starts to increase. RTRMC with rank-one updating did not give better results hence, is omitted. The least MSE for cases with and without rank updating have been shown in boldface. R3MC gave the second-best scores in both the cases.

Table 1: The mean square errors obtained on the 10% (test) entries of the MovieLens-1M dataset.

| Rank | R3MC | qGeomMC | Pol. Fac. | ScGrassMC | LRGeom | LMaFit | RTRMC |
|------|------|---------|-----------|-----------|--------|--------|-------|
| 5    | 0.7600 | 0.7692 | 0.7532    | 0.7748    | 0.7595 | 0.7636 | 0.7745 |
| 6    | 0.7583 | 0.7812 | 0.7525    | 0.7826    | 0.7628 | 0.7780 | 0.7931 |
| 7    | 0.7604 | 0.7835 | 0.7552    | 0.7966    | 0.7710 | 0.7838 | 0.7881 |
| Rank-updating MSE | **0.7315** (9) | 0.7379 (9) | 0.7337 (28) | **0.7271** (10) | 0.7334 (9) | 0.7382 (10) | - |

6 Conclusion

We have presented a conjugate gradient algorithm R3MC for low-rank matrix completion. The algorithm is the result of a novel three-factor Riemannian quotient geometry with a scaled Riemannian metric. Numerical examples on synthetic and real datasets suggest that our algorithm is robust to scarcely sampled and ill-conditioned problem instances of matrix completion, both of which are challenging exercises. All experiments suggest intrinsic robustness of algorithms that differentiate subspace and mass optimization through SVD like factorizations. We have no definite justification of this empirical observation.

Acknowledgements

We thank Paul Van Dooren for helpful discussions on Lyapunov equations. We also thank Bart Vandereycken, Nicolas Boumal and the RANSO group for useful discussions on the topic. 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

[1] J. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd International Conference on Machine learning, pages 713–719, 2005.

[2] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. Foundations of Computational Mathematics, 9:717–772, 2008.

[3] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. IEEE Transactions on Information Theory, 56(6):2980–2998, 2010.

[4] D. Gross. Recovering low-rank matrices from few coefficients in any basis. IEEE Transactions on Information Theory, 57(3):1548–1566, 2011.

[5] J. F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization, 20(4):1956–1982, 2010.

[6] Kiryung Lee and Yoram Bresler. Admira: Atomic decomposition for minimum rank approximation. arXiv:0905.0044v2, May 2009.

[7] L. Simonsson and L. Eldén. Grassmann algorithms for low rank approximation of matrices with missing values. BIT Numerical Mathematics, 50(1):173–191, 2010.

[8] P. Jain, R. Meka, and I. Dhillon. Guaranteed rank minimization via singular value projection. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, Advances in Neural Information Processing Systems 23, pages 937–945. 2010.

[9] R. Mazumder, T. Hastie, and R. Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. Journal of Machine Learning Research, 11(Aug):2287–2322, 2010.

[10] G. Meyer, S. Bonnabel, and R. Sepulchre. Linear regression under fixed-rank constraints: a Riemannian approach. In Proceedings of the 28th International Conference on Machine Learning (ICML), 2011.

[11] N. Boumal and P.-A. Absil. RTRMC: A Riemannian trust-region method for low-rank matrix completion. In Neural Information Processing Systems conference, NIPS, 2011.

[12] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. Technical report, arXiv.com, 2011.

[13] B. Mishra, K. Adithya Apuroop, and R. Sepulchre. A Riemannian geometry for low-rank matrix completion. Technical report, arXiv:1211.1550, 2012.

[14] T. T. Ngo and Y. Saad. Scaled gradients on Grassmann manifolds for matrix completion. In NIPS, pages 1421–1429, 2012.
[15] B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 2013.

[16] B. Mishra, G. Meyer, S. Bonnabel, and R. Sepulchre. Fixed-rank matrix factorizations and Riemannian low-rank optimization. Technical report, arXiv:1209.0430, 2012.

[17] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

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

[19] A. Edelman, T.A. Arias, and S.T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

[20] Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Mathematical Programming Computation*, 2012.

[21] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. *arXiv:1006.4046v1*, June 2010.

[22] Yu. Nesterov. *Introductory lectures on convex optimization: a basic course*. Number 87 in Applied Optimization. Kluwer Academic Publishers, 2003.

[23] http://www.grouplens.org/node/73.