May 2015

Rank-Constrained Optimization: A Riemannian Manifold Approach

Guifang Zhou
Florida State University

FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS AND SCIENCES

RANK-CONSTRAINED OPTIMIZATION: A RIEMANNIAN MANIFOLD APPROACH

By

GUIFANG ZHOU

A Dissertation submitted to the
Department of Mathematics
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2015

Guifang Zhou defended this dissertation on May 5, 2015.
The members of the supervisory committee were:

Kyle A. Gallivan
Professor Co-Directing Dissertation

Paul Van Dooren
Professor Co-Directing Dissertation

Adrian Barbu
University Representative

Giray Okten
Committee Member

Xiaoming Wang
Committee Member

The Graduate School has verified and approved the above-named committee members, and certifies that the dissertation has been approved in accordance with university requirements.

This work is dedicated to my parents and my family, for their constant encouragement.

ACKNOWLEDGMENTS

I would like to give my sincere thanks to the following individuals who, without their help, this dissertation would never have been completed.

First, I would like to thank my advisor, Professor Kyle A. Gallivan, for his guidance and support. His expertise in applied and computational mathematics improved my research skills and prepared me for future challenges. I thank my co-advisor, Professor Paul Van Dooren, who gave me detailed and delicate instructions in this dissertation. I would also like to thank my committee members, Professor Adrian Barbu, Professor Giray Okten, and Professor Xiaoming Wang, for their contributions and counsel on this dissertation.

Second, I give my great appreciations to my friends, for giving me lots of help in the process pursuing my Ph.D degree at FSU.

I am grateful to my parents, my fiance who gave me unselfish love and support in my life.

Finally, I appreciate the financial support from NSF Grant 1262476.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

ABSTRACT

This dissertation considers optimization problems on a Riemannian matrix manifold $\mathcal{M} \subseteq \mathbb{R}^{m \times n}$ with an additional rank inequality constraint. A novel technique for building new rank-related geometric objects from known Riemannian objects is developed and used as the basis for new approach to adjusting matrix rank during the optimization process.

The new algorithms combine the dynamic update of matrix rank with state-of-the-art rapidly converging and well-understood Riemannian optimization algorithms. A rigorous convergence analysis for the new methods addresses the tradeoffs involved in achieving computationally efficient and effective optimization. Conditions that ensure the ranks of all iterates become fixed eventually are given. This guarantees the desirable consequence that the new dynamic-rank algorithms maintain the convergence behavior of the fixed rank Riemannian optimzation algorithm used as the main computational primitive.

The weighted low-rank matrix approximation problem and the low-rank approximation approach to the problem of quantifying the similarity of two graphs are used to empirically evaluate and compare the performance of the new algorithms with that of existing methods. The experimental results demonstrate the significant advantages of the new algorithms and, in particular, the importance of the new rank-related geometric objects in efficiently determining a suitable rank for the minimizer.

CHAPTER 1

INTRODUCTION

In recent years, substantial progress has been made on the theory, design and efficient implementation of effective algorithms to solve optimization problems with constraints that specify a Riemannian manifold. Such problems are found in a wide variety of areas, but of particular interest in this dissertation are those involving matrix manifolds, e.g., the Grassmann manifold, the compact Stiefel manifold, symmetric positive definite matrices, symmetric positive semidefinite matrices with fixed rank, along with associated products and quotient manifolds. There are many important matrix-based optimization problems that have an additional constraint related to the rank of the optimal solutions [Mar11]. These problems have been solved, for the most part, in an ad hoc manner. This dissertation investigates optimization problems that involve a rank inequality constraint on a union of Riemannian manifolds. New algorithms are proposed, the theoretical properties that influence their convergence are analyzed and the efficiency and effectiveness of careful implementations on selected problems are demonstrated.

This chapter is organized as follows. In Section 1.1, an overview of optimization problems with rank constraints and a brief history of research on methods for solving rank-constrained optimization are given. Rank-constrained matrix approximation in two common constrained optimization problems is presented in Section 1.2. The chapter closes in Section 1.3 with an overview of the remainder of the dissertation.

1.1 The Problem of Rank-constrained Optimization

Euclidean rank-constrained matrix optimization problems have the form

$$\min f(X) \quad \text{subject to } X \in \mathcal{M}_{\leq k}, \tag{1.1}$$

where $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is a smooth objective function and $\mathcal{M}_{\leq k} = \{X \in \mathbb{R}^{m\times n} | \text{rank}(X) \leq k\}$, i.e., the set of matrices of rank at most k, In general, (1.1) is an NP-hard problem [VB94]. There are special cases, though, where an exact solution can be found, e.g., using the singular value

decomposition (SVD) [HJ90]. However, for many applications where the dimensions m and n are large or where there are significant time constraints, e.g., real-time or near real-time problems, calculating the SVD is impractical. Hence, a number of algorithms have focused on approaches that are faster than the SVD-based algorithm and require less memory, making them more suitable for such applications, see [DM05, DV06, DKM06, AM07].

Recently, optimization on manifolds has attracted significant attention as a general approach to reduce the dimension of optimization problems compared with solving the original problem in their ambient Euclidean space. Attempts have been made to understand (1.1) by considering a related but simpler problem

$$\min_{X \in \mathcal{M}_k} f(X). \tag{1.2}$$

where $\mathcal{M}_k = \{X \in \mathbb{R}^{m \times n} | \text{rank}(X) = k\}$, [Ye05, ABG07, MMBS13, JBAS10b, LKLS13]. Since $\mathcal{M}_k$ is a submanifold of $\mathbb{R}^{m \times n}$ of dimension $(m + n - k)k$, (1.2) can be solved using techniques from Riemannian optimization applied to matrix manifolds [AMS08].

However, a disadvantage of (1.2) is that the manifold $\mathcal{M}_k$ is not closed in $\mathbb{R}^{m \times n}$, which complicates considerably the convergence analysis and performance of an iteration. The solution may be on the boundary of $\mathcal{M}_k$, e.g., a singular matrix; or a convergent sequence $\{X_n\}$ generated by some optimization algorithm may include a singular matrix. Furthermore, simply approaching the boundary causes the smallest singular values become very small, leading to numerically undefined Hessian matrices precluding the use of some algorithms with superlinear or quadratic convergence, e.g., RTR-Newton algorithm [ABG07, Bak08].

Fortunately, the difficulty of (1.2) disappears when we consider the optimization problem (1.1) since the set $\mathcal{M}_{\leq k}$ is the closure of the set $\mathcal{M}_k$. However, $\mathcal{M}_{\leq k}$ is not a manifold, the gradient is not defined at singular points with $\text{rank}(X) < k$ since the set $\mathcal{M}_{\leq k}$ is no longer smooth at those points. Hence, algorithms for smooth manifolds are not applicable on $\mathcal{M}_{\leq k}$ directly.

To overcome this problem, alternating between fixed-rank optimization and a simple update to the rank has been employed in many papers [JBAS10a, MS13, MV14]. The optimization scheme is most often started with a rank-1 problem and after solving the associated local optimization on the fixed-rank manifold, a new problem is considered on a fixed-rank manifold with rank, typically, incremented by 1. However, this scheme is usually not efficient. Solving the optimization on each fixed-rank manifold is often computationally demanding if simple manifold algorithms are used

and finding an optimal point on the current fixed-rank manifold may not be required or useful if the optimal for the problem is considerably distant from that point. A combined perspective on applying a state-of-the-art Riemannian optimization algorithm to sufficiently reduce the local cost function and altering rank appropriately and rigorously is needed.

A family of random multistart-type algorithms, called Alternating Projections with Backtracking and Randomization, has been developed to solve the structured low-rank approximation problems arising in computational statistics [GZ13, GZ14]. This method can be viewed as a global random search extension of the alternating projection method. However, it has some shortcomings. First, it targets specifically the structured low-rank approximation problems. Second, there is no rigorous understanding of how to choose the values of parameters p (backtracking) and q (randomization) in the algorithm. Third, it does not increase the rank.

Very recently, a more global view of a simple basic line-search method on $\mathcal{M}_{\leq k}$ along with a convergence analysis has developed independently in [SU14]. The analysis generalizes ideas from the Euclidean projected gradient algorithm combined with an idea inspired by retraction on Riemannian manifolds in a manner similar to the discussion later in this dissertation. It is shown that the tangent cone of $\mathcal{M}_{\leq k}$ at the problematic singular points has a useful characterization, that supports the definition of line-search schemes using gradient-related search directions on tangent cones to achieve linear or sublinear convergence. Based on the explicit characterization of tangent cones, they extend the Riemannian optimization techniques from the smooth manifold of fixed rank to its closure. In [UV14], a rank-adaptive optimization strategy where local optimal solutions of some smaller rank are used as a starting point for an improved approximation with a larger rank is proposed. However, the rank increment is still a small fixed number each time, i.e. the rank is increased by 1 or 2 each time, which is not efficient and a convergence analysis is not given.

This dissertation addresses combining rank inequality constraints with a matrix manifold constraint in a problem of the form

$$\min_{X \in \mathcal{M}_{\leq k}} f(X) \tag{1.3}$$

where $\mathcal{M}_{\leq k} = \{X \in \mathcal{M} | \text{rank}(X) \leq k\}$ and $\mathcal{M}$ is a submanifold of $\mathbb{R}^{m \times n}$. Typical choices for $\mathcal{M}$ are the entire set $\mathbb{R}^{m \times n}$, a sphere, symmetric matrices, elliptopes, as well as spectrahedrons. The approach developed provides a more sophisticated use of higher order information, the geometry of

the manifolds involved, and extensions to recent advances in Riemannian optimization algorithms [Hua13].

1.2 Motivation and Applications

The rank-constrained optimization problems in the form of (1.1) have numerous applications and arise in diverse areas such as signal and image processing [MK97, JHSX11], system identification [FHB04, Mes98], computational finance [Wu02, ZW03], low dimensional embedding [LLR95]. Brief introductions to two common constrained optimization problems are given in the following sections.

1.2.1 Weighted Low-rank Approximation

Approximating a given data matrix with a matrix of acceptably low-rank is an important problem in data analysis. It is widely used for mathematical modeling and data compression. The rank constraint is related to a constraint on the complexity of a model that fits the data. In some cases, the deviation between the observed matrix and the low-rank approximation is measured relative to a weighted norm. Zero weights can be taken into account when some entries of the data matrix are missing or unknown. More generally, weights may be introduced in response to some external estimate of the noise variance associated with each measurement. For a 2-D filter design problem [LPW97], the matrix to be approximated is obtained via a sampling procedure and the number of samples and/or the expected variance vary among the entries. Setting the weights can discriminate between the important and unimportant elements of the data.

Finding a low-rank matrix which is an approximation to the given matrix with respect to a certain weighted norm is an optimization problem called *weighted low-rank approximation* and is formulated as: given a real matrix $R \in \mathbb{R}^{m\times n}$, a positive definite symmetric weighting matrix $W \in \mathbb{R}^{mn\times mn}$ and a positive integer $k < \min(m, n)$, find an $m-$by$-n$ matrix X_* with rank at most k that approximates R as closely as possible

$$X_* = \underset{X\in\mathbb{R}^{m\times n},\mathrm{rank}(X)\leq k}{\operatorname{argmin}} \|R - X\|_W^2, \tag{1.4}$$

where the weighted norm of an $m-$by$-n$ matrix A is defined as $\|A\|_W^2 = \mathrm{vec}\{A\}^T W \mathrm{vec}\{A\}$ and $\mathrm{vec}\{A\}$ stands for the vectorized form of A, i.e., a vector constructed by stacking the consecutive columns of matrix A in one vector. If the weight matrix W is an identity matrix, then the problem

(1.4) reduces to the well-known unweighted low-rank approximation problem. In practice, X_* is not always required. It is often the case that a matrix $\tilde{X}$ that approximates R well-enough with rank even lower than that of X_* is taken as the solution to the problem. The methods discussed in this dissertation are motivated, in part, by this practical consideration.

Unlike the unweighted low-rank approximation problem, the weighted low-rank approximation problem has received less attention in the literature. The few reasonable algorithms available at present are the alternating projection algorithm of [LPW97], gradient-based optimization methods developed in [MMH03] and a method due to Brace and Manton [BM06] that was presented as a heuristic but that is derived and analyzed rigorously in this dissertation using recent advances from [Hua13]. There are some algorithms for specific weighting matrices. For example, EW-TLS [PR02b] and GTLS [VV89] are used to solve weighted low-rank approximation problem when the weighting matrix W has a specific block diagonal structure.

1.2.2 Graph Similarity

When studying graphs and their structures, a common requirement is the ability to compare two graphs and quantitatively assess their similarity, i.e., given two graphs, answer the questions"How similar is each vertex in the first graph to each vertex in the second graph?" and "What is the best match for each vertex in the first graph to a vertex in the second graph?". One solution to these problems is the computation of a similarity matrix S. Graph similarity has applications in diverse fields such as image processing, biological networks, social networks and chemical compounds.

Certainly, the meaning of "similarity" is particular to the application. Many kinds of similarities have been considered, see [PDGM10] for more details about classification of similarity metrics. A large class of similarity algorithms take a very local perspective on similarity; namely, two nodes of two different graphs are considered "similar" if their neighboring nodes are "similar". This is a cyclic definition, and very naturally leads to iterative updates by which similarity scores between graph elements propagate to neighboring elements on each iteration. Blondel et al. in [BGH$^+$04] give such an iterative method to define a similarity matrix. However, when the graphs are large, their algorithm becomes computationally expensive.

The use of low-rank approximation to estimate the similarity matrix has been considered. Ideally, the problem would be formulated as: given adjacency matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$

of two graphs G_A and G_B with m and n nodes respectively, find an $m-$by$-n$ matrix S_* with rank at most k that approximates the real similarity matrix $S^{Blondel}$ as closely as possible

$$S_* = \underset{S\in\mathbb{R}^{m\times n},\|S\|_F=1,\mathrm{rank}(S)\leq k}{\operatorname{argmin}} \|S - S^{Blondel}\|_F \tag{1.5}$$

where $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2}$ denotes the *Frobenius norm* of a matrix $A \in \mathbb{R}^{m\times n}$. However, the similarity matrix $S^{Blondel}$ is not known and this formulation is not possible. Therefore, alternate definitions that in some sense are consistent with the similarity matrix $S^{Blondel}$ are proposed and associated algorithms derived. In [FND08], Fraikin et al. approach the similarity matrix defined by Blondel et al. by a rank$-k$ matrix with k identical singular values. Cason et al. in [CAD13] consider two kinds of low-rank approximations of the similarity matrix by using truncated SVD with either k nonzero identical singular values or at most k nonzero, not necessarily identical, singular values. More detail on these approaches is given in Chapter 5.

1.3 Research Overview and Thesis Statement

The main goal of this dissertation is the development, analysis and evaluation of a novel approach to optimization problems with rank inequality constraints combined with matrix manifold constraints, i.e., with constraint set $\mathcal{M}_{\leq k}$. This approach exploits the fact that $\mathcal{M}_{\leq k}$ is the union of fixed-rank manifolds, i.e.,

$$\mathcal{M}_{\leq k} = \{X \in \mathcal{M} | \mathrm{rank} \leq k\} = \bigcup_{0\leq r\leq k} \mathcal{M}_r, \tag{1.6}$$

where $\mathcal{M}_r = \{X \in \mathcal{M} | \mathrm{rank}(X) = r\}$ is assumed to be a manifold, and so each major step of the approach can exploit state-of-the-art rapidly converging Riemannian optimization algorithms on $\mathcal{M}_r$ and a dynamic update of the rank r using a line search.

This dissertation asserts the following thesis:

1. The geometric structure of the set $\mathcal{M}_{\leq k}$ supports the
   - identification of specific relevant geometric objects on each fixed-rank manifold $\mathcal{M}_r$;
   - development of a novel rank-related vector that defines a search direction on tangent cones;
   - definition of a novel rank-related retraction that facilitates the change from one fixed-rank manifold to another one.

2. These objects, direction and retraction can be used to develop a novel approach to solve optimization problems with rank inequality constraints.

3. The approach can exploit existing Riemannian optimization algorithms and the associated superlinear convergence and efficiency.

4. A systematic convergence theory for this approach can be developed that relates the behavior of the algorithms with respect to rank and cost function value, the parameter choices, convergence rate for solving the exact optimization problem, and convergence rate for solving an associated approximate optimization problem.

5. The efficiency and effectiveness of the approach can be demonstrated using two key applications of rank-inequality constrained optimization.

The remainder of this dissertation is organized as follows. Chapter 2 reviews important concepts for Riemannian manifolds and key optimization algorithms. In Chapter 3, a new approach to solve optimization problems with rank constraints is proposed and theoretical support is given. Chapter 4 discusses the application of the algorithms based on the new approach to weighted low-rank approximation problems of the form (1.4). In Chapter 5, the application of low-rank approximation of graph similarity matrix is discussed. Finally, Chapter 6 formulates the conclusions of this dissertation, summarizes the main contributions and indicates avenues for future research.

CHAPTER 2

REVIEW OF RIEMANNIAN OPTIMIZATION BASICS

This chapter reviews some important definitions and concepts of Riemanian manifolds that are extensively used in the dissertation. Additionally, the Riemannian optimization algorithms of interest are identified and characterized briefly.

2.1 Riemannian Geometry

Optimization on Riemannian manifolds (also called *Riemannian optimization*) concerns finding an optimum (global, or more reasonably, local) of a real-valued function f defined over a (smooth) manifold, and appears in a wide variety of computational problems in science and engineering. Roughly speaking, a manifold is a set endowed with coordinate patches that overlap smoothly.

Optimization on manifolds is usefully thought of as unconstrained optimization on a constrained space. The ideas of algorithms for unconstrained optimization on a Euclidean space have been adapted for optimization on manifolds. This required the careful reconsideration of many basic definitions, constructs and algorithmic techniques that cannot be extended simply from Euclidean space to a manifold. For example, addition of two points in Euclidean space is well-defined but does not extend to two points on manifold, in general.

A *manifold* is a topological space that resembles Euclidean space near each point. More precisely, each point of an d-dimensional manifold has a neighborhood that is homeomorphic to the Euclidean space of dimension d. In Riemannian geometry, a *smooth manifold* of dimension d is defined as a set $\mathcal{M}$ that locally looks like a d-dimensional Euclidean space but can be very different globally. Since optimization generally requires taking derivatives and gradients of a function, calculus on $\mathcal{M}$ must be performed. Therefore, a smooth structure on $\mathcal{M}$ that allows us to do the calculation is required. A *Riemannian manifold* is a real smooth manifold equipped with an inner product on the tangent space at each point that varies smoothly from point to point.

In this chapter, some ingredients of Riemannian manifolds and associated basic computations are reviewed followed by a summary of key Riemannian optimization algorithms. More detail can be found in [AMS08] and [Hua13].

2.1.1 Tangent Space and Tangent Vector

For a smooth manifold $\mathcal{M}$, the most intuitive way to define tangent vectors, i.e., directions of motion, is to use curves. Let $\gamma(t)$

$$\gamma : \mathbb{R} \to \mathcal{M} : t \mapsto \gamma(t).$$

be a smooth curve on $\mathcal{M}$. Given a smooth function f on $\mathcal{M}$, the function $f \circ \gamma : t \mapsto f(\gamma(t))$ is a smooth function from $\mathbb{R}$ to $\mathbb{R}$ with a well-defined classical derivative. This approach, combining curves and smooth functions on differentiable manifolds, allows the definition of a tangent vector. Let $\mathcal{F}_x(\mathcal{M})$ be the set of smooth functions defined on a neighborhood of $x \in \mathcal{M}$, a tangent vector is defined as follows.

Definition 1. *(*Tangent vector*). A tangent vector ξ_x to a manifold $\mathcal{M}$ at a point x is a mapping from $\mathcal{F}_x(\mathcal{M})$ to $\mathbb{R}$ such that there exists a curve γ on $\mathcal{M}$ with $\gamma(0) = x$, satisfying*

$$\xi_x f = \dot{\gamma}(0) f := \left. \frac{d(f(\gamma(t)))}{dt} \right|_{t=0}.$$

for all $f \in \mathcal{F}_x(\mathcal{M})$. Such a curve γ is said to realize the tangent vector ξ_x. The point x is called the foot of the tangent vector ξ_x.

The *tangent space* to $\mathcal{M}$ at x, denoted by $\mathrm{T}_x\mathcal{M}$, is the set of all tangent vectors to $\mathcal{M}$ at x. It is a linear space, i.e., closed under linear combination, with the same dimension as the manifold.

The *tangent bundle* $\mathrm{T}\mathcal{M}$ is defined as the union of the tangent spaces at all elements of $\mathcal{M}$:

$$\mathrm{T}\mathcal{M} := \bigcup_{x \in \mathcal{M}} \mathrm{T}_x\mathcal{M}.$$

A smooth *vector field* is a smooth mapping $\xi : \mathcal{M} \to \mathrm{T}\mathcal{M}$ that assigns to each point $x \in \mathcal{M}$ a tangent vector $\xi_x \in \mathrm{T}_x\mathcal{M}$, i.e.,

$$\xi : \mathcal{M} \to \mathrm{T}\mathcal{M}, x \mapsto \xi_x \in \mathrm{T}_x\mathcal{M}.$$

The set of all smooth vector fields on $\mathcal{M}$ is denoted by $\chi(\mathcal{M})$.

2.1.2 Riemannian Metric

The tangent space can be viewed as a vector space that approximates the manifold locally. A Riemannian metric defines angles and vector length in any tangent space of $\mathcal{M}$.

A *Riemannian metric* g is a correspondence between each point $x \in \mathcal{M}$ and an inner product $g_x : \mathrm{T}_x\mathcal{M} \times \mathrm{T}_x\mathcal{M} \to \mathbb{R}$. The following equivalent notation is used throughout this dissertation

$$g_x(\xi, \zeta) = g(\xi, \zeta) = \langle \xi, \zeta \rangle_x = \langle \xi, \zeta \rangle$$

to denote the inner product of two elements ξ, ζ of $\mathrm{T}_x\mathcal{M}$ and the subscript x is dropped when context makes it clear. The notation, flat $\flat$, is also used in the later sections. $\xi^\flat$ denotes a function from $\mathrm{T}_x\mathcal{M}$ to $\mathbb{R}$ that is $\xi^\flat \eta = g(\xi, \eta)$ for all $\eta \in \mathrm{T}_x\mathcal{M}$. This inner product, defined at each point on the manifold, turns each tangent space into an abstract Euclidean space capable of supporting a wide variety of algorithms. A *Riemannian manifold* is the combination $(\mathcal{M}, g)$.

Since a Riemannian metric provides an inner product on the tangent space, the norm induced by this inner product can be used to define a distance metric on $\mathcal{M}$ as follows:

$$d(x, y) = \inf_\gamma \left\{ \int_0^1 \sqrt{g_{\gamma(t)}(\dot\gamma(t), \dot\gamma(t))} dt \right\} = \inf_\gamma \left\{ \int_0^1 \|\dot\gamma(t)\|_{g_{\gamma(t)}} dt \right\}$$

where γ is a curve on $\mathcal{M}$ with $\gamma(0) = x$ and $\gamma(1) = y$. This definition of distance on the manifold allows a definition of neighborhoods on the manifold. The open ball of radius δ around x, denoted $B_\delta(x)$, is:

$$\mathcal{B}_\delta(x) = \{y \in \mathcal{M} : d(x, y) < \delta\}.$$

Finally, the idea of a neighborhood is used to define local minimizers for a function on a manifold. Given a function $f : \mathcal{M} \to \mathbb{R}$, a point x^* is a strict local minimizer if there exists some $\delta > 0$ such that

$$f(x^*) < f(y) \quad \text{for all } y \in \mathcal{B}_\delta(x^*).$$

2.1.3 Affine Connection, Geodesics, Exponential Mapping and Parallel Translation

Many algorithms in optimization require second-order information. In general, this second-order information is obtained by taking the derivative of one vector field with respect to another. In a Euclidean space, taking the derivative of one vector field along another one, i.e.,

$$\mathrm{D}\eta(x)[\xi_x] = \lim_{t \to 0} \frac{\eta(x + t\xi_x) - \eta(x)}{t}. \tag{2.1}$$

always returns a vector field. On a general Riemannian manifold $\mathcal{M}$, however, for vector fields ξ, η on $\mathcal{M}$, (2.1) need not be a vector field on $\mathcal{M}$ even if all the operations in the expression of the limit are well-defined. Therefore, the principle of taking derivatives of vector fields on manifold is generalized to the so-called affine connection.

Definition 2. *(*Affine Connection*). Let $\mathcal{F}_x(\mathcal{M})$ be the set of all smooth functions in $x \in \mathcal{M}$, $\chi(\mathcal{M})$ be the set of all smooth vector fields on $\mathcal{M}$. Then the affine connection is a smooth mapping, denoted by*

$$\nabla : \chi(\mathcal{M}) \times \chi(\mathcal{M}) \to \chi(\mathcal{M}) : \quad (\xi, \eta) \mapsto \nabla_\xi \eta$$

that satisfies the following properties: for all $f, g \in \mathcal{F}_x(\mathcal{M}), a, b \in \mathbb{R}$ and $\eta, \xi, \zeta \in \chi_x(\mathcal{M})$,

1. *$\nabla_{f\eta+g\zeta}\xi = f\nabla_\eta\xi + g\nabla_\zeta\xi : \mathcal{F}(\mathcal{M})$-linearity in the first argument η;*
2. *$\nabla_\eta(a\xi + b\zeta) = a\nabla_\eta\xi + b\nabla_\eta\zeta : \mathbb{R}$-linearity in the second argument ξ;*
3. *$\nabla_\eta(f\xi) = (\eta f)\xi + f\nabla_\eta\xi$: Product rule/Leibniz's law.*

Note that ηf denotes the application of the vector field η to the function f. For any smooth manifold $\mathcal{M}$, there are an infinite number of affine connections. For a Riemannian manifold $(\mathcal{M}, g)$, there exists a unique affine connection that satisfies two additional conditions:

1. symmetry: $(\nabla_\eta\xi - \nabla_\xi\eta)f = \eta(\xi f) - \xi(\eta f)$;
2. compatibility with Riemannian metric: $\zeta g(\eta, \xi) = g(\nabla_\zeta\eta, \xi) + g(\eta, \nabla_\zeta\xi)$,

for all $\eta, \xi, \zeta \in \chi_x(\mathcal{M})$. This affine connection ∇, called the *Riemannian connection* or *Levi-Civita connection* of $\mathcal{M}$.

A straight line in Euclidean space can now be generalized to a geodesic on a manifold. Let $(\mathcal{M}, g)$ be a Riemannian manifold with connection ∇. The parameterized curve $\gamma : (a, b) \to \mathcal{M}$ is called *geodesic* if and only if it is a curve with zero acceleration:

$$\nabla_{\dot\gamma(t)}\dot\gamma(t) := \frac{D^2}{dt^2}\gamma(t) = 0$$

for all t in the domain of γ. Note that different affine connections produce different geodesics. When the affine connection is the Riemannian connection, by virtue of its compatibility with the metric g, one of geodesics is also a length minimizing curve. This is consistent with the straight line in Euclidean space. In this dissertation, we only consider the Riemannian connection.

Given a point $x \in \mathcal{M}$ and a tangent vector $\eta \in \mathrm{T}_x\mathcal{M}$, there is a unique geodesic $\gamma(t; x, \eta)$ satisfying $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$. In addition, the geodesic also satisfies the homogeneity property $\gamma(t; x, a\eta) = \gamma(at; x, \eta)$. This unique curve defines the mapping

$$\mathrm{Exp}_x : \mathrm{T}_x\mathcal{M} \to \mathcal{M} : \eta \mapsto \mathrm{Exp}_x\eta = \gamma(1; x, \eta)$$

called the *exponential mapping* at x. If the domain of Exp_x is all of $\mathrm{T}_x\mathcal{M}$ for all $x \in \mathcal{M}$, the manifold $\mathcal{M}$ (endowed with the affine connection ∇) is termed *geodesically complete*. Exponential mapping gives a method to relate tangent vectors of x to elements in the neighborhood of x. For optimization algorithms, that may move around in the tangent space $T_x\mathcal{M}$ in order to select its next point in $\mathcal{M}$, the Exponential mapping is one way to map the chosen tangent vector back to manifold.

In many situations, e.g., for some Riemannian optimization algorithms, it is necessary to compare or combine tangent vectors in different tangent spaces. Since the affine connection provides the idea of differentiating tangent vectors in different tangent spaces, it can also be used to define moving a tangent vector from one tangent space to another. In a Euclidean space the simplest such motion is parallel translation that is simply moving the root of the given vector to any other point in the space to yield a parallel vector field. For a Riemannian manifold parallel translation produces a suitably generalized notion of a parallel vector field along a single curve. A vector field ξ on a curve γ satisfying $\frac{D}{dt}\xi = \nabla_{\dot{\gamma}}\xi = 0$ is called *parallel*. Given $a \in \mathbb{R}$ in the domain of γ and $\xi_{\gamma(a)} \in \mathrm{T}_{\gamma(a)}\mathcal{M}$, there is a unique parallel vector field ξ on γ such that $\xi(a) = \xi_{\gamma(a)}$. The operator $P_\gamma^{b \leftarrow a}$ sending $\xi(a)$ to $\xi(b)$ is called *parallel translation along* γ. In other words, we have

$$\frac{D}{dt}(P_\gamma^{t \leftarrow a}\xi(a)) = 0.$$

If ∇ is the Riemannian connection, the resulting parallel translation is an isometry.

2.1.4 Riemannian Gradient and Riemannian Hessian

Gradient-based optimization requires the notion of a gradient as the direction of steepest ascent of an objective function. Newton's method, requires additionally second-order information, the Hessian. In case of manifolds, these concepts have been generalized to the Riemannian setting as follows.

Definition 3. *(*Riemannian gradient*). Let f be a function defined on a Riemannian manifold $(\mathcal{M}, g)$. The Riemannian gradient of f at x, denoted as* $\mathrm{grad} f(x)$*, is the unique tangent vector in* $\mathrm{T}_x\mathcal{M}$ *satisfying*

$$\langle \mathrm{grad} f(x), \xi \rangle_x = \mathrm{D} f(x)[\xi], \quad \forall \xi \in \mathrm{T}_x\mathcal{M}, \tag{2.2}$$

where the directional derivative is denoted $\mathrm{D}f$ *and the definition of a tangent vector identifies* $\mathrm{D}f(x)[\xi] = \xi f$*.*

Definition 4. *(*Riemannian Hessian*). Given a real-valued function f on a Riemannian manifold $(\mathcal{M}, g)$, the Riemannian Hessian of f at a point x in the direction of $\eta \in \mathrm{T}_x\mathcal{M}$, denoted* $\mathrm{Hess} f(x)[\eta]$*, is the unique linear mapping*

$$\mathrm{Hess} f(x) : \mathrm{T}_x\mathcal{M} \to \mathrm{T}_x\mathcal{M}$$

that satisfies

$$\mathrm{Hess} f(x)[\eta] = \nabla_\eta \mathrm{grad} f(x), \tag{2.3}$$

for all $\eta \in \mathrm{T}_x\mathcal{M}$, where ∇ is the Riemannian connection chosen for $\mathcal{M}$.

From the symmetric property of Riemannian connection, we know Hessian is a self-adjoint operator with respect to the Riemannian metric, i.e.,

$$\langle \mathrm{Hess} f(x)[\eta], \xi \rangle_x = \langle \eta, \mathrm{Hess} f(x)[\xi] \rangle_x,$$

for all $\xi, \eta \in \mathrm{T}_x\mathcal{M}$.

2.1.5 Retraction and Vector Transport

Computationally efficient Riemannian optimization algorithms have been derived, analyzed and implemented in recent years by generalizing the notions of the Exponential mapping and parallel translation to retraction and vector transport respectively. The idea of retraction used here and in the analysis of the Riemannian optimization algorithms of interest is due to Shub [Shu86] (see also [ADM+02]).

Definition 5. *(*Retraction*). A smooth mapping $R : \mathrm{T}\mathcal{M} \to \mathcal{M}$ is said to be a retraction on $\mathcal{M}$ if, for every $x \in \mathcal{M}$, let R_x denote the restriction of R to $\mathrm{T}_x\mathcal{M}$ with the following properties.*

1. *$R_x(0_x) = x$, where 0_x denotes the zero element of $\mathrm{T}_x\mathcal{M}$.*

2. *With the canonical identification* $\mathrm{T}_{0_x}\mathrm{T}_x\mathcal{M} \simeq \mathrm{T}_x\mathcal{M}, R_x$ *satisfies* $\mathrm{D}R_x(0_x) = id_{\mathrm{T}_x\mathcal{M}}$*, where* $id_{\mathrm{T}_x\mathcal{M}}$ *denotes the identity mapping on* $\mathrm{T}_x\mathcal{M}$*.*

Retraction provides a potentially more efficient way to map a tangent vector in $T_x\mathcal{M}$ to an element in a neighborhood of x than the more constrained special case of the Exponential mapping. By creating a correspondence between the manifold and the tangent plane, a retraction also can be used to "lift" a function f defined on a manifold to the tangent plane as follows:

$$\hat{f}_x : \mathrm{T}_x\mathcal{M} \to \mathbb{R} : \eta \mapsto f(R_x(\eta)).$$

A vector transport is a map from one tangent space to another tangent space that is potentially more efficient than parallel translation.

Definition 6. *(*Vector Transport*). Vector transport on a manifold* $\mathcal{M}$ *is a smooth mapping*

$$\mathrm{T}\mathcal{M} \oplus \mathrm{T}\mathcal{M} \to \mathrm{T}\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_\S}(\xi_x) \in \mathrm{T}\mathcal{M}$$

satisfying the following properties for all $x \in \mathcal{M}$*.*

1. *(Associated retraction) There exists a retraction* R*, called the retraction associated with* $\mathcal{T}$*, such that the following diagram commutes*

$$\begin{array}{ccc} (\eta_x, \xi_x) & \xrightarrow{\mathcal{T}} & \mathcal{T}_{\eta_x}(\xi_x) \\ \downarrow & & \downarrow \pi \\ \eta & \xrightarrow{R} & \pi(\mathcal{T}_{\eta_x}(\xi_x)) \end{array}$$

   where $\pi(\mathcal{T}_{\eta_x}(\xi_x))$ *denotes the foot of the tangent vector* $\mathcal{T}_{\eta_x}(\xi_x)$*.*

2. *(Consistency)* $\mathcal{T}_{0_x}\xi_x = \xi_x$ *for all* $\xi_x \in \mathrm{T}_x\mathcal{M}$*.*

3. *(Linearity)* $\mathcal{T}_{\eta_x}(a\xi_x + b\zeta_x) = a\mathcal{T}_{\eta_x}(\xi_x) + b\mathcal{T}_{\eta_x}(\zeta_x)$*.*

Vector transport is called isometric if it also satisfies

$$g_{R(\eta_x)}(\mathcal{T}_{\eta_x}\xi_x, \mathcal{T}_{\eta_x}\zeta_x) = g_x(\xi_x, \zeta_x).$$

An important class of vector transports is vector transport by differentiated retraction which is a vector transport given by

$$\mathcal{T}_{\eta_x}\xi_x = \mathrm{D}R_x(\eta_x)[\xi_x];$$

i.e.,

$$\mathcal{T}_{\eta_x}\xi_x = \frac{d}{dt}R_x(\eta_x + t\xi_x)\Big|_{t=0},$$

where R is a retraction.

2.2 Riemannian Optimization Algorithms

The concept of optimization on manifolds can be traced back to the work of Luenberger [Lue72, Lue73] in the early 1970s and earlier where he views equality constraints as defining a surface in $\mathbb{R}^n$ and describes an idealized line search along geodesics on the surface. However, this approach is not computationally feasible, in general, and was not pursued largely for that reason. More importantly however, as has been shown recently in great detail, for many optimization algorithms on manifolds, an approximation of the geodesics is enough to guarantee the desired convergence properties.

The idea of carefully considering efficient computation has been investigated in several specific contexts. Gabay [Gab82] proposed a Newton method on embedded submanifold of $\mathbb{R}^n$ in 1982. He uses projective methods to compute a gradient vector tangent to the submanifold, computes a minimum in $\mathbb{R}^n$ along this direction, then projects the minimum point back onto the submanifold. Smith [Smi93] analyzed the optimization of differentiable functions on general Riemannian manifolds in 1993, generalized three algorithms (steepest descent, Newton's method and conjugate gradient method) onto Riemannian manifolds and proves their convergence. Many other efforts have also attempted to keep the computation required at acceptable levels, see [DPM02, EAS98, MM02, OW00, Man02, HT04].

While Riemannian Newton-like algorithms are able to achieve superlinear and quadratic convergence, there are some disadvantages: first, the Newton iteration requires the exact solution of a linear system at each step, which increases the computational cost. Second, there is no guarantee that the algorithm will converge to a local minimum. Without appropriate checks, it will converge to the closest critical point, which might be a local maximum, local minimum, or a saddle point.

Finally, the method may not even converge to stationary points, unless it satisfies some strong conditions, like the convexity of the cost function.

In 2008, the dissertation of C. Baker tames Riemannian Newton-like methods by developing a complete convergence theory, implementing a numerical library and analyzing the performance of a Riemannian trust-region family of methods (RTR-Newton) [Bak08] and [ABG07]. Riemannian trust-region methods construct a quadratic model of the objective function f around the current iterate and produce a candidate new iterate by (approximately) minimizing the model within a region where it is "trusted".

Baker's approach follows the "lift-solve-retract" procedure to solve the constrained problem. First, a retraction R is chosen on the Riemannian manifold $\mathcal{M}$ and used to "lift" the cost function f on $\mathcal{M}$ to a cost function $\hat{f}_x = f \circ R_x$ on the tangent space $\mathrm{T}_x\mathcal{M}$ for any point $x \in \mathcal{M}$. Since $\mathrm{T}_x\mathcal{M}$ is an Euclidean space, a quadratic model (trust-region subproblem) is then defined on $\mathrm{T}_x\mathcal{M}$ and a minimizer of the subproblem (or at least a point that sufficiently reduces the cost function) is computed by the "inverse-free" truncated conjugate-gradient method [Ste83]. This minimizer is then retracted back from $\mathrm{T}_x\mathcal{M}$ to $\mathcal{M}$ using R_x. This point is a candidate for the new iterate, which will be accepted or rejected depending on the quality of the agreement between the lifted cost function $\hat{f}$ and the function f itself. The approach requires the exact second-order term, i.e., the Hessian of f, or more usefully the action of the Hessian on a tangent vector (or a very good approximation) which may not be acceptable in terms of computational cost. Huang's work described below provides a solution to this computational cost difficulty.

The book by Absil et al. [AMS08] provides an excellent introduction to the area including a computationally-oriented description of the geometry of manifolds through the dissertation of Baker.

More recent efforts have concentrated on considering generalizing methods based on line-search-based Euclidean algorithms that achieve superlinear and quadratic convergence to a Riemannian setting in a systematic manner. For example, besides the Newton methods, quasi-Newton methods are extensively used on optimization problems in Euclidean spaces. They achieve superlinear convergence without computing the Hessian or a good approximation of the linear system defined by the Hessian. One of the most successful of these is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method and the associated restricted Broyden Family of methods.

In 2011, C.-H. Qi proposed and analyzed an approach to generalize BFGS method on Riemannian manifolds and developed the convergence analysis in her dissertation [Qi11]. The Riemannian generalization of the BFGS method combines the retraction-based ideas above with vector transport, which is used to connect different tangent spaces. The basic step is the following: given a smooth cost function on a Riemannian manifold $\mathcal{M}$ with Riemannian metric g, the Riemannian BFGS (RBFGS) defined the search direction $p_k \in \mathrm{T}_{x_k}\mathcal{M}$ at iteration k as the solution to the equation $\mathcal{B}_k p_k = -\mathrm{grad} f(x_k)$, where B_k is a linear operator that approximates the action of the Hessian in an appropriate direction and that is updated on each iteration. (As with Euclidean BFGS a version that propagates the inverse of B_k is also developed.) The new iterate point x_{k+1} is generated by an appropriate line search method with step size α_k, i.e., $x_{k+1} = R_{x_k}(\alpha_k p_k)$.

The most important aspect of the Riemannian BFGS algorithm is the manner in which $\mathcal{B}_k$ is updated since the classical update formulas used in Euclidean spaces have no meaning in a Riemannian manifold setting. Qi proposes the following update formula for $\mathcal{B}_k$ based on the vector transport $\mathcal{T}$ with associated retraction R to define a linear operator $\mathcal{B}_{k+1} : \mathrm{T}_{x_{k+1}}\mathcal{M} \to \mathrm{T}_{x_{k+1}}\mathcal{M}$,

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k},$$

where $a^\flat$ denotes the flat of a and $\mathcal{A}^*$ denotes the adjoint operator of $\mathcal{A}$, $s_k = \mathcal{T}_{\alpha_k p_k}(\alpha_k p_k)$, $y_k = \mathrm{grad} f(x_{k+1}) - \mathcal{T}_{\alpha_k p_k}(\mathrm{grad} f(x_k))$ and $\tilde{\mathcal{B}}_k = \mathcal{T}_{\alpha_k p_k} \circ \mathcal{B}_k \circ \mathcal{T}^{-1}_{\alpha_k p_k}$. The update formula for, $\mathcal{H}_k = \mathcal{B}_k^{-1}$ is

$$\mathcal{H}_{k+1} = \tilde{\mathcal{H}}_k - \frac{(\tilde{\mathcal{H}}_k^* y_k)^\flat s_k}{y_k^\flat s_k} - \frac{s_k^\flat (\tilde{\mathcal{H}}_k y_k)}{s_k^\flat y_k} + \frac{s_k y_k^\flat (\tilde{\mathcal{H}}_k^* y_k) s_k^\flat}{(y_k^\flat s_k)^2} + \frac{s_k^\flat s_k}{s_k^\flat y_k},$$

where $\tilde{\mathcal{H}}_k = \mathcal{T}_{\alpha_k p_k} \circ \mathcal{H}_k \circ \mathcal{T}^{-1}_{\alpha_k p_k}$. This approach offers the advantage that it is not necessary to solve a system of equations. Qi's dissertation includes a generalization of the Dennis and Moré condition to the Riemannian setting. However, Qi's convergence analysis is restricted the approach of BFGS on Riemannian manifold based on exponential mapping and parallel transport.

Ring and Wirth [RW12] improved on Qi's work with an approach to generalize BFGS to a Riemannian manifold in 2012. They consider an infinite dimensional manifold and prove superlinear convergence under some specific assumptions [RW12, Corollary13]. While not requiring exponential mapping and parallel vector transport, the analysis requires the use of a differentiated retraction as the vector transport which is usually computationally expensive.

Most recently, W. Huang's dissertation [Hua13] takes a very large step forward in the understanding and design of Riemannian quasi-Newton methods and computational efficiency for both line-search based algorithms and trust-region-based algorithms. Huang proposes a systematic generalization of three well-known unconstrained optimization approaches from Euclidean spaces to Riemannian manifolds: the Broyden family of methods, the symmetric rank-one trust region method and the gradient sampling method for both continuous and partly smooth cost functions. The dissertation includes a complete convergence theory, a comprehensive implementation strategy for library design (demonstrated by an implementation to support the empirical studies of the dissertation) and strategies for large scale problems for an appropriate subset of the methods.

As in the Euclidean case, the Riemannian Broyden family is defined by taking a linear combination of the Riemannian Davidon-Fletcher-Powell (DFP) and the Riemannian BFGS methods based on a parameter ϕ_k. Huang gives the formula of the important updates step as follows

$$\mathcal{B}_{k+1} = \tilde{\mathcal{B}}_k - \frac{\tilde{\mathcal{B}}_k s_k (\tilde{\mathcal{B}}_k^* s_k)^\flat}{(\tilde{\mathcal{B}}_k^* s_k)^\flat s_k} + \frac{y_k y_k^\flat}{y_k^\flat s_k} + \phi_k g(s_k, \tilde{\mathcal{B}}_k s_k) v_k v_k^\flat,$$

where $v_k = \frac{y_k}{g(y_k,s_k)} - \frac{\tilde{\mathcal{B}}_k s_k}{g(s_k,\tilde{\mathcal{B}}_k s_k)}$ and $\tilde{\mathcal{B}}_k = \mathcal{T}_{S_{\alpha_k p_k}} \circ \mathcal{B}_k \circ \mathcal{T}^{-1}_{S_{\alpha_k p_k}}$, $\mathcal{T}_S$ is an isometric vector transport. When $\phi_k = 0$, the Riemannian Broyden family of methods reduce to Riemannian BFGS methods. The restricted Riemannian Broyden Family is defined by convex combination and the update preserves the positive definiteness of the Hessian approximation when suitable restrictions are placed on the step size and vector transport. When the combination is not convex the family becomes the entire Riemannian Broyden Family. In the latter case, convergence behavior and the choice of ϕ_k is more involved as in the Euclidean case. The well-posedness of the Broyden Family (restricted and not restricted) and the convergence rate as a function of ϕ_k are analyzed.

The convergence theory includes several novel extensions. Riemannian Dennis and Moré conditions are developed that subsume that of Qi and characterizes the required correspondence between the action of B_k and the true Hessian to ensure superlinear convergence for optimization problems and the related, more general, problem of finding zeros of Riemannian vector fields. The theory also introduces a key result that allows superlinear convergence of the restricted Riemannian Broyden Family while avoiding the unacceptably large computational load of the differentiated retraction required by Ring and Wirth. This is the notion of the **locking condition** that specifies the relationship between the vector transport used to the differentiated associated retraction. The locking

condition and Riemannian Wolfe conditions are key to guaranteeing both superlinear convergence and well-posedness of the Riemannian Broyden Family. In general, the theory weakens the requirements on retraction and vector transport, thus subsumes the earlier Riemannian BFGS work of [Qi11] and [RW12], and extends significantly the understanding of Riemannian quasi-Newton methods.

In Euclidean spaces, the symmetric rank-1 (SR1) method is a member of the Broyden Family defined by a nonconvex combination. The update in SR1 method does not preserve the positive definiteness and was for a long time considered to be an ineffective method. However, the SR1, in fact, has a key difference from the Broyden Family updates: it provides better approximation of the action of the Hessian on the entire space, i.e. not just in the single search direction of Broyden's methods. As a result, SR1 underwent a revival for Euclidean optimization. Huang's dissertation generalizes this to the Riemannian setting and proposes combining Riemannian SR1 with a Riemannian trust-region method that makes use of all the directional information of Hessian approximation. The Riemannian symmetric rank-one trust region methods (RTR-SR1) is an efficient way to solve the problems. Its convergence analysis in the Riemannian setting when restricted to the Euclidean setting actually extends the Euclidean results in the literature. It also provides a way to avoid requiring the locking condition since a line-search approach is not taken.

For large scale problems, saving storage is required for a practical algorithm. Huang develops, analyzes and empirically evaluates limited-memory versions of RTR-SR1 and $RBFGS$, that only store a few vectors that implicitly represent the update $\mathcal{B}_k$. The exploitation of these methods for large problems are crucial when considering the problems in this dissertation.

Finally, to solve the optimization of partly smooth functions, Huang also generalized the gradient sampling methods from Euclidean spaces to Riemannian manifolds. We do not review this method here since the functions considered in this dissertation and the approaches taken do not require considering nonsmooth situations. However, as noted below, methods for partly smooth cost functions may be useful when attempting to exploit higher order information in ways different than those pursued in this dissertation.

The exploitation of these state-of-the-art optimization algorithms in the solution of rank-inequality constrained problems of the form (1.1) introduced in Chapter 1 on the fixed-rank manifolds is a main motivation for this dissertation.

CHAPTER 3

RANK INEQUALITY CONSTRAINED OPTIMIZATION METHODS

The main approach to solving optimization problems with rank inequality constraints and its analysis are presented in this chapter.

3.1 Problem Statement and the Tangent Cone

Combining rank inequality constraints with a matrix manifold constraint results in a problem of the form

$$\min_{X \in \mathcal{M}_{\leq k}} f(X) \tag{3.1}$$

where $\mathcal{M}_{\leq k} = \{X \in \mathcal{M} | \text{rank}(X) \leq k\}$. $\mathcal{M}$ is a submanifold in $\mathbb{R}^{m \times n}$. The problem (3.1) does not require the cost function defined on $\mathcal{M}$, however, it is assumed throughout that it is since many applications give such a cost function [MMH03, CAD13].

The Euclidean metric

$$g^E(A, B) = \langle A, B \rangle_F := \text{tr}(A^T B), \quad \text{for all } A, B \in \mathbb{R}^{m \times n}, \tag{3.2}$$

where "tr" denotes the trace of a matrix, is the simplest metric for $\mathbb{R}^{m \times n}$. The metric on $\mathcal{M}$ is taken to be endowed from $g^E(A, B)$ to turn the manifold $\mathcal{M}$ into a Riemannian manifold $(\mathcal{M}, g)$.

The set $\mathcal{M}_{\leq k}$ usually does not have a manifold structure. Generally speaking, any $X \in \mathcal{M}_{\leq k}$ with rank less than k does not have a tangent space (see Section 4.3). However, for every point in $\mathcal{M}_{\leq k}$, a tangent cone, an extension of the tangent space, always exists. The tangent cone $\text{T}_x Z$ of a set $Z \subset \mathbb{R}^d$ at a point $x \in Z$ consists of all rays that originate from x that can be written as the limit of a sequence of secants defined using a sequence of points $x_i \in Z \setminus \{x\}$ that converges to x. Specifically, a sequence of points $x_i \in Z \setminus \{x\}$ that converges to x defines a sequence of secants r_i originating at x and passing through x_i. The limit rays of the sequence of secants are elements of the tangent cone at x. Note that for a given sequence of secants there may be more than one limit ray. These objects, which are generalizations of tangent spaces to smooth submanifolds, were

first used by Whitney [Whi92] to study the singularities of real analytic varieties, and also play a fundamental role in geometric measure theory. Here they are used to study the set $\mathcal{M}_{\leq k}$.

According to [OW04], the tangent cone to $\mathcal{M}_{\leq k}$ at a point $X \in \mathcal{M}_{\leq k}$ is equal to the set of all smooth admissible directions for $\mathcal{M}_{\leq k}$ at X, i.e.

$$\mathrm{T}_X\mathcal{M}_{\leq k} := \left\{ \dot{\gamma}(0) : \gamma \text{ is an smooth curve on } \mathcal{M}_{\leq k} \text{ with } \gamma(0) = X, \right. \\ \left. \text{and } \gamma(t) \in \mathcal{M}_{\leq k}, \text{for all } t \geq 0 \right\}.$$

Given an inner product in $\mathcal{M}$, the normal cone can be defined. Under the inner product $\langle \cdot, \cdot \rangle_F$, the set

$$\mathrm{N}_X\mathcal{M}_{\leq k} := \{\zeta \in \mathrm{T}_X\mathcal{M} : \langle \zeta, \xi \rangle_F \leq 0, \forall\ \xi \in \mathrm{T}_X\mathcal{M}_{\leq k}\},$$

is the normal cone to $\mathcal{M}_{\leq k}$ at a point $X \in \mathcal{M}_{\leq k}$.

Remark 7. *The tangent cone of $\mathcal{M}_{\leq k}$, $k < \min\{m, n\}$ is not a convex set and it is not closed under the operation of vector addition, see examples in Chapter 4.*

Since a tangent cone at a point in the set $\mathcal{M}_{\leq k}$ is not necessarily closed under addition, the properties of a manifold clearly break down. These points of $\mathcal{M}_{\leq k}$ at which there a tangent space does not exist are those matrices with rank $r < k$. Note that these matrices are elements of the manifold $\mathcal{M}_r = \{X \in \mathcal{M} | \mathrm{rank}(X) = r\}$ and the tangent space $T_X\mathcal{M}_r$ is a subset of the tangent cone $T_X\mathcal{M}_{\leq k}$. Therefore, general Riemannian optimization algorithms cannot be reliably applied directly to $\mathcal{M}_{\leq k}$.

The set $\mathcal{M}_{\leq k}$ is equivalent to $\bigcup_{r \leq k} \mathcal{M}_r$. If $\mathcal{M} = \mathbb{R}^{m \times n}$, then $\mathcal{M}_r$ is a manifold (see e.g., [AAM14]). In order to avoid abusing the notation, $\mathbb{R}_r^{m \times n}$ denotes $\mathcal{M}_r$ when $\mathcal{M}$ is $\mathbb{R}^{m \times n}$. In general, it is unclear whether $\mathcal{M}_r$ is a manifold or not. In this dissertation, the following is assumed for the manifold $\mathcal{M}$.

Assumption 1. *The Riemannian manifold $\mathcal{M} \subseteq \mathbb{R}^{m \times n}$ with Riemannian metric $\langle \cdot, \cdot \rangle_F$ satisfies the following properties:*

*(**A.1**) $\mathcal{M}_r = \{X \in \mathcal{M} | \mathrm{rank}(X) = r\}$ is a manifold;*

*(**A.2**) the closure of $\mathcal{M}_r$ is a subset of or equal to $\mathcal{M}_{\leq r}$;*

(**A.1**) of Assumption 1, of course, must be checked for any particular problem. It is true for all of the problems considered in this dissertation. Note that $\mathcal{M}_r$ is the intersection of $\mathcal{M}$ and $\mathbb{R}_r^{m\times n}$. A sufficient condition for $\mathcal{M}_r$ to be a manifold is that the pair $\mathcal{M}$ and $\mathbb{R}_r^{m\times n}$ intersect transversally [GP10, Chapter 1]. (**A.2**) of Assumption 1 is not a strong assumption. For example, one frequently encountered situation is that $\mathcal{M}$ is a closed subset of $\mathbb{R}^{m\times n}$. (**A.2**) of Assumption 1 is true since the closure of $\mathcal{M}_r$ is a subset of or equal to the intersection of two closed sets, $\mathbb{R}_{\leq r}^{m\times n} = \{X \in \mathbb{R}^{m\times n} | \text{rank}(X) \leq r\}$ and $\mathcal{M}$ [Mun00].

3.2 A Tangent Cone Descent Algorithm

The literature on optimization on manifolds is large and growing, see Section 2.2 in Chapter 2. Since we assume each fixed-rank $\mathcal{M}_r$ is a manifold, those algorithms can be applied directly for a specific choice of r. The remaining issue is how to change from one fixed-rank manifold to another fixed-rank manifold. Line-search methods (or steepest descent methods) on a fixed-rank manifold $\mathcal{M}_r \subseteq \mathcal{M}$ are based on the update formula

$$X_{n+1} = R_{X_n}(t_n P_{\mathrm{T}_{X_n}\mathcal{M}_r}(\eta_n)), \tag{3.3}$$

where $t_n \geq 0$ is a step-size, $\eta_n \in \mathrm{T}_{X_n}\mathcal{M}$, $P_{\mathrm{T}_{X_n}\mathcal{M}_r} : \mathrm{T}_{X_n}\mathcal{M} \to \mathrm{T}_{X_n}\mathcal{M}_r$ is a projector. Therefore, $P_{\mathrm{T}_{X_n}\mathcal{M}_r}(\eta_n)$ is a search direction on the tangent space $\mathrm{T}_{X_n}\mathcal{M}_r$. R is a retraction of $\mathcal{M}_r$, which takes vectors from the tangent space back to manifold [AMS08].

For any point $X \in \mathcal{M}_{\leq k}$ with rank less than k, the tangent space does not exist but a tangent cone always does. Although the definition of tangent cone to a closed set at a point looks complicated, it has a simple explicit characterization for some particular $\mathcal{M}_{\leq k}$, see [SU14] (where $\mathcal{M} = \mathbb{R}^{m\times n}$) and [CAD13] (where $\mathcal{M}$ is a sphere). Let $X \in \mathcal{M}_{\leq k}$ have rank $r \leq k$. The tangent cone $\mathrm{T}_X\mathcal{M}_{\leq k}$ contains the tangent space $\mathrm{T}_X\mathcal{M}_r$. What is more, if $r < k$, it also contains the curves approaching X by points of rank greater than r, but not beyond k. Therefore, from the orthogonal decomposition and semi-continuity of matrix rank, we have

$$\mathrm{T}_X\mathcal{M}_{\leq k} = \mathrm{T}_X\mathcal{M}_r + \{\eta_{k-r} \in \mathrm{N}_X\mathcal{M}_r \cap \mathrm{T}_X\mathcal{M} | \text{rank}(\eta_{k-r}) \leq k - r\}. \tag{3.4}$$

Given this structure of tangent cone at points with rank $r < k$, the projection of a tangent vector $\eta \in \mathrm{T}_X\mathcal{M}$ onto it can be calculated. We consider a general line-search method on $\mathcal{M}_{\leq k} \subseteq \mathcal{M}$,

$$X_{n+1} = R_{X_n}(t_n P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\eta_n)), \tag{3.5}$$

where $\eta_n \in \mathrm{T}_{X_n}\mathcal{M}$, $P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}} : \mathrm{T}_{X_n}\mathcal{M} \to \mathrm{T}_{X_n}\mathcal{M}_{\leq k}$ is a projector. Therefore, $P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\eta_n)$ provides a search direction in the tangent cone $\mathrm{T}_{X_n}\mathcal{M}_{\leq k}$ at X_n that allows us to move to another fixed-rank manifold. That is, for an iteration $X_n \in \mathcal{M}_{\leq k}$ has rank $r < k$, any search direction $P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\eta_n)$ of the form (3.4) will increase, maintain or decrease the rank for the next iterate by $\mathrm{rank}(\eta_{k-r}) \leq k - r$. Rank decrease only happens at intersection points of a curve on the $\mathcal{M}_r$ and the boundary of $\mathcal{M}_r$, say $\mathcal{M}_{r-1}$. Therefore, a choice to lower rank results from a decision made in the selection of the step size for line search methods or trust region methods.

A retraction that takes vectors from tangent cone $\mathrm{T}_X\mathcal{M}_{\leq k}$ back to $\mathcal{M}_{\leq k}$ can be defined in a manner that is rank-related (see Section 3.4). Given an iterate $X_n \in \mathcal{M}_{\leq k}$ with rank $r < k$, a rank-increasing step is taken by determining a rank-related direction vector (see Section 3.4) $\eta_{X_n,\tilde{r}}$ with $r < \tilde{r} \leq k$ based on the projection, and given, e.g., the step size that satisfies appropriate conditions. The next iterate $X_{n+1} \in \mathcal{M}_{\tilde{r}} \subseteq \mathcal{M}_{\leq k}$ is computed by applying the rank-related retraction.

In [SU14], Schneider and Uschmajew, independently of this dissertation, defined a gradient-related line search method for problem (1.1). The idea is similar to the approach described above when $\mathcal{M} = \mathbb{R}^{m\times n}$. They use search directions in the tangent cone at points with rank less than k and a generalized retraction [SU14, Definition 2.4] to get the next iteration in $\mathcal{M}_{\leq k}$. They use a practical retraction defined as the best approximation by a matrix of rank at most k in the Frobenius norm, i.e.

$$X_{n+1} = R_{X_n}(\xi) \in \operatorname*{argmin}_{Y\in\mathcal{M}_{\leq k}} \|Y - (X_n + \xi)\|_F.$$

This is significantly simpler than the retractions discussed in Section 3.4.

Schneider and Uschmajew prove the convergence of their gradient-related projection methods on $\mathcal{M}_{\leq k}$ based on Łojasiewicz inequality [SU14, Theorem 3.9]. However, the convergence result relies on the assumption, often satisfied in practice, that the limit points have rank k. Under this assumption, a line-search method on $\mathbb{R}^{m\times n}_{\leq k}$ is ultimately the same as a line-search method on $\mathbb{R}^{m\times n}_{k}$. Linear convergence is nearly always observed in their numerical experiments, but the rates in their theorem are not explicit, i.e., it is between sublinear and linear. What is more, they do not provide an efficient way to update the rank. So essentially their algorithm is a steepest descent approach on $\mathcal{M}_k$ that ignores rank decreases and does not carefully handle rank increases.

3.3 Motivation for a New Approach

In practice, it is not easy to give a best choice of the constraint k. If k is taken too small, the resulting minimizer of the optimization problem may not be an acceptable solution to the associated application problem, e.g., the matrix may not be approximated well enough. Therefore, there is pressure to choose a sufficiently large k based on application-related knowledge or intuition. However, large k may increase unacceptably the computational cost since there is a tendency to use all of the degrees of freedom available to reduce the value of the cost function, e.g., larger rank approximations of a matrix tend to be better. This can happen even if the minimizer of the optimization problem has a rank significantly lower than the constraint k due to finite precision effects on rank estimation.

The first key factor of the efficiency of any algorithm for these optimization problems is therefore to be able to assess when increasing rank does not suitably increase the quality of the approximation and similarly to know when decreasing rank does not introduce unacceptable approximation error. A rigorous rank adaptation strategy must allow both of these considerations to be assessed efficiently and thereby to solve an associated approximate optimization problem.

The second key factor of the efficiency of any algorithm is a superlinear convergence rate that is, preferably, provable. Specifically, the analysis should identify aspects of the problems (exact or approximate) and algorithms that prevent or support the exploitation of recent algorithmic and theoretical advances in high-performance Riemannian optimization algorithms. Section 3.4 describes a fairly straightforward algorithmic approach that addresses all of these issues.

3.4 A Modified Riemannian Optimization Algorithm

As mentioned above, the basic approach has two components per major step. The first is, given a current point X with rank r, apply one of the efficient superlinearly convergent Riemannian optimization algorithms briefly reviewed in Chapter 2 using the necessary Riemannian geometric objects (tangent space, Riemannian gradient, retraction, Riemannian Hessian etc.) on the fixed-rank manifold $\mathcal{M}_r$ to produce a sequence of rank r matrices.

Due to (**A.2**) of Assumption 1, the matrices in the sequence on $\mathcal{M}_r$ might indicate convergence to a lower rank matrix, i.e., on a different submanifold. Therefore, the nearness to a lower rank matrix is monitored while producing the sequence. If a matrix is close enough to matrices of lower

rank, the iteration on $\mathcal{M}_r$ is stopped and a rank adjustment is considered. Otherwise, the sequence continues until an approximate optimal solution X_r^* on $\mathcal{M}_r$ is found, or a matrix at which the cost function has been reduced sufficiently from its value at X. A rank adjustment is then considered in the second component of the step. Note that the rank adjustment procedure is the same for both of these cases.

A rank adjustment decision considers the following two functions, one is the extension of (3.1) on $\mathcal{M}$, the other is the restriction of (3.1) on a fixed-rank manifold $\mathcal{M}_r$.

$$f_{\mathrm{F}} : \mathcal{M} \to \mathbb{R} : X \mapsto f_{\mathrm{F}}(X), \tag{3.6}$$

so that $f = f_{\mathrm{F}}\big|_{\mathcal{M}_{\leq k}}$ (we assume the extension is well-defined) and

$$f_r : \mathcal{M}_r \to \mathbb{R} : X \mapsto f_r(X), \tag{3.7}$$

so that $f_r = f|_{\mathcal{M}_r}$. Not all functions can be arbitrarily extended, but the extension and restriction of (3.1) are well-defined in all applications discussed in this dissertation.

Incrementing the rank is based on the angle between the gradient of f_{F}, denoted by $\mathrm{grad} f_{\mathrm{F}}(X)$, at the approximate optimal point X_r^* on the manifold $\mathcal{M}$ and the gradient of f_r, denoted by $\mathrm{grad} f_r(X)$, at the approximate optimal point X_r^* on the specific rank manifold $\mathcal{M}_r$. The angle between them is $\theta = \angle(\mathrm{grad} f_{\mathrm{F}}(X_r^*), \mathrm{grad} f_r(X_r^*)) = \arccos \frac{\langle \mathrm{grad} f_{\mathrm{F}}(X_r^*), \mathrm{grad} f_r(X_r^*) \rangle}{\|\mathrm{grad} f_{\mathrm{F}}(X_r^*)\| \|\mathrm{grad} f_r(X_r^*)\|}$, which is shown in Figure 3.1. If the angle is greater than some given angle θ_0 ($\epsilon_1 = \tan(\theta_0)$) and the difference between $\|\mathrm{grad} f_{\mathrm{F}}(X_r^*)\|$ and $\|\mathrm{grad} f_r(X_r^*)\|$ is larger than a threshold ϵ_2, the rank is increased but, of course, not beyond the boundary value of k.

Note the two parameters, ϵ_1 and ϵ_2, only provide information about increasing rank. The strategies of rank reduction will be discussed later since in the neighborhood of a point, there exists only points with rank equal to or greater than the rank of current point. Therefore, the local information cannot be used to reduce the rank.

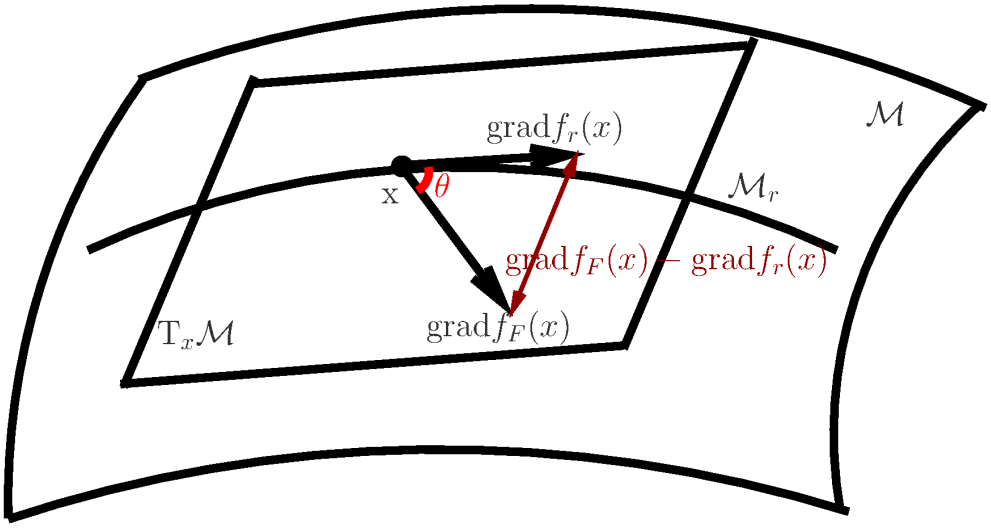

Figure 3.1: The plot of full gradient of a point on $\mathcal{M}$, $\mathrm{grad} f_{\mathrm{F}}(X)$, and the local gradient of a point on a fixed-rank manifold $\mathcal{M}_r$, $\mathrm{grad} f_r(X)$. θ is the angle between the two gradients and $\mathrm{grad} f_{\mathrm{F}}(X) - \mathrm{grad} f_r(X)$ represents the difference between $\|\mathrm{grad} f_{\mathrm{F}}(X)\|$ and $\|\mathrm{grad} f_r(X)\|$.

The two parameters ϵ_1, ϵ_2 are important for finding the exact/approximate solutions and controlling the computational efficiency of the method. A smaller ϵ_1 value makes it easier to increase rank per iteration. The smaller ϵ_2 is, the stricter the accuracy of the approximate local minimizer required. In particular, convergence to critical points of (3.1) is obtained if ϵ_2 is set to be zero.

Furthermore, there is a relationship between the two parameters. ϵ_1 can be seen as ϵ_2 scaled by the norm of local gradient. Therefore, when ϵ_2 approaches zero, ϵ_1 is not necessary approaching zeros since the norm of local gradient might be also small. Therefore, in real applications, ϵ_2 can be chosen small when ϵ_1 is not.

When the rank is increased, a rank-related retraction $\tilde{R}$ is required to determine the next point in the iteration.

Definition 8 (Rank-related retraction)**.** *Let $X \in \mathcal{M}_r$. A mapping $\tilde{R}_X : \mathrm{T}_X\mathcal{M} \to \mathcal{M}$ is a* rank-related retraction *if, $\forall \eta_X \in \mathrm{T}_X\mathcal{M}$, (i) $\tilde{R}_X(0) = X$, (ii) $\exists \delta > 0$ such that $[0, \delta) \ni t \mapsto \tilde{R}_X(t\eta_X)$ is smooth and $\tilde{R}_X(t\eta_X) \in \mathcal{M}_{\tilde{r}}$ for all $t \in [0, \delta)$, where $\tilde{r}$ is the integer such that $r \leq \tilde{r}$, $\eta_X \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}$, and $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\tilde{r}-1}$, (iii) $\frac{d}{dt}\tilde{R}_X(t\eta_X)|_{t=0} = \eta_X$.*

Note $\tilde{R}_X$ is not necessarily a retraction on $\mathcal{M}$ since it may not be smooth on the tangent bundle $\mathrm{T}\mathcal{M} := \bigcup_{X\in\mathcal{M}} \mathrm{T}_X\mathcal{M}$. The following lemma is used to show a rank-related retraction always exists.

Lemma 9. *Let $X \in \mathcal{M}$ be a matrix of rank r. For any $\tilde{r}$ that satisfies $r < \tilde{r} \leq k$, $\eta_X \in \mathrm{T}_X\mathcal{M}_{\leq\tilde{r}}$ but $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\leq\tilde{r}-1}$, there exists a smooth curve $\gamma(t), t \in [0,\delta), \delta > 0$ satisfying*

1. *$\gamma(0) = X$;*
2. *$\frac{d}{dt}\gamma(0) = \eta_X$;*
3. *the rank of $\gamma(t)$ is equal to $\tilde{r}$ for all $t \in (0,\delta)$.*

Proof. For any point $X \in \mathcal{M}_r$, assuming there is a matrix ΔX such that $\mathrm{rank}(X+\Delta X) = \tilde{r}$ and $X + \Delta X \in \mathcal{M}_{\tilde{r}}$, consider the differential equation on the manifold $\mathcal{M}_{\tilde{r}} \subset \mathcal{M}$

$$\begin{cases} \frac{d}{dt}\gamma(t) = \mathrm{P}_{\mathrm{T}_{\gamma(t)}\mathcal{M}_{\tilde{r}}}\eta_X \\ \gamma(0) = X + \Delta X. \end{cases} \tag{3.8}$$

where $\mathrm{P}_{\mathrm{T}_{\gamma(t)}\mathcal{M}_{\tilde{r}}} : \mathrm{T}_{\gamma(t)}\mathcal{M}_{\leq\tilde{r}} \to \mathrm{T}_{\gamma(t)}\mathcal{M}_{\tilde{r}}$ is a projector. By the analysis in [Hai01], there exists a unique solution $\gamma(t) \in \mathcal{M}_{\tilde{r}} \subset \mathcal{M}$ to equation (3.8) for a given initial value $\gamma(0) = X + \Delta X$ and $\mathrm{rank}(\gamma(t)) = \tilde{r}$.

Next, we need to show such a ΔX exists. Note that by definition of η_X, a curve $\tilde{\gamma}(t) \in \mathcal{M}_{\leq\tilde{r}}$ exists such that $\tilde{\gamma}(0) = X$, $\frac{d}{dt}\tilde{\gamma}(0) = \eta_X$ and $\eta_X \in \mathrm{T}_X\mathcal{M}_{\leq\tilde{r}}$, $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\leq\tilde{r}-1}$. Therefore, there exists a $\delta > 0$, such that for $t \in (0,\delta), \mathrm{rank}(\tilde{\gamma}(t)) \not< \tilde{r}$ and there must exist a sequence $\{t_i\}, t_i \to 0$ such that $\mathrm{rank}(\tilde{\gamma}(t_i)) = \tilde{r}$. Obviously, $\lim_{t_i\to 0}\tilde{\gamma}(t_i) = \tilde{\gamma}(0) = X$. Set $\Delta X_i = \tilde{\gamma}(t_i) - X$, to define a sequence $\Delta X_i \to 0$ such that $\mathrm{rank}(X + \Delta X_i) = \tilde{r}$.

On the other hand, considering $\Delta X = 0$, on the full manifold $\mathcal{M}$ the equation

$$\begin{cases} \frac{d}{dt}\gamma(t) = \mathrm{P}_{\mathrm{T}_{\gamma(t)}\mathcal{M}_{\tilde{r}}}\eta_X \\ \gamma(0) = X. \end{cases} \tag{3.9}$$

has a unique solution $\gamma_0(t) \in \mathcal{M}$.

Based on [Hai11, Theorem 3.3 (dependence on initial value)] and the construction above, we have a sequence $\gamma_i(t) \in \mathcal{M}$, where $\gamma_i(t)$ is the solution of the equation $\frac{d}{dt}\gamma(t) = \mathrm{P}_{\mathrm{T}_{\gamma(t)}\mathcal{M}_{\tilde{r}}}\eta_X$, $\mathrm{rank}(\gamma_i(t)) = \tilde{r}$ such that $\gamma(t) \to \gamma_0(t)$ for each $t \in (0,\delta)$, which implies $\mathrm{rank}(\gamma_0(t)) \leq \tilde{r}$, $\forall t \in (0,\delta)$.

Finally, to see the rank of $\gamma_0(t)$ can only be equal to $\tilde{r}$, assume first that there exists a sequence $\{t_i\} \subset (0,\delta)$, $t_i \to 0$, $\mathrm{rank}(\gamma_0(t_i)) = \tilde{r}$ and consider the following equation on $\mathcal{M}_{\tilde{r}} \subset \mathcal{M}$

$$\begin{cases} \frac{d}{dt}\gamma(t) = \mathrm{P}_{\mathrm{T}_{\gamma(t)}\mathcal{M}_{\tilde{r}}}\eta_x \\ \gamma(t_i) = \gamma_0(t_i). \end{cases} \tag{3.10}$$

Based on [Hai01], there is a unique solution $\gamma(t) \in \mathcal{M}_{\tilde{r}} \subset \mathcal{M}, t \in [t_i, \delta)$ for each t_i and $\text{rank}(\gamma(t)) = \tilde{r}$. As $t_i \to 0$, the rank is fixed, i.e., $\text{rank}(\gamma(t)) = \tilde{r}, \forall t \in (0, \delta)$, which is the desired result.

If the sequence $\{t_i\}$ above does not exist, there must exsit a $\tilde{\delta}$ such that for $t \in (0, \tilde{\delta})$, $\text{rank}(\gamma(t)) < \tilde{r}$. It must be the case that

$$\frac{d}{dt}\gamma(t) \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}-1}$$

and this implies

$$\frac{d}{dt}\gamma(0) = \eta_X \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}-1}$$

that contradicts the assumption $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}-1}$. Therefore, there must exist a curve $\gamma(t) \in \mathcal{M}$ with constant rank $\tilde{r}$ for all $t \in (0, \delta)$. □

Remark 10. *In fact, care must be taken because the rank of $\gamma(t)$ can be any number as long as it is greater than or equal to the constant $\tilde{r} = r + \Delta r$ (assuming $\tilde{r}$ is less than $\min(m, n)$). For example, if $\mathcal{M} = \mathbb{R}^{m\times n}$, given $X = U_r D_r V_r^T \in \mathcal{M}_r$, $\eta_X = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & E \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix}$, where $\text{rank}(E) = \Delta r$, obviously, $\eta_X \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}$ but $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}-1}$. The function $\gamma(t)$ can be written as follows and its rank is greater than $\tilde{r}$:*

$$\gamma(t) = \begin{bmatrix} U_r & U_{\Delta r} & U_{\tilde{r}\perp} \end{bmatrix} \left\{ \begin{bmatrix} D_r & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} + t \begin{bmatrix} \dot{D}_r & 0 & 0 \\ 0 & \dot{D}_{\Delta r} & 0 \\ 0 & 0 & 0 \end{bmatrix} + t^2 I_{m\times n} \right\} \begin{bmatrix} V_r^T \\ V_{\Delta r}^T \\ V_{\tilde{r}\perp}^T \end{bmatrix}.$$

The existence of such curves with rank $\tilde{r} + \Delta r$ means that when building the desired retraction it is necessary to make sure that the minimum rank is used to avoid excessive rank increase. While using the minimum rank increase is convenient, but not crucial, to proving convergence, it is very important for the computational efficiency of the resulting algorithms.

In general, for any $\eta_X \in \mathrm{T}_X\mathcal{M}$, there exists $\tilde{r}$ such that $\eta_X \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}$ but $\eta_X \notin \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}-1}$ since $\emptyset \subseteq \mathrm{T}_X\mathcal{M}_{\leq 1} \subseteq \cdots \subseteq \mathrm{T}_X\mathcal{M}_{\leq \min\{m,n\}} = \mathrm{T}_X\mathcal{M}$. We call such a vector a rank-$\tilde{r}$-related vector and denote it by $\eta_{X,\tilde{r}}$. The choice of $\tilde{r}$ is important since we want neither the rank increased too conservatively, i.e., only increased by a small amount, nor too aggressively, i.e., increased to the upper bound k directly. A reasonable $\tilde{r}$ can be obtained such that the angle between the full gradient, $\text{grad} f_{\mathrm{F}}(X)$, and the rank-$\tilde{r}$-related vector, $\eta_{X,\tilde{r}}$, is less than a certain value $\hat{\theta}$. Assume $\epsilon_4 = \tan(\hat{\theta})$, it is related to parameter ϵ_1, i.e., it cannot go beyond ϵ_1. By adjusting the value of ϵ_4, we are able to control the rank increment. The larger $\tilde{r}$ we want, the smaller ϵ_4 is set.

Since for any point $X \in \mathcal{M}_{\leq k}$, the tangent cone always exists, the projection of the full gradient onto it is well-defined. This motivates one way to obtain a rank$\leq \tilde{r}$-related vector.

Definition 11. *Let $X \in \mathcal{M}$ be a matrix with rank $r \leq \tilde{r}$. $\eta_{X,\tilde{r}}$ is a rank - $\leq \tilde{r}$-related vector at X if $\eta_{X,\tilde{r}} \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}$. Moreover, if $\eta_{X,\tilde{r}} \notin \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}-1}$, then it is a rank-$\tilde{r}$-related vector at X.*

Given $\xi \in \mathrm{T}_X\mathcal{M}$, one practical way to obtain a rank-$\leq \tilde{r}$-related vector $\eta_{X,\tilde{r}}$ is

$$\eta_{X,\tilde{r}} \in \mathrm{P}_{\mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}}(\xi) = \underset{\eta \in \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}}{\operatorname{argmin}} \|\xi - \eta\|_F, \tag{3.11}$$

where $\mathrm{P}_{\mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}} : \mathrm{T}_X\mathcal{M} \to \mathrm{T}_X\mathcal{M}_{\leq \tilde{r}}$ is a projector.

Schneider and Uschmajew define a general retraction from the tangent cone to the set $\mathcal{M}_{\leq k}$ [SU14, Definition 2.4]. They claim for every tangent vector $\eta_X \in \mathrm{T}_X\mathcal{M}$, there exists an analytic arc $\gamma : [0, \epsilon) \to \mathcal{M}$ such that $\eta_X = \dot{\gamma}(0)$. However, as stated in Remark 10, the arc γ is not unique. If $\mathcal{M} = \mathbb{R}^{m\times n}$, the arc γ is chosen with rank $\tilde{r}$ and η_X satisfying Lemma 9, then their definition is similar to Definition 8. Furthermore, if $\mathcal{M} = \mathbb{R}^{m\times n}$ and $\tilde{r} = k$, Definition 11 is equivalent to the definition in [SU14, Corollary 3.3]. However, as noted earlier for efficiency and numerical flexibility the rank-related retraction and vectors are preferred, especially when k is large.

Figure 3.2 shows the idea of rank-$\tilde{r}$-related vector and rank-$\tilde{r}$-related retraction. Given a tangent vector $\xi \in \mathrm{T}_X\mathcal{M}$, $\eta_{X,\tilde{r}} \in \mathrm{T}_X\mathcal{M}_{\tilde{r}}$ is a rank-$\tilde{r}$-related vector satisfies Definition 11. $R_X(\eta_{X,\tilde{r}})$ is a rank-$\tilde{r}$-related retraction.

Given a rank-related vector and retraction, the next point in the iteration is taken to be $X_{new} = \tilde{R}_X(t^*\eta_{X,\tilde{r}})$, where $\eta_{X,\tilde{r}}$ is a rank$\leq \tilde{r}$-related direction vector and t^* is the step size chosen using the well-known Riemannian form of Armijo's back-tracking procedure.

Definition 12. *(**Armijo Point** [AMS08]) Given a cost function f on a Riemannian manifold $\mathcal{M}$ with retraction R, a point $X \in \mathcal{M}$, a tangent vector $\eta \in \mathrm{T}_X\mathcal{M}$, and a scalar $\bar{\alpha} > 0, \beta, \sigma \in (0, 1)$, the Armijo point is $\eta^A = t^A\eta = \beta^m\bar{\alpha}\eta$, where m is the smallest nonnegative integer such that*

$$f(X) - f(R_X(\beta^m\bar{\alpha}\eta)) \geq -\sigma\langle \mathrm{grad} f(X), \beta^m\bar{\alpha}\eta\rangle_X.$$

The real t^A is the Armijo step size.

When the sequence of matrices produces by the iteration on $\mathcal{M}_r$ indicates by the rank and singular values of the matrices in the sequence that r should be decreased, a direction vector $\eta_{X,\tilde{r}}$ is

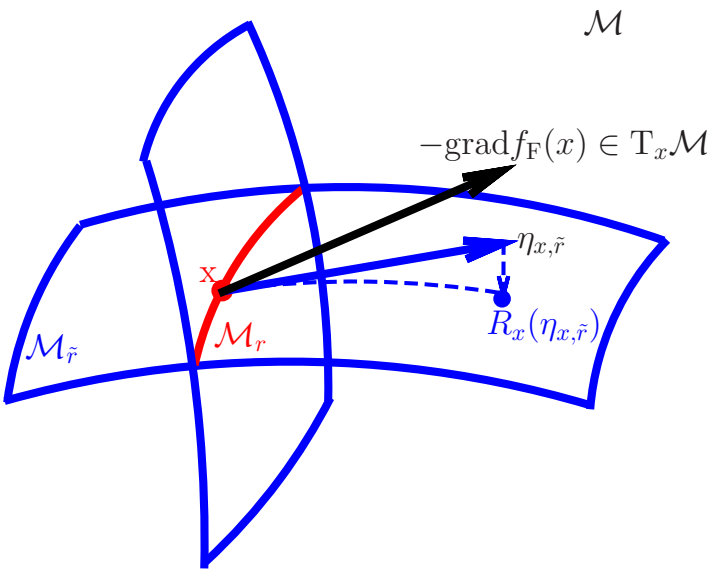

Figure 3.2: The plot of rank-related vector and rank-related retraction. $\mathcal{M}$ is a submanifold of $\mathbb{R}^{m\times n}$, $\mathcal{M}_r, \mathcal{M}_{\tilde{r}}$ are rank-r and rank-$\tilde{r}$ manifolds respectively. $X \in \mathcal{M}_r$, $\mathrm{grad} f_{\mathrm{F}}(X) \in \mathrm{T}_X\mathcal{M}$, $\eta_{X,\tilde{r}}$ is a rank-$\tilde{r}$-related vector and $R_X(\eta_{X,\tilde{r}}) \in \mathcal{M}_{\tilde{r}}$ is a rank-$\tilde{r}$-related retraction.

not required. The following two ways are considered for rank reduction, depending on whether the rank has been increased or not in any of the previous iteration. If the rank has not been increased in any previous iteration, given a new rank $\hat{r} < r$, the next iterate X_{new} is constructed by a projection of X defined by

$$X_{new} \in \operatorname*{argmin}_{\hat{X}\in\mathcal{M}_{\leq\hat{r}}} \|X - \hat{X}\|_F. \tag{3.12}$$

One practical way to find $\hat{r}$ is by examining the numerical Δ-rank of thin SVD of X defined in Algorithm 1 below.

We point out that each iterate is represented by three factors (see Chapter 4). Thus, the computation of SVD is avoided, which makes the realization of Algorithm 1 more efficient. If the rank has been increased before, for example, assume the latest rank increment was from X_i to X_{i+1}, then the next iterate X_{new} satisfies $f(X_{new}) - f(X_i) \leq c(f(X_{i+1}) - f(X_i))$, and the new rank $\hat{r} = \mathrm{rank}(X_{new})$. In this case, the iteration is updated based on the earlier information, i.e., the difference of the function values when the rank increased, which is more efficient than the simple truncation.

Algorithm 1 Determine the Δ_n numerical rank r

Require: An matrix $X \in \mathbb{R}^{m\times n}$ and a threshold Δ_n.

Ensure: Rank r.

```
Find the singular values σ_1 ≥ σ_2 ≥ ··· σ_{min{m,n}} ≥ 0 of matrix X;
r = 1;
for i = 2, ··· , min(m, n) do
    if σ_i/σ_1 > Δ_n then
        r ← r + 1;
    end if
end for
```

The modified Riemannian optimization approach is given by Algorithm 2.

For each application problem, the associated cost functions, the particular manifolds and, most importantly, the representations chosen are vital considerations in making this approach computationally efficient. For example, there is no need to repeatedly compute the SVD of a series of matrices. These crucial aspects of the success of the proposed approach are considered in the discussions of the application problems.

3.5 Convergence Analysis

The convergence properties of the Riemannian optimization algorithms used on each manifold $\mathcal{M}_r$ are understood and well-developed elsewhere [AMS08, Bak08, Hua13, Qi11]. However, given the added complexity of the rank changing discretely, the task of proving that the superlinear convergence is maintained for the rank inequality constrained problem requires additional theory. This section presents the analysis of the convergence properties of Algorithm 2.

3.5.1 Convergence Analysis for Exact Solution

For the convergence analysis, the concepts of a stationary point on $\mathcal{M}_{\leq k}$ and a radially L-C^1 function are required.

Definition 13. *([CAD13, SU14]) A point $X \in \mathcal{M}_{\leq k}$ is a stationary point of the cost function f if the gradient* $\mathrm{grad} f_{\mathrm{F}}(X)$ *belongs to* $\mathrm{N}_X\mathcal{M}_{\leq k}$*, the normal cone to $\mathcal{M}_{\leq k}$ at X, i.e.,*

$$\mathrm{grad} f_{\mathrm{F}}(X) \in \mathrm{N}_X\mathcal{M}_{\leq k} := \{\zeta \in \mathrm{T}_X\mathcal{M} : \langle \zeta, \xi\rangle_F \leq 0, \forall\ \xi \in \mathrm{T}_X\mathcal{M}_{\leq k}\}. \tag{3.13}$$

Algorithm 2 Modified Riemannian Optimization Algorithm

Require: A real-valued fuction f defined on $\mathcal{M}_{\leq k}$; A retraction R on a fixed-rank manifold and a rank-related retraction $\tilde{R}$, initial iterate $\tilde{X}_0 \in \mathcal{M}_{\leq k}$, $\epsilon_1 > 0$, $\epsilon_2, \in [0, 1)$, $\epsilon_3, \epsilon_4, \Delta_0 \in (0, 1)$;

Ensure: Sequence of iterates $\{X_n\}$.

Find the rank r by Algorithm 1 with input $\tilde{X}_0$ and Δ_0;
Set X_0 to be one of the solutions of $\operatorname{argmin}_{X \in \mathcal{M}_r} \|X - \tilde{X}_0\|^2$.
for n = 0,1, 2,... **do**
  Apply a Riemannian algorithm (e.g. one of GenRTR [ABG07], RBFGS [RW12, Hua13], RTR-SR1 [Hua13]) for cost function f_r over $\mathcal{M}_r$ with initial point X_n and stop at $\tilde{X}_n \in \mathcal{M}_r$, where either $\|\text{grad} f_r(\tilde{X}_n)\| < \epsilon_3$ ($\texttt{flag} \leftarrow 1$) or $\tilde{X}_n$ is close to $\mathcal{M}_{\leq r-1}$ ($\texttt{flag} \leftarrow 0$);
  if $\texttt{flag} = 1$ **then**
    if $\|\text{grad} f_{\text{F}}(\tilde{X}_n) - \text{grad} f_r(\tilde{X}_n)\| > \max\{\epsilon_1 \|\text{grad} f_r(\tilde{X}_n)\|, \epsilon_2\}$ **then**
      Set $\tilde{r}$ and η^* to be r and $\text{grad} f_r(\tilde{X}_n)$ respectively;
      while $\|\text{grad} f_{\text{F}}(\tilde{X}_n) - \eta^*\| > \epsilon_4 \|\eta^*\|$ **do**
        Set $\tilde{r}$ to be $\tilde{r} + 1$ and η^* to be a rank-$\tilde{r}$-related vector of $\text{grad} f_{\text{F}}(\tilde{X}_n)$ at $\tilde{X}_n$;
      end while
      Obtain X_{n+1} by applying an Armijo-type line search algorithm along η^* using a rank-related retraction;
    else
      If ϵ_3 is small enough, stop. Otherwise, $\epsilon_3 \leftarrow \tau \epsilon_3$, where $\tau \in (0, 1)$;
    end if
  else $\{\texttt{flag} = 0\}$
    If the rank has not been increased on any previous iteration, reduce the rank of $\tilde{X}_n$ based on (3.12) while keeping the function value decrease, update r, obtain next iterate X_{n+1};
    Else reduce the rank of $\tilde{X}_n$ such that the next iterate X_{n+1} satisfies $f(X_{n+1}) - f(X_i) \leq c(f(X_{i+1}) - f(X_i))$, where i is such that the latest rank increase was from X_i to X_{i+1}, $0 < c < 1$. Set r to be the rank of X_{n+1};
  end if
end for

Given a sequence $\{X_i\}_{i=0,1,\ldots} \in \mathcal{M}$, we say that X is an *accumulation point* or a *limit point* of the sequence if there exists a subsequence $\{X_{j_i}\}_{i=0,1,\ldots} \in \mathcal{M}$ that converges to X. The set of accumulation points of a sequence is called the limit set of the sequence.

Definition 14. *([AMS08]) Let $\hat{f}$ be defined as the pullback of f through a retraction R, i.e.,*

$$\hat{f} : \mathrm{T}\mathcal{M} \to \mathbb{R} : \xi \mapsto f \circ R(\xi).$$

*The function $\hat{f}$ is radially Lipschitz continuously differentiable (*radially L-C^1 function*) if there exist real $\beta_{RL} > 0$ and $\delta_{RL} > 0$ such that, for all $X \in \mathcal{M}$, for all $\xi \in \mathrm{T}\mathcal{M}$ with $\|\xi\| = 1$, and for all $t < \delta_{RL}$, it holds that*

$$\left| \frac{d}{d\tau} \hat{f}_X(\tau\xi)|_{\tau=t} - \frac{d}{d\tau} \hat{f}_X(\tau\xi)|_{\tau=0} \right| < \beta_{RL} t. \tag{3.14}$$

To support the flexibility to choose from different Riemannian optimization algorithms on the fixed-rank manifold (see Step 4 in Algorithm 2), it is assumed that all of the conditions in the relevant convergence analyses that ensure that iterates on the fixed-rank manifold converge globally are met.

Assumption 2. *The algorithm chosen in Step 4 has the property of global convergence. In other words, let $\{Z_n\}$ denote the sequence generated by the algorithm and f be a nonincreasing function on $\{X_n\}$. If the limit points of $\{Z_n\}$ are not rank deficient, i.e., ranks stay at r, then* $\liminf_{n\to\infty} \|\mathrm{grad} f_r(Z_n)\| = 0$.

Additionally, when considering an increase in rank, the analysis is on the manifold $\mathcal{M}$. Therefore, it is assumed the lifted function on $\mathrm{T}\mathcal{M}$ satisfies the following property.

Assumption 3. *Let f be a continuously differentiable function bounded below in $\mathcal{D} = \{X \in \mathcal{M}_{\leq k} | f(X) \leq f(X_0)\}$, $\mathcal{D}$ is compact. $\hat{f} : \mathrm{T}\mathcal{M} \to \mathbb{R} : \xi \mapsto f_{\mathrm{F}} \circ R(\xi)$ is a radially L-C^1 function with sufficient large δ_{RL} defined in Definition 14 such that for any $X, Y \in \mathcal{D}$, $\|R_X^{-1}(Y)\| < \delta_{RL}$.*

Lemmas 15, 16 and 17 are used to show the convergence properties of Algorithm 2 in Theorems 18 and 19.

Lemma 15. *Let X^* be a matrix with rank r. Let $\mathbb{R}^{m\times n}_{\leq(r-1)}$ be the set of all matrices with rank less than r. Then,*

$$\mathrm{Edist}(X^*, \mathbb{R}^{m\times n}_{\leq(r-1)}) = \min_{X\in\mathbb{R}^{m\times n}_{\leq(r-1)}} \|X^* - X\|_F > 0, \tag{3.15}$$

where $\mathrm{Edist}(X^*, \mathbb{R}^{m\times n}_{\leq(r-1)})$ *denotes the distance between X^* and $\mathbb{R}^{m\times n}_{\leq(r-1)}$ in Euclidean space.*

Proof. It is obvious that $\mathbb{R}^{m\times n}_{\leq(r-1)}$ is a closed set. Since the rank of X^* is r, $X^* \notin \mathbb{R}^{m\times n}_{\leq(r-1)}$. Thus,

$$\mathrm{Edist}(X^*, \mathbb{R}^{m\times n}_{\leq(r-1)}) = \min_{X\in\mathbb{R}^{m\times n}_{\leq(r-1)}} \|X^* - X\|_F > 0, \tag{3.16}$$

where $\mathrm{Edist}(X^*, \mathbb{R}^{m\times n}_{\leq(r-1)})$ denotes the distance between X^* and $\mathbb{R}^{m\times n}_{\leq(r-1)}$ in Euclidean space.

□

Lemma 16. *Let $X \in \mathcal{M}_{\leq k}$ be a matrix with rank k, then $\mathrm{T}_X\mathcal{M}_k = \mathrm{T}_X\mathcal{M}_{\leq k}$, i.e., on the boundary the tangent cone is a tangent space.*

Proof. Since $X \in \mathcal{M}_{\leq k}$ with rank k is an interior point of the set $\mathcal{M}_{\leq k}$, it implies $\mathrm{T}_X\mathcal{M}_k = \mathrm{T}_X\mathcal{M}_{\leq k}$.

□

Lemma 17. *Let $X \in \mathcal{M}$ be a matrix with rank r and $\mathrm{grad} f_{\mathrm{F}}(X)$ be the gradient of a cost function f_{F} at X on $\mathcal{M}$. If $\eta_{X,\tilde{r}}$ is a rank$\leq \tilde{r}$-related vector of $\mathrm{grad} f_{\mathrm{F}}(X)$ at X with $\tilde{r} > r$ then*

$$\langle \eta_{X,\tilde{r}}, \mathrm{grad} f_{\mathrm{F}}(X)\rangle = \|\eta_{X,\tilde{r}}\|_F^2$$

holds.

Proof. Assume $\eta_{X,\tilde{r}}$ is a rank$\leq \tilde{r}$-related vector of $\mathrm{grad} f_{\mathrm{F}}(X)$ at X, it follows from Definition 11 that $\eta_{X,\tilde{r}} \in \mathrm{T}_X\mathcal{M}_{\leq\tilde{r}}$.

Since the tangent cone is closed under multipliction by positive reals, $t\eta_{X,\tilde{r}} \in \mathrm{T}_X\mathcal{M}_{\leq\tilde{r}}, \forall t > 0$ holds. Furthermore, by Definition 11, $\eta_{X,\tilde{r}} \in \mathrm{argmin}_{\eta\in\mathrm{T}_X\mathcal{M}_{\leq\tilde{r}}} \|\mathrm{grad} f_{\mathrm{F}}(X) - \eta\|_F$ and therefore

$$\frac{d}{dt}\|\mathrm{grad} f_{\mathrm{F}}(X) - t\eta_{X,\tilde{r}}\|_F^2|_{t=1} = 0,$$

which implies

$$\langle \eta_{X,\tilde{r}}, \mathrm{grad} f_{\mathrm{F}}(X)\rangle = \langle \eta_{X,\tilde{r}}, \eta_{X,\tilde{r}}\rangle = \|\eta_{X,\tilde{r}}\|_F^2.$$

□

The global convergence of Algorithm 2 is given in the following theorem. It is well-known that an occasional steepest descent step is sufficient to guarantee global convergence in a Euclidean setting [NW06, p41]. This idea is adapted to the Riemannian situation.

Theorem 18. *Let $\{X_n\}$ be an infinite sequence of iterates generated by Algorithm 2. Suppose $\epsilon_2 = 0$ is chosen in Algorithm 2 and Assumptions 2 and 3 hold. Then* $\liminf_{n\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\| = 0$.

Proof. Assume $\{r_n\}$ is the rank sequence associated with $\{X_n\}$. If there exists a $K > 0$ such that $\{r_n\}_{n=K,K+1,\ldots}$ is a constant sequence, then the iterates remain on a manifold of matrices with fixed rank, denoted it by r^*. If r^* is less than k, then the full gradient, $\mathrm{grad} f_{\mathrm{F}}(X_n)$, on $\mathcal{M}$ must converge as $\mathrm{grad} f_{\mathrm{F}}(X_n) \to 0$. If this were not true, Algorithm 2 would increase the rank since this would cause descent in the cost function, contrary to $\{r_n\}_{n=K,K+1,\ldots}$ is a constant sequence. The conclusion holds.

If r^* reaches k, Algorithm 2 stays on the fixed-rank manifold $\mathcal{M}_k$ and since by Lemma 16, $\mathrm{T}_{X_*}\mathcal{M}_{\leq k} = \mathrm{T}_{X_*}\mathcal{M}_k$, i.e., the tangent cone is a tangent space and, based on convergence properties of any of the Riemannian optimization algorithms on the fixed-rank manifold, it follows that

$$\begin{aligned}\liminf_{n\to\infty} \|\mathrm{grad} f_k(X_n)\| &= \liminf_{n\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_k}\mathrm{grad} f_{\mathrm{F}}(X_n)\| \\ &= \liminf_{n\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}\mathrm{grad} f_{\mathrm{F}}(X_n)\| = 0.\end{aligned}$$

Now assume $\{r_n\}_{n=K,K+1,\ldots}$ is not a constant sequence for any $K > 0$ and let the subsequence $\{X_{n_j}\}_{n_j\in\mathcal{K}}$ denote the iterates that increase the rank, i.e., $\mathrm{rank}(X_{n_j+1}) > \mathrm{rank}(X_{n_j})$. According to Assumptions 2, the latest rank reduce iteration X_m satisfies $f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_m) \leq f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_{n_{j+1}})$. From Step 17 of Algorithm 2, the following holds

$$c(f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_{n_j+1})) \leq f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_m) \leq f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_{n_{j+1}}), \tag{3.17}$$

where $c \in (0, 1)$ is the coefficient defined in Step 17 of Algorithm 2. Therefore, $\{f_{\mathrm{F}}(X_{n_j})\}$ is nonincreasing when the rank reduces and the function is bounded below. Thus, the sequence of differences $f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_{n_{j+1}})$ must go to zero. What is more, from (3.17), $f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_{n_j+1})$ must go to zero as well.

By definition of Algorithm 2 (Step 11),

$$f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(X_{n_j+1}) \geq -c_1\sigma\alpha_{n_j}\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \eta^*_{n_j}\rangle_{X_{n_j}}.$$

where $\sigma \in (0,1)$, $\{\eta^*_{n_j}\}$ is an infinite subsequence produced in Step 11 of Algorithm 2.

Contradiction is used to show $\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \eta^*_{n_j} \rangle_{X_{n_j}} \to 0$. Suppose that $\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \eta^*_{n_j} \rangle_{X_{n_j}} \nrightarrow 0$, then it must be that $\{\alpha_{n_j}\}_{n_j \in \tilde{\mathcal{K}}} \to 0$, where $\tilde{\mathcal{K}}$ is a subsequence of $\mathcal{K}$. The α_{n_j}'s are determined from the Armijo rule, and it follows that for all n_j greater than some $\bar{n}, \alpha_{n_j} = \beta^{m_{n_j}} \bar{\alpha}$, where m_{n_j} is an integer greater than zero, $\bar{\alpha} > 0$ is a scalar. This means that the update $\frac{\alpha_{n_j}}{\beta} \eta^*_{n_j}$ did not satisfy the Armijo condition, hence

$$f_{\mathrm{F}}(X_{n_j}) - f_{\mathrm{F}}(\tilde{R}_{X_{n_j}}(\frac{\alpha_{n_j}}{\beta}\eta^*_{n_j})) < -\sigma \frac{\alpha_{n_j}}{\beta} \langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \eta^*_{n_j} \rangle_{X_{n_j}}, \quad \forall n_j \in \tilde{\mathcal{K}}, n_j \geq \bar{n}.$$

Denoting

$$\tilde{\eta}_{n_j} = \frac{\eta^*_{n_j}}{\|\eta^*_{n_j}\|} \text{ and } \tilde{\alpha}_{n_j} = \frac{\alpha_{n_j} \|\eta^*_{n_j}\|}{\beta},$$

the rank-related retraction $\tilde{R}$ is defined in Definition 8, the inequality above reads

$$\frac{\tilde{f}_{\tilde{\eta}_{n_j}}(0) - \tilde{f}_{\tilde{\eta}_{n_j}}(\tilde{\alpha}_{n_j})}{\tilde{\alpha}_{n_j}} < -\sigma \langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}}, \quad \forall n_j \in \tilde{\mathcal{K}}, n_j \geq \bar{n},$$

where $\tilde{f}_\eta(t) = f_F(\tilde{R}_X(t\eta))$ denotes a scalar function of t, for all $\eta \in \mathrm{T}_X\mathcal{M}$. The mean value theorem ensures that there exists $t_{n_j} \in [0, \tilde{\alpha}_{n_j}]$ such that

$$-\frac{d}{dt}\tilde{f}_{\tilde{\eta}_{n_j}}(t)|_{t=t_{n_j}} < -\sigma \langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}}, \quad \forall n_j \in \tilde{\mathcal{K}}, n_j \geq \bar{n}, \tag{3.18}$$

where the differential is taken with respect to the Euclidean structure on $\mathrm{T}_{X_{n_j}}\mathcal{M}$. Since $f_{\mathrm{F}} \in C^1$ is compact in $\mathcal{D}$ and $\|\eta^*_{n_j}\| \leq \|\mathrm{grad}_F f(X_{n_j})\|$, $\eta^*_{n_j}$ is upper bounded. Also, since $\{\alpha_{n_j}\}_{n_j \in \tilde{\mathcal{K}}} \to 0$, the convergence $\{\tilde{\alpha}_{n_j}\}_{n_j \in \tilde{\mathcal{K}}} \to 0$ follows. Because $f_{\mathrm{F}} \in C^1$, the gradient satisfies $\frac{d}{dt}\tilde{f}_{\tilde{\eta}_{n_j}}(t)|_{t=0} = \langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}}$. From (3.18), the inequality

$$\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}} - \frac{d}{dt}\tilde{f}_{\tilde{\eta}_{n_j}}(t)|_{t=t_{n_j}} < (1-\sigma) \langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}} \tag{3.19}$$

holds. Based on the assumption of the contradiction proof, there exists a constant $\mu > 0$ such that $\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}} < -\mu$. Since $\hat{f}$ is radially L-C^1, there exists a constant $C > 0$ such that

$$\left| \frac{d}{d\tau}\tilde{f}_{\tilde{\eta}_{n_j}}(\tau)|_{\tau=t_{n_j}} - \frac{d}{d\tau}\tilde{f}_{\tilde{\eta}_{n_j}}(\tau)|_{\tau=0} \right| = \| \langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \tilde{\eta}_{n_j} \rangle_{X_{n_j}} - \frac{d}{dt}\tilde{f}_{\tilde{\eta}_{n_j}}(t)|_{t=t_{n_j}} \| \leq C t_{n_j}.$$

By (3.19), it follows that

$$-C t_{n_j} < -(1-\sigma)\mu.$$

However, $\lim_{n_j\to\infty} t_{n_j} = 0$ since $t_{n_j} \in [0, \tilde{\alpha}_{n_j}]$ and $\tilde{\alpha}_{n_j} \to 0$, which implies $\mu < 0$ and the desired contradiction. Therefore, the convergence

$$\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \eta^*_{n_j} \rangle_{X_n} \to 0$$

follows. Based on Lemma 17, the convergence of $\langle \mathrm{grad} f_{\mathrm{F}}(X_{n_j}), \eta^*_{n_j} \rangle_{X_n} \to 0$ gives $\eta^*_{n_j} \to 0$, which means $\liminf_{n\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\| = 0$. □

The rank of matrices in the convergent sequence must satisfy certain properties. Intuitively, a convergent sequence of matrices cannot have a limit point with rank higher than the ranks of all X_n beyond a certain point, i.e., it is not convergent if there is such a jump, this was shown rigorously in Lemma 15. A sequence of matrices may have a limit point with lower rank, e.g., the limit point as $n \to \infty$ of $\{X_n\} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2^n} \end{pmatrix}$ is $\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$. However, note that in this case the rank never drops to r_* for any X_n beyond some point in the sequence. Therefore, it is not expected that the ranks of the X_n must eventually achieve the desired rank r_*. Thus, a local or asymptotic rank property must hold. Such a local rank property of the convergent sequence of Algorithm 2 to an isolated local minima is given in Theorem 19.

Theorem 19. *Suppose f_{F} is a C^2 function, X_* is a nondegenerate minimizer of f_{F} on $\mathcal{M}$, i.e., $\mathrm{grad} f_{\mathrm{F}}(X_*) = 0$ and $\mathrm{Hess} f_{\mathrm{F}}(X_*)$ is positive definite. Furthermore, it is an isolated minimizer of f on $\mathcal{M}_{\leq k}$. Let the rank of X_* be $r_* \leq k$ and denote the singular values of X_* by $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{r_*} > 0$. The value $\epsilon_2 = 0$ is used in Algorithm 2 to compute the sequence $\{X_n\}$. There exists a neighborhood $\mathcal{U}_{X_*} \in \mathcal{M}$ such that if $\{X_{n_j}\} \subset \{X_n\}$ is a subsequence with rank increasing, i.e., $\mathrm{rank}(X_{n_j+1}) > \mathrm{rank}(X_{n_j})$, and $\{X_{n_j}\}$ stay in $\mathcal{U}_{X_*}$, then if $\{X_{n_j}\}$ is a finite subsequence, $\liminf_{n\to\infty} \|\mathrm{grad} f_{\mathrm{F}}(X_n)\| = 0$, else $\lim_{j\to\infty} \|\mathrm{grad} f_{\mathrm{F}}(X_{n_j})\| = 0$.*

In addition, there exists $K > 0$ such that $\forall n > K, \quad \mathrm{rank}(X_n) \geq r_*$.

Proof. If $\{X_{n_j}\}$ is finite, according to Assumption 2 and following the proof of Theorem 18, the results can be obtained immediately.

Since f_F is a C^2 function on $\mathcal{M}$, $\mathrm{grad} f_{\mathrm{F}}(X)$ is a C^1 vector field on the Riemannian manifold $\mathcal{M}$. From [GQA12, Lemma 14.5], there exists a neighborhood $\hat{\mathcal{U}}_{X_*}$ of X_* and $C_0, C_1 > 0$ such that for all $X \in \mathcal{U}_{X_*}$,

$$C_0 \mathrm{dist}(X_*, X) \leq \|\mathrm{grad} f_{\mathrm{F}}(X)\| \leq C_1 \mathrm{dist}(X_*, X). \tag{3.20}$$

By Theorem 18, the sequence $\{X_{n_j}\}$ satisfies that $\liminf_{n_j\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}\mathrm{grad}f_{\mathrm{F}}(X_{n_j})\| = 0$. Since X_* is an isolated minimizer of f on $\mathcal{M}_{\leq k}$, there exists a neighborhood $\tilde{\mathcal{U}}_{X_*}$ such that if $X_{n_j} \in \tilde{\mathcal{U}}_{X_*}$,

$$\lim_{j\to\infty} \mathrm{dist}(X_*, X_{n_j}) = 0. \tag{3.21}$$

Thus, taking $\mathcal{U}_{X_*} = \hat{\mathcal{U}}_{X_*} \cap \tilde{\mathcal{U}}_{X_*}$, from (3.20), we can obtain if $X_{n_j} \in \mathcal{U}_{X_*}$,

$$\lim_{n_j\to\infty} \|\mathrm{grad}f_{\mathrm{F}}(X_{n_j})\| = 0. \tag{3.22}$$

Furthermore, since X_* is an isolated local minimizer with rank r_*, taking $0 < \delta < \mathrm{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq r_*-1})$, where $\mathrm{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq (r^*-1)})$ is defined in Lemma 15, there exists a neighborhood of X_*,

$$\mathcal{B}_\delta(X_*) = \{X | \|X - X_*\|_F < \delta\},$$

such that $X_n \in \mathcal{B}_\delta(X_*)$. If $\mathrm{rank}(X_n) < r_*$, based on Lemma 15, then it must be that $\|X_n - X_*\|_F > \mathrm{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq r_*-1})$. However, $X_n \in \mathcal{B}_\delta(X_*)$, which means $\|X_n - X_*\|_F < \delta < \mathrm{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq r_*-1})$, a contradiction. Thus, there exists $K > 0$ such that $\forall n > K, \quad \mathrm{rank}(X_n) \geq r_*$. □

Theorems 18 and 19 show the global and local convergence analyses for Algorithm 2 when $\epsilon_2 = 0$. Note that for these theorems because $\epsilon_2 = 0$, it is unlikely to stop updating rank. This is the consequence when we try to obtain the exact solution. The basic result about the rank of the matrices in a convergent sequence is generic. If the entire sequence is converging, i.e., not just a subsequence, then eventually the rank of the matrices X_n must remain at or above the rank of X_*.

The best case, of course, is when $\mathrm{rank}(X_n) = \mathrm{rank}(X_*)$ for $n > n_0$. In this case, the rate of convergence is inherited from the Riemannian optimization algorithm for the fixed rank manifold and can therefore be superlinear. When the ultimate convergent subsequence does not have rank equal to that of X_*, the iteration does not necessarily inherit the convergence rate of the Riemannian optimization algorithm for the fixed rank manifold.

Figure 3.3 gives an intuitive illustration of why the rank might increase while converging. No matter how close the iterates are to the stationary point X_* the angle between $\mathrm{grad}f_{\mathrm{F}}(X_i)$ and $\mathrm{grad}f_r(X_i)$ is not approaching zero. In other words, only using angle is not enough to guarantee fixed rank ultimately. In this case, the full gradient $\mathrm{grad}f_{\mathrm{F}}(X_i)$ is pushing a rank increase based on locally sound information but clearly globally, i.e., second order, ultimately misleading. Note that eventually, after possibly repeatedly increasing the rank, $\mathrm{grad}f_{\mathrm{F}}(X_i)$ points toward X_* and a

decrease in rank (possibly only in the limit) must take place. This indicates the clear danger, from a complexity point of view, in using exact solution reasoning and only first order information. As mentioned earlier, some sort of second order information could help address the problem. In fact, if the angle threshold is set correctly relative to the spectrum of the Hessian at X_*, assuming a C^2 cost function, the problem with rank increase can be avoided. However, generically, at present the complexity of the Hessian of f_{F} is unacceptable and one adapted appropriately to a tangent cone remains undeveloped and the subject of future work.

In the following convergence analysis, the approach of approximate optimization, as is done in practice for virtually all numerical optimization, is shown to restore the expected result that the iteration for the rank inequality constrained problem maintains the convergence rate of the fixed rank manifold algorithm.

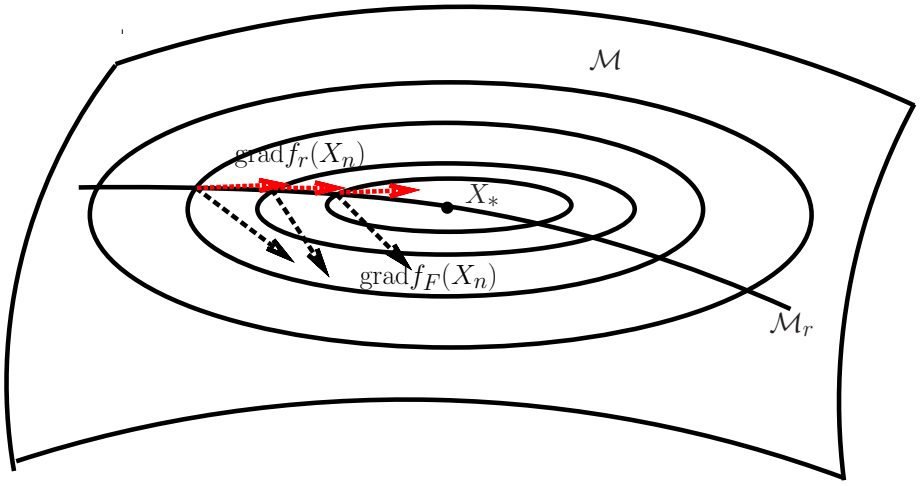

Figure 3.3: The plot of gradient of points on $\mathcal{M}$, $\mathrm{grad} f_{\mathrm{F}}$, and the gradient of points on a fixed-rank manifold $\mathcal{M}_r$, $\mathrm{grad} f_r$. The black dot line represents $\mathrm{grad} f_{\mathrm{F}}$ and the red dot line represents $\mathrm{grad} f_r$, the curve represents a fixed rank manifold $\mathcal{M}_r$, the circles represent the level sets of f_{F}, X_* represents a stationary point.

3.5.2 Convergence Analysis for Approximate Solution

In practice, the important behavior of the iteration is how quickly and reliably the size of the gradient can be reduced. In this section, the behavior of the iteration when reducing the size of the full gradient projected to the tangent cone, $P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))$, is analyzed and shown to

consistent with the behavior of the Riemannian optimization algorithm on the fixed rank manifold. This allows the iteration to keep the rank of the approximate solution as small as possible even when the true rank of the minimizer is larger and deals with the inevitable numerical noise affecting local rank. The result is given in Theorem 20. This can be used in concert with knowledge of the cost functions for problems discussed later to show that the value of the cost function at the approximate solution is also acceptably close to the value at the minimizer.

Theorem 20. *Let $\{X_n\}$ be an infinite sequence of iterates generated by Algorithm 2. If $\epsilon_2 \geq 0$ is used in Algorithm 2 and Assumptions 2 and 3 hold then*

$$\liminf_{n\to\infty} \|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\| \leq \left(\sqrt{1+\frac{1}{\epsilon_1^2}}\right)\epsilon_2.$$

Proof. If there exists an infinite subsequence $\{X_{n_j}\}$ from Algorithm 2 where the requirement on Step 6 does not hold, due to $\|\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_{n_j}) - \mathrm{grad} f_r(\tilde{X}_{n_j})\| > \epsilon_1 \|\mathrm{grad} f_r(\tilde{X}_{n_j})\|$ and $\|\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_{n_j}) - \mathrm{grad} f_r(\tilde{X}_{n_j})\| \leq \epsilon_2$, then

$$\|\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_{n_j}) - \mathrm{grad} f_r(\tilde{X}_{n_j})\| = \sin(\theta)\|\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_{n_j})\| \leq \epsilon_2,$$

where θ is the angle between $\mathrm{grad} f_{\mathrm{F}}(\tilde{X}_{n_j})$ and $\mathrm{grad} f_r(\tilde{X}_{n_j})$ and $\tan(\theta) \geq \epsilon_1$. Therefore,

$$\|\mathrm{grad} f_{\mathrm{F}}(X_{n_j})\| \leq \frac{1}{\sin(\theta)}\epsilon_2 = \left(\sqrt{1+\frac{1}{\tan^2(\theta)}}\right)\epsilon_2 \leq \left(\sqrt{1+\frac{1}{\epsilon_1^2}}\right)\epsilon_2.$$

If the subsequence is finite, then there exists a K such that the sequence $\{X_n\}, n > K$ does not use ϵ_2 and, therefore, by considering X_K to be the initial point of the iteration of interest the result follows by Theorem 18. □

In addition to relating the parameters of Algorithm 2 to the size of the projected gradient, it is also important to understand the effect of their values on the rank of the matrices in the sequence produced by the algorithm.

Theorem 21. *Suppose $f \in C^2$ and let $X_* \in \mathcal{M}_{\leq k}$ be an isolated local minimizer with rank r_* and the Riemannian Hessian of f_{F} at X_*, $\mathrm{Hess} f_{\mathrm{F}}(X_*)$, is positive-definite with minimal eigenvalue*

$\lambda_{\min} > 0$. *Then for any* $\nu > 0$, *there exists a neighborhood* $\mathcal{U}_\nu$ *of* X_* *such that for any* $X \in \mathcal{U}_\nu$, *the full gradient is bounded below by*

$$\|\mathrm{grad} f_{\mathrm{F}}(X)\|_F \geq (\lambda_{\min} - \nu)\|\eta\|_F, \tag{3.23}$$

where $\eta = \mathrm{Exp}_{X_*}^{-1} X$ *and* Exp *denotes the exponential mapping of* $\mathcal{M}$.

In addition, if there exists a $\tilde{\nu} < \lambda_{min}$ *such that* ϵ_2 *in Algorithm 2 satisfies* $\epsilon_2 < \frac{\epsilon_1}{\sqrt{1+\epsilon_1^2}}(\lambda_{min} - \tilde{\nu})\mathrm{Edist}(X_*, \mathcal{N}_r)$ *for some* $r \leq r_*$ *and the sequence* $\{X_n\}$ *generated by Algorithm 2 stays in* $\mathcal{U}_{\tilde{\nu}}$ *for all* $n > N$, *where* N *is an integer, then the rank of* X_n *eventually remains at least* r.

Proof. From [AMS08, Lemma 7.4.7], there exists a neighborhood $\mathcal{U}$ of X_*, $\forall X \in \mathcal{U}$,

$$\mathrm{P}_\gamma^{0\leftarrow 1}\mathrm{grad} f_{\mathrm{F}}(X) = \mathrm{Hess} f_{\mathrm{F}}(X_*)[\eta] + \int_0^1 (\mathrm{P}_\gamma^{0\leftarrow\tau}\mathrm{Hess} f_{\mathrm{F}}(\gamma(\tau))[\gamma'(\tau)] - \mathrm{Hess} f_{\mathrm{F}}[X_*][\eta])d\tau, \tag{3.24}$$

where $\mathrm{P}_\gamma^{0\leftarrow\tau}$ is the parallel translation, γ is the unique minimizing geodesic satisfying $\gamma(0) = X^*$ and $\gamma(1) = X$, $\eta = \mathrm{Exp}_{X_*}^{-1} X = \gamma'(0)$. Furthermore, according to [AMS08, Lemma 7.4.8],

$$\begin{aligned} &\| \int_0^1 (\mathrm{P}_\gamma^{0\leftarrow\tau}\mathrm{Hess} f_{\mathrm{F}}(\gamma(\tau))[\gamma'(\tau)] - \mathrm{Hess} f_{\mathrm{F}}[X_*][\eta])d\tau\|_F \\ &= \| \int_0^1 (\mathrm{P}_\gamma^{0\leftarrow\tau} \circ \mathrm{Hess} f_{\mathrm{F}}(\gamma(\tau)) \circ \mathrm{P}_\gamma^{0\leftarrow\tau} - \mathrm{Hess} f_{\mathrm{F}}[X_*][\eta])d\tau\|_F \\ &\leq \varepsilon(\mathrm{dist}(X, X_*))\mathrm{dist}(X, X_*), \end{aligned} \tag{3.25}$$

where $\lim_{\mathrm{dist}(X,X_*)\to 0} \varepsilon(\mathrm{dist}(X, X_*)) = 0$, $\mathrm{dist}(X, X_*)$ represents the distance between X and X_* on $\mathcal{M}$. Based on the definition of exponential mapping, $\mathrm{dist}(X, X_*) = \|\mathrm{Exp}_{X_*}^{-1}(X)\|_F = \|\eta\|_F$ holds and

$$\| \int_0^1 (\mathrm{P}_\gamma^{0\leftarrow\tau}\mathrm{Hess} f_{\mathrm{F}}(\gamma(\tau))[\gamma'(\tau)] - \mathrm{Hess} f_{\mathrm{F}}[X_*][\eta])d\tau\|_F \leq \varepsilon(\|\eta\|_F)\|\eta\|_F, \tag{3.26}$$

where $\lim_{\|\eta\|_F\to 0} \varepsilon(\|\eta\|_F) = 0$. Therefore, for any $\nu > 0$, there exists an $\epsilon_\nu > 0$ such that $\eta \in \mathcal{U}_\nu := \{Y | \mathrm{dist}(X_*, Y) < \epsilon_\nu\}$ implies $\varepsilon(\|\eta\|_F) < \nu$. Since the parallel translation is an isometry, (3.24) implies

$$\begin{aligned} \|\mathrm{grad} f_{\mathrm{F}}(X)\|_F &\geq \|\mathrm{Hess} f_{\mathrm{F}}(X_*)[\eta]\|_F - \| \int_0^1 (\mathrm{P}_\gamma^{0\leftarrow\tau}\mathrm{Hess} f_{\mathrm{F}}(\gamma(\tau))[\gamma'(\tau)] - \mathrm{Hess} f_{\mathrm{F}}[X_*][\eta])d\tau\|_F \\ &\geq \lambda_{min}\|\eta\|_F - \varepsilon(\|\eta\|_F)\|\eta\|_F. \end{aligned} \tag{3.27}$$

Thus, the following bound holds for any $X \in \mathcal{U}_\nu$:

$$\|\mathrm{grad} f_{\mathrm{F}}(X)\|_F \geq (\lambda_{\min} - \nu)\|\eta\|_F. \tag{3.28}$$

Next, we prove the rank of $\{X_n\}$ remains at least r eventually. Assuming for any ν, the rank of X_n is less than r for all $n > N$, where N is an integer, Lemma 15 implies

$$\text{dist}(X_n, X_*) \geq \|X_n - X_*\|_F > \text{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq(r-1)}), \tag{3.29}$$

where $\text{dist}(X_n, X_*)$ denotes the distance between X_n and X_* on $\mathcal{M}$. Since $\text{dist}(X_n, X_*) = \|\text{Exp}^{-1}_{X_*}(X_n)\|_F = \|\eta_n\|_F$, (3.29) implies

$$\|\eta_n\|_F > \text{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq(r-1)}). \tag{3.30}$$

Let $\text{rank}(X_n) = \hat{r}$. By assumption, $\hat{r} < r$. Assume the angle between $\text{grad}f_{\hat{r}}(X_n)$ and $\text{grad}f_{\text{F}}(X_n)$ is θ, then

$$\|\text{grad}f_{\text{F}}(X_n) - \text{grad}f_{\hat{r}}(X_n)\|_F = \sin(\theta)\|\text{grad}f_{\text{F}}(X_n)\|_F \geq \frac{\epsilon_1}{\sqrt{1+\epsilon_1^2}}(\lambda_{\min} - \nu)\text{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq(r-1)}). \tag{3.31}$$

The rank is increased on Step 6 in Algorithm 2 if

$$\|\text{grad}f_{\text{F}}(X_n) - \text{grad}f_{\hat{r}}(X_n)\|_F > \max\{\epsilon_1\|\text{grad}f_{\hat{r}}(X_n)\|_F, \epsilon_2\}.$$

Thus, there exists a $\tilde{\nu} \in (0, \lambda_{\min})$ such that if $X_n \in \mathcal{U}_{\tilde{\nu}}$, ϵ_2 in Algorithm 2 satisfies $\epsilon_2 < \frac{\epsilon_1}{\sqrt{1+\epsilon_1^2}}(\lambda_{\min} - \tilde{\nu})\text{Edist}(X_*, \mathbb{R}^{m\times n}_{\leq(r-1)})$, then the rank of X_n will be increased to at least r for all $n > N$. □

Theorem 22 shows that when started close enough to X_*, a nondegenerate minimizer of f_{F}, the matrices in a sequence generated by Algorithm 2 with $\epsilon_2 > 0$ remain in a neighborhood of X_* and eventually have fixed rank (not necessarily the rank of the X_*).

Theorem 22. *Let f_{F} be a C^2 function and X_* be a nondegenerate minimizer of f_{F} on $\mathcal{M}$ with rank $r_*(r_* \leq k)$, i.e., $\text{grad}f_{\text{F}}(X_*) = 0$ and $\text{Hess}f_{\text{F}}(X_*)$ is positive definite. Furthermore, it is an isolated minimizer of f on $\mathcal{M}_{\leq k}$. Suppose $\epsilon_2 > 0$ and Assumption 2 holds. Denote the sequence of iterates generated by Algorithm 2 by $\{X_n\}$.*

There exists a neighborhood of X_*, $\mathcal{U}_{X_*}$, such that if $\mathcal{D} = \{X \in \mathcal{M} | f(X) \leq f(X_0)\} \subset \mathcal{U}_{X_*}$; $\mathcal{D}$ is compact; $\hat{f} : \text{T}\mathcal{M} \to \mathbb{R} : \xi \mapsto f_{\text{F}} \circ R(\xi)$ is a radially L-C^1 function with sufficient large δ_{RL} defined in Definition 14 such that for any $X, Y \in \mathcal{D}$, $\|R_X^{-1}(Y)\| < \delta_{RL}$, then there exists $N > 0$ such that

$$\forall n_j > N \quad rank(X_{n_j}) = r \quad and \quad X_{n_j} \in \mathcal{U}_{X_*}.$$

Proof. Assume $\{X_{n_j}\} \subset \{X_n\}$ is a subsequence with rank increasing, i.e., $\text{rank}(X_{n_j+1}) > \text{rank}(X_{n_j})$ and $\{X_{n_j}\}$ stay in $\mathcal{U}_{X_*}$.

If $\{X_{n_j}\}$ is a finite sequence, according to Assumption 2, we can obtain the results directly.

If $\{X_{n_j}\}$ is not a finite sequence, following the proofs in Theorem 19: there exists a neighborhood $\mathcal{U}_{X_*}$ of X_* such that for $X_{n_j} \in \mathcal{U}_{X_*}$,

$$\lim_{j\to\infty} \|\text{grad} f_{\text{F}}(X_{n_j})\| = 0.$$

Therefore, there exists a $N > 0$ such that for all X_{n_j}, $n_j > N$,

$$\|\text{grad} f_{\text{F}}(X_{n_j})\| \leq \left(\sqrt{1 + \frac{1}{\epsilon_1^2}}\right) \epsilon_2.$$

Note by Step 6 in Algorithm 2, the ranks of $\{X_{n_j}\}$ are not increasing for $n_j > N$. Combined with Theorem 21 yields the result.

□

So by choosing an approximate solution approach with $\epsilon_2 > 0$, $r \leq r_*$, computational advantages are gained. (Note r_* can be smaller or larger than k, the constraint on rank in the optimization problem.) This is summarized in the the following corollary.

Corollary 23. *(Convergence Rate). If all assumptions in Theorem 20 hold and there exists a K such that all assumptions in Theorem 22 hold for $\{X_n\}$, with $n > K$ then*

- *The sequence $\{X_n\}$ enters a neighborhood $\mathcal{U}_{X_*}$ and remains in that neighborhood so it is known that* $\text{dist}(X_n, X_*)$ *and* $|f(X_*) - f(\tilde{X})|$ *are bounded based ϵ_1, ϵ_2 and* $\text{Hess} f_{\text{F}}(X_*)$.
- $\|P_{\text{T}_{X_n}\mathcal{M}_{\leq k}}(\text{grad} f_{\text{F}}(X_n))\| \leq \delta$ *where δ is based on ϵ_1 and ϵ_2.*
- *The sequence converges on $\mathcal{M}_r$, where $r \leq r_*$, i.e., $X_n \to \tilde{X}$ at the rate of the local Riemannian optimization algorithm.*

3.6 Summary of Algorithmic and Analysis Results

This chapter has defined and analyzed strategies for optimizing a function with a manifold and rank inequality constraints. The main results are as follows.

1. The structure of the tangent cone and a related descent-based method have been described and critiqued from the point of view of convergence rate and practical performance.

2. The differences in the methods by which rank is increased and decreased in descent-based methods has been characterized.

3. Objects have been defined that are used to carefully update the rank, i.e., rank-related vector and rank-related retraction; Rank-related vector and rank-related retraction support increasing the rank appropriately and avoid excessive increase of rank in order to save computations.

4. An algorithmic approach for optimizing a cost function with rank inequality constraint has been developed. Algorithm 2 provides strategy to update the rank carefully. Parameter ϵ_2 is crucial in the sense that it provides a trade-off between the accuracy of solution and efficiency of an algorithm. Setting $\epsilon_2 = 0$ results in an algorithm to find a minimizer to the optimization problem; setting $\epsilon_2 > 0$ results in an algorithm to find an approximate solution, i.e., one that has a small gradient, is near a local minimizer, and has a cost function value that is close to that at the local minimizer, most likely with a lower rank than the nearby local minimizer.

5. A convergence analysis for exact solutions has been completed that shows the following results. ($\{X_n\}$ denotes a sequence of iterates generated by Algorithm 2.)

   - The global convergence analysis shows the infimum limit of $\|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\|$ goes to zero as n goes to ∞;
   - The local convergence analysis shows $\lim_{j\to\infty} \|\mathrm{grad} f_{\mathrm{F}}(X_{n_j})\| = 0$ with $r_* \leq k$ where X_* is assumed to be the unique minimizer in the neighborhood of X_* and $r_* = \mathrm{rank}(X_*)$.
   - The local rank property shows that the ranks of all X_n are eventually greater than or equal to r_*.

6. A convergence analysis for approximate solutions has been completed that shows the following results.

   - The global convergence analysis shows the infimum limit of $\|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\|$ stays small based on given parameters ϵ_1 and ϵ_2. By assumption, the minimizer X_* should be close to a matrix with low-rank. Therefore, it is desired to ignore small singular values and consider approximate solution that has lower rank than X_* but is near X_*. It follows that $\|P_{\mathrm{T}_{X_n}\mathcal{M}_{\leq k}}(\mathrm{grad} f_{\mathrm{F}}(X_n))\|$ is not expected to converge to zero if such an approximate solution to the optimization problem is required. The global convergence analysis shows that Algorithm 2 with $\epsilon_2 > 0$ has the desired property.
   - The local convergence analysis shows that the ranks of $\{X_n\}$ are fixed eventually. Parameter ϵ_2 is used to determine whether the rank is increased. Theorem 21 shows that ϵ_2 can be used to adjust the accuracy of the approximate solution. Theorem 22 proves that the iterates in the sequence eventually have a fixed rank r. Therefore, the convergence

rate is dependent on the local Riemannian algorithm. In addition, if $r < r_*$, then existing local convergence analyses of Riemannian optimization algorithms are applicable, e.g., RTR-Newton, RBFGS, RTR-SR1.

CHAPTER 4

WEIGHTED LOW-RANK APPROXIMATION

In this chapter, the modified Riemannian optimization method is adapted to the weighted low-rank approximation problem and its performance evaluated. In Section 4.1, the specific formulation of the weighted low-rank approximation problem of interest is given. Some relevant existing manifold algorithms are reviewed in Section 4.2. Section 4.2.3 reviews some algorithms for the unweighted and weighted problems that have structure. Section 4.3, the Riemannian geometry of the problem is presented along with the geometric objects required by the proposed algorithm. Finally, the proposed algorithm is empirically evaluated, including comparisons to competing manifold algorithms, in Section 4.4.

4.1 Problem Formulation

The weighted low-rank approximation problem determines a matrix approximation X of a given data matrix R that comes as close as possible with respect to a certain weighted norm:

$$\underset{X \in \mathcal{M}_{\leq k}}{\operatorname{argmin}} \|R - X\|_W^2, \quad f(X) = \|R - X\|_W^2 = \operatorname{vec}\{R - X\}^T W \operatorname{vec}\{R - X\} \tag{4.1}$$

where $R \in \mathbb{R}^{m \times n}$ is given and $W \in \mathbb{R}^{mn \times mn}$ is a positive definite symmetric weighting matrix and $\operatorname{vec}\{A\}$ denotes the vectorized form of A, i.e., a vector constructed by stacking the consecutive columns of A in one vector. The minimizing X in (4.1) is the best rank $k, 0 < k < \min(m, n)$, approximation of R under the norm $\|\cdot\|_W$. Note that for this problem $\mathcal{M} = \mathbb{R}^{m \times n}$ not a submanifold.

Given the constraints, this is a nonconvex optimization problem. When the weighting matrix has significant structure, there may be analytical insight into the form of the minimizer that can be exploited algorithmically. For example, for the more common weighted norm $\|X\|_M = \operatorname{tr}\{X^T M X\}$, where $\operatorname{tr}\{\cdot\}$ is the trace operator and $M \in \mathbb{R}^{m \times n}$ is symmetric positive definite. In this case, the matrix W is in fact a block diagonal matrix with blocks M. This has another common special case when $M = I$. Such problems can often exploit approaches related to truncated factorizations.

Examples of algorithms for such structured problems are briefly reviewed in Section 4.2.3 but are not pursued further in this dissertation.

4.2 Related Work and Historical Context

In this section, two manifold-based optimization approaches for finding locally optimal solutions, starting from a given initial approximation are reviewed.

4.2.1 Alternating Projections Method

Since any m-by-n matrix with rank at most k can be expressed as the product of two matrices of dimension m-by-k and k-by-n, suppose $X = UV^T$, $U \in \mathbb{R}^{m\times k}, V \in \mathbb{R}^{n\times k}$, then (4.1) turns into the following parameter optimization problem

$$\underset{U\in\mathbb{R}^{m\times k},V\in\mathbb{R}^{n\times k}}{\operatorname{argmin}} \|R - UV^T\|_W^2, \quad \|R - UV^T\|_W^2 = \operatorname{vec}\{R - UV^T\}^T W \operatorname{vec}\{R - UV^T\}. \tag{4.2}$$

A well-known possibility for iteratively solving the weighted low-rank approximation problem is the alternating projections procedure [LPW97, WAK97]. It is started from an initial guess of one of the parameters U or V. Fix a value for U and minimize over V, then fix V, minimize over U, repeat until the product UV^T converges. It can be shown that, in general, $X = UV^T$ converges to a local minimum of (4.1) [Kri06] and that the local convergence rate is linear. In practice, however, the method can be rather slow.

4.2.2 Double Minimization Method

Manton et al. present in [MMH03] a novel reformulation of (4.1) and derived a general framework for minimizing a cost function on a Grassmann manifold. They reformulate (4.1) as a double minimization problem

$$\min_{\substack{N\in\mathbb{R}^{n\times(n-k)}\\ N^TN=I}} \quad \min_{\substack{X\in\mathbb{R}^{m\times n}\\ XN=0}} \|R - X\|_W^2. \tag{4.3}$$

They showed that if N and X are the minimizing arguments of the two minimizations in (4.3), then X is the solution of (4.1); the restriction $XN = 0$ enforces the constraint $\operatorname{rank}(X) \le k$ since every column of N must belong to the null space of X. Moreover, the inner minimization, call it $f(N)$,

$$f(N) = \min_{\substack{X\in\mathbb{R}^{m\times n}\\ XN=0}} \|R - X\|_W^2 \tag{4.4}$$

has a closed form solution, given by

$$f(N) = \text{vec}\{R\}^T (N \otimes I_m)[(N \otimes I_m)^T W^{-1} (N \otimes I_m)]^{-1} (N \otimes I_m)^T \text{vec}\{R\}, \tag{4.5}$$

where $\otimes$ is the Kronecker product. The cost function $f(N)$ depends only on the range space (i.e., the column space) of N, rather than on the actual value of N. That is, $f(NO) = f(N)$ for any orthogonal matrix $O \in \mathbb{R}^{(n-k)\times(n-k)}$. The outer minimization can therefore be performed over a smaller space than $\{N : N^T N = I\}$. In fact it is sufficient to minimize $f(N)$ as a function on the Grassmann manifold (the collection of all subspaces of a certain dimension, [EAS98]). A method that combines steepest descent and a Newton-type algorithm was derived in [MMH03] for minimizing $f(N)$. They show that given N that minimizes $f(N)$, the solution to the original problem (4.1) is the unique matrix X satisfying

$$\text{vec}\{X\} = \text{vec}\{R\} - W^{-1}(N \otimes I_m)[(N \otimes I_m)^T W^{-1} (N \otimes I_m)]^{-1} (N \otimes I_m)^T \text{vec}\{R\}. \tag{4.6}$$

Later, Brace and Manton [BM06] used the same transformation of the problem and applied a heuristic version of Riemannian BFGS that was motivated by maintaining low computational complexity through the suppression of vector transport. An interesting relationship with the Riemannian Broyden Family [Hua13] and these two methods is discussed later in Section 4.3.8.

The main problem with the reformulated weighted low-rank approximation is the complexity of the form and computation of the cost function $f(N)$ in (4.5) and recovering the final approximating matrix X in (4.6).

4.2.3 Some Algorithms for Structured W

There are many methods that address matrix approximation problems where the matrix W has significant structure. An important theoretical tool for some of these problems is the truncated factorization. The (generalized) singular value decomposition, when truncated to the leading k terms, provides an optimal approximation in terms of Frobenius norm and the matrix 2-norm. This is referred to the Eckart-Young-Mirsky Theorem [EY36] (see for example the GTLS algorithm of van Huffel and Vandewalle [VV89]). These methods have an advantage when the matrix is not too large and most of the singular values / vectors are needed.

In [ZSJC12], the authors parameterize the approximation matrix $X = LR^T$, with L an $m \times k$ matrix, R an $n \times k$ matrix. An iterative algorithm based on least-squares estimation is proposed

and has been successfully used for exploiting structure, such as Hankel and Toeplitz matrices, and positive semidefiniteness. While this effectively lowers the dimension of the search space, it suffers from linear convergence. Second-order methods like Newton's methods cannot be applied easily. This is essentially the method of Section 4.2.1 applied to a problem where $W = I$.

For $W = I$, a dynamical system low-rank approximation has been proposed by Koch and Lubich [KL07]. This method consists of finding a low-rank matrix on the fixed-rank manifold $\mathcal{M}_k$, for which the authors derive differential equations for the factors that define the rank-k approximation. By numerically integrating this set of differential equations, the rank can be dynamically reduced. However, the large-scale systems involving PDEs are usually expensive to solve and they did not provide a way to increase the rank.

The total least square problem with elementwise weighting (EW-TLS) [PR02a], is the weighted low-rank approximation problem with the weighting matrix W has block diagonal structure

$$W = \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_n \end{bmatrix}$$

where $W_i \in \mathbb{R}^{m \times m}$ and the approximation matrix X satisfies $X \begin{bmatrix} B \\ -I_{n-k} \end{bmatrix} = 0$, where $B \in \mathbb{R}^{k \times (n-k)}$. This weighted low-rank approximation problem can be rewritten as follows:

$$\min_{B \in \mathbb{R}^{k \times (n-k)}} \left(\min_{X \in \mathbb{R}^{m \times n}} \sum_{i=1}^{n} (R_i - X_i)^T W_i^{-1} (R_i - X_i) \right), \tag{4.7}$$

where $R_i, X_i \in \mathbb{R}^m$ are the i-th column of R and X, respectively, W_i is the i-th block in the weighting matrix W. The problem (4.7) can be solved partially by minimizing analytically with respect to X. In this way the following equivalent unconstrained optimization problem is derived

$$B_* = \operatorname*{argmin}_B g(B), \tag{4.8}$$

where

$$g(B) = \sum_{i=1}^{n} R_i^T Z^T (Z W_i Z^T)^{-1} Z R_i, \quad Z = \begin{bmatrix} B^T & -I \end{bmatrix}. \tag{4.9}$$

Given an optimal solution B_*, X can be obtained using the expression:

$$X = R + \Delta X,$$

where

$$\Delta X^T = -\begin{bmatrix} R_1^T Z^T (Z W_1 Z^T)^{-1} Z W_1 \\ \vdots \\ R_n^T Z^T (Z W_n Z^T)^{-1} Z W_n \end{bmatrix}.$$

Premoli and Rastello proposed an iterative algorithm [PR02a] to solve this special case. The algorithm is proven to be locally convergent with a super linear convergence rate [MRP$^+$06]. However, it is not globally convergent and simulation results suggest that the region of convergence to a minimum point can be rather small.

For many applications where the data matrix is large, calculating the SVD can be impractical and other approximate methods must be considered. Baker et al.[BGV12] considered the problem of restricting explicitly the amount of storage available to store an approximate factorization based on a dominant space that is updated as new information becomes available. This can be applied in the generic situation where new columns of the "matrix" correspond to observations of an environment over time that are not stored or where an extremely large matrix is available on very slow, very remote storage and the restricted storage is relatively small and associated with the computational platform used to compute the approximation. The latter case is clearly relevant to the analysis of large-scale data analysis and has the significant advantage of allowing multiple read-only passes through the stored data.

An alternative to Baker et al. for large data is given by randomized or stochastic algorithms that select a subset of the rows and/or columns of the large data matrix, possibly by taking multiple passes through the data. The appropriate decomposition, e.g., eigenvalue or SVD, is used to approximate that of the large data matrix, see the recent survey by Halko et al. [HMT11].

Baker et al., Halko et al. and related approaches are all motivated by very large data and the resulting storage constraint. For this dissertation, it is assumed that, while possibly large, the matrices involved are not large enough to restrict access to read-only.

The modified Riemannian optimization method provides another way to solve low-rank approximation problem. The algorithm has the following potential advantages: first, the problem can be solved without considering reformulation; second, the computation time required is often less than other algorithms, especially when m and n are large, due to the exploitation of state-of-the-art Riemannian optimization algorithms; third, even when the rank constraint k is chosen too large, the algorithm allows an approximate solution of the optimization problem with a reasonable rank.

4.3 Differential Geometry

In this section, the differential geometric objects used in the modified Riemannian optimization method applied to the weighted low-rank approximation problem are considered. Specifically, the tangent and normal spaces, the Riemannian metric, the orthogonal projections, retraction, rank-related retraction and the Riemannian Hessian are characterized. These objects are the building blocks of the modified Riemannian optimization methods described in Chapter 3.

Recall, the Riemannian manifold comprising $m \times n$ matrices is denoted $\mathcal{M} = \mathbb{R}^{m\times n}$, the sub-manifold of matrices with rank r is denoted $\mathcal{M}_r$ and $\mathcal{M}_{\leq k} = \bigcup_{r\leq k} \mathcal{M}_r$ is the set of manifolds defined by the rank inequality constraint. Using the SVD, each fixed-rank manifold $\mathcal{M}_r$ has the equivalent characterization

$$\mathcal{M}_r = \{UDV^T : U \in \mathrm{St}(m,r), V \in \mathrm{St}(n,r), D = \mathrm{diag}(\sigma_1, \ldots, \sigma_r), \sigma_1 \geq \cdots \geq \sigma_r > 0\},$$

where $\mathrm{St}(m,k) = \{X \in \mathbb{R}^{m\times k} | X^T X = I_k\}$ is the compact Stiefel manifold, $I_k \in \mathbb{R}^{k\times k}$ is an identity matrix and $\mathrm{diag}(\sigma_1, \ldots, \sigma_r)$ denotes a diagonal matrix with $\sigma_1, \cdots, \sigma_r$ on the main diagonal.

The representation $X = UDV^T, X \in \mathcal{M}_r$ is not unique. The factorization $X = (UP)(P^T DQ)(VQ)^T$, for any orthogonal matrices $P, Q \in \mathbb{R}^{r\times r}$ where $P^T DQ$ is a real nonnegative diagonal matrix is also an SVD. Therefore, the effect of the choice of U, V and D on the representation of the tangent vector $\dot{X}$ and the determination of the factors U_+, V_+ and D_+ for the next iteration $X_+ = R_X(\dot{X})$ must be considered. The benefit of representing X using the factor U, V and D and updating them directly is avoiding the need to compute an SVD of the iterate X when moving on $\mathcal{M}_r$.

4.3.1 The Tangent Cone

The tangent cone is characterized in Proposition 24 in a manner that is computationally advantageous for the algorithms of interest in this dissertation. The tangent cone has been considered when the embedding space is $\mathbb{R}^{m\times n}$ independently in [SU14] and their characterization is the same as the one used here. Cason et al. in [CAD13] considered the special case of the unit sphere defined by the Frobenius norm in $\mathbb{R}^{m\times n}$.

Proposition 24. *Let $X \in \mathcal{M}_{\leq k}$ with rank $r \leq k$, the tangent cone to $\mathcal{M}_{\leq k}$ at a point X is*

$$\mathrm{T}_X\mathcal{M}_{\leq k} := \left\{ \begin{array}{r} U_r A V_r^T + U_r B V_{r\perp}^T + U_{r\perp} C V_r^T + U_{r\perp} E V_{r\perp}^T : \\ A, B, C, E \quad \textit{are arbitrary matrices,} \ \mathrm{rank}(E) \leq k - r, \end{array} \right\}.$$

$$= \left\{ \begin{array}{r} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & E \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ A, B, C, E \ \textit{are arbitrary matrices,} \ \mathrm{rank}(E) \leq k - r \end{array} \right\}.$$

In general, the tangent cone is not closed under the operator of addition. For example, suppose a matrix $X = U_{k-1}\Sigma V_{k-1}^T$ has rank $k-1$, then the tangent cone to $\mathcal{M}_{\leq k}$ at X has the following structure

$$\begin{bmatrix} U_{k-1} & U_{k-1\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & E \end{bmatrix} \begin{bmatrix} V_{k-1}^T \\ V_{k-1\perp}^T \end{bmatrix}$$

where $A \in \mathbb{R}^{(k-1)\times(k-1)}, B \in \mathbb{R}^{(k-1)\times(n-k+1)}, C \in \mathbb{R}^{(m-k+1)\times(k-1)}, E \in \mathbb{R}^{(m-k+1)\times(n-k+1)}$ and $\mathrm{rank}(E) \leq 1$. Construct the following two matrices,

$$Z_1 = \begin{bmatrix} U_{k-1} & U_{k-1\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & \mathrm{diag}(1, 0, \ldots, 0) \end{bmatrix} \begin{bmatrix} V_{k-1}^T \\ V_{k-1\perp}^T \end{bmatrix},$$

$$Z_2 = \begin{bmatrix} U_{k-1} & U_{k-1\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & \mathrm{diag}(0, \ldots, 0, 1) \end{bmatrix} \begin{bmatrix} V_{k-1}^T \\ V_{k-1\perp}^T \end{bmatrix}.$$

While both Z_1 and Z_2 are in the tangent cone $\mathrm{T}_X\mathcal{M}_{\leq k}$, $Z_1 + Z_2$ is not.

It is not difficult to show that the normal cone to $\mathcal{M}_{\leq k}$ at a point X is

$$\mathrm{N}_X\mathcal{M}_{\leq k} := \left\{ \begin{array}{r} U_{r\perp} E_\perp V_{r\perp}^T : \\ E_\perp = 0 \text{ if } r < k, E_\perp \text{ arbitrary if } r = k. \end{array} \right\}.$$

$$= \left\{ \begin{array}{r} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & E_\perp \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ E_\perp = 0 \text{ if } r < k, E_\perp \text{ arbitrary if } r = k. \end{array} \right\}.$$

4.3.2 Gradients of Interest

$\mathcal{M}_r$ is a smooth fixed-rank manifold and the tangent space is $\forall\ X = U_r D_r V_r^T \in \mathcal{M}_r$ [Van13],

$$\mathrm{T}_X\mathcal{M}_r := \left\{ \begin{array}{r} U_r A V_r^T + U_r B V_{r\perp}^T + U_{r\perp} C V_r^T : \\ A, B, C \text{ are arbitrary matrices,} \end{array} \right\},$$

$$= \left\{ \begin{array}{r} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ A, B, C \text{ are arbitrary matrices} \end{array} \right\}.$$

and the normal space to $\mathcal{M}_r$ at the point $X = U_r D_r V_r^T \in \mathcal{M}_r$ is

$$\begin{aligned}\mathrm{N}_X\mathcal{M}_r :&= \left\{ \begin{array}{r} U_{r\perp} E V_{r\perp}^T : \\ E \text{ is an arbitrary matrix,} \end{array} \right\}, \\ &= \left\{ \begin{array}{l} \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & E \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix} : \\ \quad E \text{ is an arbitrary matrix} \end{array} \right\}.\end{aligned}$$

where $U_{r\perp}$ and $V_{r\perp}$ are any orthogonal complements of U_r and V_r respectively.

$\mathcal{M}_r$ becomes a Riemannian manifold with the choice of a Riemannian metric, in this case, the metric is inherited from $\mathcal{M}$ and is

$$g_X(\xi, \eta) := \langle \xi, \eta \rangle_F = \mathrm{vec}\{\xi\}^T \mathrm{vec}\{\eta\} \quad \textit{with } X \in \mathcal{M}_r \textit{ and } \xi, \eta \in \mathrm{T}_X\mathcal{M}_r.$$

The resulting Riemannian gradient is the orthogonal projection onto the tangent space of the gradient of f seen as a function on $\mathbb{R}^{m\times n}$.

The orthogonal projection onto the tangent space at $X = U_r D_r V_r^T \in \mathcal{M}_r$ is

$$\begin{aligned} P_X &: \mathbb{R}^{m\times n} \to \mathrm{T}_X\mathcal{M}_r \\ Z \to P_X Z &= U_r U_r^T Z V_r V_r^T + U_r U_r^T Z V_{r\perp} V_{r\perp}^T + U_{r\perp} U_{r\perp}^T Z V_r V_r^T \\ &= U_r U_r^T Z + Z V_r V_r^T - U_r U_r^T Z V_r V_r^T \end{aligned} \tag{4.10}$$

and the orthogonal projection onto the normal space at $X = U_r D_r V_r^T \in \mathcal{M}_r$ is

$$\begin{aligned} P_X^\perp &: \mathbb{R}^{m\times n} \to \mathrm{N}_X\mathcal{M}_r \\ Z \to P_X^\perp Z &= (I_m - U_r U_r^T) Z (I_n - V_r V_r^T). \end{aligned} \tag{4.11}$$

Note the simplification in notation in that the subscript X indicates the element of the manifold that defines the tangent space.

Consider the following two function f_{F} and f_r:

$$f_{\mathrm{F}} : \mathcal{M} \to \mathbb{R} : X \mapsto \|R - X\|_W^2,$$

$$f_r : \mathcal{M}_r \to \mathbb{R} : X \mapsto \|R - X\|_W^2.$$

The cost function for the rank inequality constrained problem is then $f = f_F|_{\mathcal{M}_{\leq k}}$ and $f_r = f_{\mathrm{F}}|_{\mathcal{M}_r} = f|_{\mathcal{M}_r}$.

Since the Euclidean gradient of the cost function f_{F} is $\bigtriangledown f_{\mathrm{F}} = -2\mathrm{vec}^{-1}(W\mathrm{vec}\{R - X\})$, the projection of the gradient onto tangent space, i.e., the Riemannian gradient on fixed-rank manifold $\mathcal{M}_r$, is

$$\mathrm{grad} f_r := P_X(- \bigtriangledown f_{\mathrm{F}}).$$

Note that $W\mathrm{vec}\{R - X\}$ is a vector, and it must be reshaped as a matrix. The expression $\mathrm{vec}^{-1}(W\mathrm{vec}\{R - X\})$ is used to represent this matrix formation, where the mapping $\mathrm{vec}(A) \mapsto A$ is dentoed by vec^{-1}.

4.3.3 Retraction onto a Fixed-rank Manifold

Two kinds of retraction are required for the proposed Riemannian optimization algorithm: retraction onto the fixed-rank manifolds and a rank-related retraction. In this section, retractions on the fixed-rank manifold $\mathcal{M}_r$ and their relationship to the representation using the three factors (U, D, V) for $X \in \mathcal{M}_r$ are discussed.

Given a specific triple (U, D, V) for an $X \in \mathcal{M}_r$, according to [KL07, Proposition 2.1], there exists a unique representation $(\dot{U}, \dot{D}, \dot{V})$ of any $\dot{X} \in \mathrm{T}_X\mathcal{M}_r$ that can be computed efficiently and satisfies

$$\dot{X} = \dot{U}DV^T + U\dot{D}V^T + UD\dot{V}^T, \tag{4.12}$$

$$U^T\dot{U} = 0, V^T\dot{V} = 0. \tag{4.13}$$

A method for computing $(\dot{U}, \dot{D}, \dot{V})$ is discussed in Section 4.3.4.

Once the matrices $\dot{U}, \dot{D}, \dot{V}$ are known, a retraction can be applied to determine X_+. There are several retractions related to the compact Stiefel manifold and projections that can be considered as a building block for a retraction on $\mathcal{M}_r$ based on the three-factor representation of X [AO13]. Three retractions on $\mathcal{M}_r$ are considered: the **three-factor SVD-type retraction**, the **three-factor QR-type retraction** and the **three-factor polar-type retraction**.

The **three-factor SVD-type retraction** is a projective retraction defined as

$$R_X(\dot{X}) = \underset{Y \in \mathcal{M}_r}{\mathrm{argmin}} \|Y - (X + \dot{X})\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm [AM12]. Let $\sigma_1(A), \cdots, \sigma_{\min(m,n)}(A)$ denote the singular values of an $m \times n$ matrix A in decreasing order. By Proposition 6 of [AM12] whenever $\dot{X}$ is

sufficiently small for $\|\dot{X}\| < \sigma_r(X)/2$ to hold, $R_X(\dot{X})$ exists, is unique, and

$$R_X(\dot{X}) = \sum_{i=1}^{r} \sigma_i u_i v_i, \tag{4.14}$$

where $X+\dot{X} = \begin{bmatrix} u_1 & \cdots & u_{\min(m,n)} \end{bmatrix} \operatorname{diag}(\sigma_1, \cdots, \sigma_{\min(m,n)}) \begin{bmatrix} v_1 & \cdots & v_{\min(m,n)} \end{bmatrix}^T$ is the SVD. Vandereycken [Van13, Algorithm 6] shows that the SVD-type retraction can be computed efficiently as follows:

$$R_X(\dot{X}) = U_+ D_+ V_+^T \tag{4.15}$$

where

$$\begin{aligned}
&\dot{U}D = Q_u R_u, \\
&\dot{V}D = Q_v R_v, \\
&U_s D_s V_s = \begin{bmatrix} D+\dot{D} & R_v^T \\ R_u & 0 \end{bmatrix}, \\
&U_+ = \begin{bmatrix} U & Q_u \end{bmatrix} U_s(:, 1:r), \\
&D_+ = D_s(1:r, 1:r), \\
&V_+ = \begin{bmatrix} V & Q_v \end{bmatrix} V_s(:, 1:r),
\end{aligned}$$

The algorithm requires the computation of two QR factorizations (one of an $m \times r$ matrix and one of an $n \times r$ matrix) the SVD of a $2r$-by-$2r$ matrix and $4mr^2 + 4nr^2$ operations in two matrix multiplications yielding $O((m+n)r^2) + O(r^3)$ operations where the coefficient of the first term depends on the method used to determine the QR factors.

Since the decomposition of $X = UDV^T$ is not unique, ideally the result of retraction, i.e., $U_+ D_+ V_+^T = R_X(\dot{X})$, should not depend on the particular (U, D, V) triple used. The three-factor SVD-type retraction is clearly invariant with respect to the choice of factors.

If the assumptions on D in the triple (U, D, V) are relaxed to require only nonsingularity and $\dot{X}$ is specified by the associated unique triple (U, D, V) then a retraction that is independent of the choice of (U, D, V) can be defined in terms of the polar decomposition [AO13, MS13]. The retraction is defined as

$$R_X(\dot{X}) = U_+ D_+ V_+^T, \quad U_+ = uf(U+\dot{U}), \quad D_+ = D+\dot{D}, \quad V_+ = uf(V+\dot{V})$$

where $uf\{\cdot\}$ denotes the orthonormal factor of the polar decomposition. This retraction requires the computation of two QR factorizations (one of an $m \times r$ matrix and one of an $n \times r$ matrix)

and SVD's of two $r \times r$ matrices for $O((m+n)r^2) + O(r^3)$ operations with coefficients smaller than the three-factor SVD-type retraction. So there is some benefit with respect to computational cost, however, this is only true on steps where estimates of the singular values of D_+ are not needed for rank adjustment, e.g., while the iteration remains on $\mathcal{M}_r$.

If the matrices D and D_+ are required to be nonnegative diagonal matrices the retraction must be modified. If this can be done with an acceptable increase in computational complexity then rank adjustment using Algorithm 1 can be done more often.

The **three-factor Polar-decomposition-type retraction** that imposes the diagonal requirement is defined

$$R_X(\dot{X}) = U_+ D_+ V_+^T \tag{4.16}$$

where

$$\begin{aligned} &U_S D_S V_S^T = D + \dot{D} \quad \text{using SVD,} \\ &U_+ = uf(U + \dot{U})U_S, \\ &D_+ = D_S, \\ &V_+ = uf(V + \dot{V})V_S \end{aligned}$$

and the symbol $uf(\cdot)$ denotes the orthogonal component of the polar decomposition. This retraction requires the computation of two QR factorizations (one of an $m \times r$ matrix and one of an $n \times r$ matrix) and SVD's of three $r \times r$ matrices, and $2mr^2 + 2nr^2$ operations in two matrix multiplications yielding $O((m+n)r^2) + O(r^3)$ operations.

Three-factor retractions that require fewer computations but that are not guaranteed to be invariant to the choice of (U, D, V) are also possible. Empirical evidence presented later in this dissertation demonstrates that this invariance is not necessary to achieve superlinear convergence but sufficient conditions on a three-factor retraction for superlinear convergence are not yet known and is the subject of future research.

The **three-factor QR-type retraction I** is defined as

$$R_X(\dot{X}) = U_+ D_+ V_+^T. \tag{4.17}$$

where

$$\begin{aligned} U_S D_S V_S^T &= D + \dot{D} \quad \text{using SVD}, \\ U_+ &= qf(U + \dot{U})U_S, \\ D_+ &= D_S, \\ V_+ &= qf(V + \dot{V})V_S, \end{aligned}$$

and the symbol $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument. This saves SVD's of two $r \times r$ matrices. Note that D and D_+ are diagonal matrices.

A second three-factor QR-type retraction can be defined that avoids the SVD. The **QR-type retraction II** is defined as

$$R_X(\dot{X}) = U_+ D_+ V_+^T. \tag{4.18}$$

where

$$\begin{aligned} U_+ &= qf(U + \dot{U}), \\ D_+ &= D + \dot{D}, \\ V_+ &= qf(V + \dot{V}), \end{aligned}$$

and the symbol $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument. Note that this defines a retraction even if D and D_+ are not diagonal matrices. If they are constrained to be diagonal then $\dot{D}$ must then also be a diagonal matrix. This is discussed in Section 4.3.4.

4.3.4 Computing $(\dot{U}, \dot{D}, \dot{V})$

Given the triple (U, D, V), the triple $(\dot{U}, \dot{D}, \dot{V})$ that satisfies

$$\dot{X} = \dot{U} D V^T + U \dot{D} V^T + U D \dot{V}^T,$$

can be computed efficiently [KL07, Proposition 2.1] and, if necessary, can be constrained so that $\dot{D}$ is a diagonal matrix.

Since $U \in \mathrm{St}(m, r), V \in \mathrm{St}(n, r)$ and the form of the tangent space at a point S is

$$\mathrm{T_S St}(n, r) = \{S\Omega + S_\perp K : \Omega^T = -\Omega, K \in \mathbb{R}^{(n-r)\times r}\},$$

it follows that

$$\begin{aligned} \dot{X} &= (U\Omega_U + U_\perp K_U) D V^T + U \dot{D} V^T + U D (V \Omega_V + V_\perp K_V)^T \\ &= U(\Omega_U D + \dot{D} + D\Omega_V^T) V^T + U_\perp K_U D V^T + U D K_V^T V_\perp^T, \end{aligned} \tag{4.19}$$

where $\Omega_U^T = -\Omega_U, \Omega_V^T = -\Omega_V$, $K_U \in \mathbb{R}^{(m-r)\times r}, K_V \in \mathbb{R}^{(n-r)\times r}$ are arbitrary matrices, and the subscript $\perp$ indicates a matrix with the maximum number of columns orthogonal to the matrix argument.

In order to get explicit forms of $\dot{U}$ and $\dot{V}$, the matrices Ω_U, Ω_V, K_U, and K_V are needed. By equation (4.19), the following hold

$$\Omega_U D + \dot{D} + D\Omega_V^T = U^T \dot{X} V, \tag{4.20}$$

$$K_U = U_\perp^T \dot{X} V D^{-1}, \tag{4.21}$$

$$K_V^T = D^{-1} U^T \dot{X} V_\perp. \tag{4.22}$$

If $\dot{U} = U_\perp K_U$ and $\dot{V} = V_\perp K_V$ then $\dot{D}$ is unique but not necessarily diagonal [KL07]. This is easily derived from (4.21) and (4.22) it follows that

$$\dot{U} = U_\perp K_U = U_\perp U_\perp^T \dot{X} V D^{-1} = (I - UU^T)\dot{X} V D^{-1}$$
$$\dot{V} = V_\perp K_V = V_\perp V_\perp^T \dot{X}^T U D^{-1} = (I - VV^T)\dot{X}^T U D^{-1}$$

and from (4.20)

$$\dot{D} = U^T \dot{X} V.$$

So given the $m \times n$ matrix $\dot{X}$ the triple $(\dot{U}, \dot{D}, \dot{V})$ is uniquely defined [KL07].

If D is assumed diagonal then D_+ must be diagonal if $\dot{D}$ is required to be diagonal. This can be achieved using nonzero Ω_U and Ω_V. Specifically, if D and $\dot{D}$ are diagonal then $\dot{U}$ and $\dot{V}$ are unique and easily computed. In the following, we seek the expression of $(\dot{U}, \dot{D}, \dot{V})$ with $\dot{D}$ is a diagonal matrix.

Since Ω_U and Ω_V are skew matrices their main diagonals are 0 and from (4.20) a diagonal $\dot{D}$ follows

$$\dot{D} = \operatorname{diag}(U^T \dot{X} V), \tag{4.23}$$

and therefore

$$Z := \Omega_U D + D\Omega_V^T = U^T \dot{X} V - \dot{D}. \tag{4.24}$$

Multiplying (4.24) on the right by D^{-1}, yields

$$\Omega_U + D\Omega_V^T D^{-1} = Z D^{-1}. \tag{4.25}$$

Adding (4.25) to its transpose, since $\Omega_U + \Omega_U^T = 0$ and $\Omega_V^T = -\Omega_V$, yields

$$-D\Omega_V D^{-1} + D^{-1}\Omega_V D = D^{-1}Z^T + ZD^{-1}.$$

Since D and D^{-1} are diagonal matrices, assuming $\mathrm{diag}(D) = \{a_i\}, \mathrm{diag}(D^{-1}) = \{b_j\}$, the element in the i, j position of matrix $D\Omega_V D^{-1}$ is $[D\Omega_V D^{-1}]_{ij} = a_i b_j (\Omega_V)_{ij}$. Using v_D to represent the vector comprising the diagonal elements of D, $v_{D^{-1}}$ to represent the vector comprising the diagonal elements of D^{-1}, then $D\Omega_V D^{-1} = (v_D v_{D^{-1}}^T) \circ \Omega_V$ and $D^{-1}\Omega_V D = (v_{D^{-1}} v_D^T) \circ \Omega_V$, where $\circ$ denotes the Hadamard product of two matrices. It follows that

$$(v_{D^{-1}} v_D^T - v_D v_{D^{-1}}^T) \circ \Omega_V = D^{-1}Z^T + ZD^{-1}. \tag{4.26}$$

and by similar analysis for Ω_U

$$(v_{D^{-1}} v_D^T - v_D v_{D^{-1}}^T) \circ \Omega_U = D^{-1}Z + Z^T D^{-1}. \tag{4.27}$$

Explicit expressions for Ω_U and Ω_V follow from these equations. Therefore, given $\dot{X}$, solving (4.27) and (4.26) gives Ω_U and Ω_V respectively and $\dot{U}, \dot{D}, \dot{V}$, can be computed from (4.21), (4.22), (4.23).

These approaches to computing the triple $(\dot{U}, \dot{D}, \dot{V})$ assume $\dot{X}$ is computed explicitly. It is an open question if the triple can be computed without forming $\dot{X}$. This of course depends on the definitions used by the various Riemannian optimization algorithms on $\mathcal{M}_r$ for the direction vector $\dot{X}$.

4.3.5 Rank-related Retraction

In order to change the rank, a rank-related retraction that satisfies the properties in Definition 8 of Chapter 3 is required. This is discussed in this section by first, constructing a rank-$\tilde{r}$-related vector and then using it to generalize the fixed-rank retractions.

Consider the manifold $\mathcal{M} = \mathbb{R}^{m\times n}$, the full gradient $\mathrm{grad} f_{\mathrm{F}}(x^*)$ of a point $x^* = U_r D_r V_r^T$ on $\mathcal{M}$ can written as

$$\mathrm{grad} f_{\mathrm{F}}(x^*) = \begin{bmatrix} U_r & U_{r\perp} \end{bmatrix} \begin{bmatrix} A & B \\ C & E \end{bmatrix} \begin{bmatrix} V_r^T \\ V_{r\perp}^T \end{bmatrix}.$$

From the structure, the increased rank depends on the term E. Given $\tilde{r}$, according to Definition 11 in Chapter 3, the search direction is chosen to be

$$\eta^* = \underset{\eta \in \mathrm{T}\mathcal{M}_{\leq \tilde{r}}}{\mathrm{argmin}} \, \|\mathrm{grad} f_{\mathrm{F}}(x^*) - \eta\|_2. \tag{4.28}$$

Note that η^* is not unique.

An example of a choice that satisfies (4.28) can be defined by taking the SVD of E, finding the largest Δr terms, and writing η^* as

$$\begin{aligned}\eta^* &= \begin{bmatrix}U_r & U_{\Delta r} & U_{(r+\Delta r)\perp}\end{bmatrix}\begin{bmatrix}A & B_1 & B_2\\ C_1 & E_{\Delta r} & 0\\ C_2 & 0 & 0\end{bmatrix}\begin{bmatrix}V_r^T\\ V_{\Delta r}^T\\ V_{(r+\Delta r)\perp}^T\end{bmatrix}\\ &= \begin{bmatrix}U_{\tilde{r}} & U_{\tilde{r}\perp}\end{bmatrix}\begin{bmatrix}A' & B_2\\ C_2 & 0\end{bmatrix}\begin{bmatrix}V_{\tilde{r}}^T\\ V_{\tilde{r}\perp}^T\end{bmatrix}\end{aligned} \tag{4.29}$$

where $\tilde{r} = r + \Delta r \leq k$. Based on this construction, it is clear that $\mathrm{grad} f_F(x^*)$ and η^* are not approximately orthogonal to each other. Furthermore, since η^* have the structure shown in (4.29), for $X = U_r D_r V_r^T$, it can be written

$$\begin{aligned}X &= \begin{bmatrix}U_r & U_{\Delta r} & U_{(r+\Delta r)\perp}\end{bmatrix}\begin{bmatrix}D_r & 0^{r\times\Delta r} & 0^{r\times(n-\tilde{r})}\\ 0^{\Delta r\times r} & 0^{\Delta r\times\Delta r} & 0^{\Delta r\times(n-\tilde{r})}\\ 0^{(m-\tilde{r})\times r} & 0^{(m-\tilde{r})\times\Delta r} & 0^{(m-\tilde{r})\times(n-\tilde{r})}\end{bmatrix}\begin{bmatrix}V_r^T\\ V_{\Delta r}^T\\ V_{(r+\Delta r)\perp}^T\end{bmatrix}\\ &= \begin{bmatrix}U_{\tilde{r}} & U_{\tilde{r}\perp}\end{bmatrix}\begin{bmatrix}D_{\tilde{r}} & 0^{\tilde{r}\times(n-\tilde{r})}\\ 0^{(m-\tilde{r})\times\tilde{r}} & 0^{(m-\tilde{r})\times(n-\tilde{r})}\end{bmatrix}\begin{bmatrix}V_{\tilde{r}}^T\\ V_{\tilde{r}\perp}^T\end{bmatrix}.\end{aligned}$$

Therefore, given $X \in \mathcal{M}_r$ and η^*, the rank-related retractions versions of the three-factor retractions discussed earlier can be constructed.

Since η^* can be written

$$\eta^* = \dot{U}_{\tilde{r}} D_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} \dot{D}_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} D_{\tilde{r}} \dot{V}_{\tilde{r}}, \tag{4.30}$$

$$U_{\tilde{r}}^T \dot{U}_{\tilde{r}} = 0, V_{\tilde{r}}^T \dot{V}_{\tilde{r}} = 0. \tag{4.31}$$

and the decomposition (4.30) is given by

$$\begin{aligned}\dot{U}_{\tilde{r}} &= U_{\tilde{r}\perp} U_{\tilde{r}\perp}^T \eta^* V_{\tilde{r}} D_{\tilde{r}}^{-1} = (I_m - U_{\tilde{r}} U_{\tilde{r}}^T)\eta^* V_{\tilde{r}} D_{\tilde{r}}^{-1},\\ \dot{D}_{\tilde{r}} &= U_{\tilde{r}}^T \eta^* V_{\tilde{r}},\\ \dot{V}_{\tilde{r}} &= V_{\tilde{r}\perp} V_{\tilde{r}\perp}^T \eta^{*T} U_{\tilde{r}} D_{\tilde{r}}^{-T} = (I_n - V_{\tilde{r}} V_{\tilde{r}}^T)\eta^{*T} U_{\tilde{r}} D_{\tilde{r}}^{-T},\end{aligned} \tag{4.32}$$

the **Polar-decomposition-type rank-related retraction** is defined

$$\tilde{R}_X(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+, \tag{4.33}$$

where

$$\tilde{U}_+ = uf(U_{\tilde{r}} + \dot{U}_{\tilde{r}})U_S,$$
$$\tilde{D}_+ = D_S,$$
$$\tilde{V}_+ = uf(V_{\tilde{r}} + \dot{V}_{\tilde{r}})V_S,$$
$$D_{\tilde{r}} + \dot{D}_{\tilde{r}} = U_S D_S V_S^T,$$

where the symbol $uf(\cdot)$ denotes the orthogonal component of the polar decomposition.

The **SVD-type rank-related retraction** is defined as

$$\tilde{R}_X(\eta^*) = \sum_{i=1}^{\tilde{r}} \sigma_i u_i v_i, \tag{4.34}$$

where $X + \eta^* = \begin{bmatrix} u_1 & \cdots & u_{\min(m,n)} \end{bmatrix} \operatorname{diag}(\sigma_1, \cdots, \sigma_{\min(m,n)}) \begin{bmatrix} v_1 & \cdots & v_{\min(m,n)} \end{bmatrix}^T$ is the SVD with singular values in decreasing order.

Defining $\tilde{U}_p = \tilde{U}_\perp \tilde{D}$ and $\tilde{V}_p = \tilde{V}_\perp \tilde{D}$ allows η^* to be rewritten in the form

$$\eta^* = \tilde{U}_p V_{\tilde{r}}^T + U_{\tilde{r}} \dot{D}_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} \tilde{V}_p, \tag{4.35}$$

$$U_{\tilde{r}}^T \tilde{U}_p = 0, V_{\tilde{r}}^T \tilde{V}_p = 0 \tag{4.36}$$

and the decomposition (4.35) rewritten as

$$\begin{aligned} \tilde{U}_p &= U_{\tilde{r}\perp} U_{\tilde{r}\perp}^T \eta^* V_{\tilde{r}} = (I_m - U_{\tilde{r}} U_{\tilde{r}}^T)\eta^* V_{\tilde{r}}, \\ \dot{D}_{\tilde{r}} &= U_{\tilde{r}}^T \eta^* V_{\tilde{r}}, \\ \tilde{V}_p &= V_{\tilde{r}\perp} V_{\tilde{r}\perp}^T \eta^{*T} U_{\tilde{r}} = (I_n - V_{\tilde{r}} V_{\tilde{r}}^T)\eta^{*T} U_{\tilde{r}}. \end{aligned} \tag{4.37}$$

Therefore, the retraction (4.34) can be efficiently computed by

$$\tilde{R}_X(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+, \tag{4.38}$$

where

$$\begin{aligned} Q_u R_u &= \tilde{U}_p, \\ Q_v R_v &= \tilde{V}_p, \\ U_s D_s V_s &= \begin{bmatrix} D_{\tilde{r}} + \dot{D}_{\tilde{r}} & R_v^T \\ R_u & 0 \end{bmatrix}, \\ \tilde{U}_+ &= \begin{bmatrix} U_{\tilde{r}} & Q_u \end{bmatrix} U_s(:, 1:\tilde{r}), \\ \tilde{D}_+ &= D_s(1:\tilde{r}, 1:\tilde{r}), \\ \tilde{V}_+ &= \begin{bmatrix} V_{\tilde{r}} & Q_v \end{bmatrix} V_s(:, 1:\tilde{r}), \end{aligned}$$

and $\tilde{r} = r + \Delta r$, $U_{\tilde{r}} = \begin{bmatrix} U_r & U_{\Delta r} \end{bmatrix}$, $D_{\tilde{r}} = \begin{bmatrix} D_r & 0^{r\times\Delta r} \\ 0^{\Delta r\times r} & 0^{\Delta r\times\Delta r} \end{bmatrix}$, $V_{\tilde{r}} = \begin{bmatrix} V_r & V_{\Delta r} \end{bmatrix}$.

Similarly, given η^* in the form of (4.30), the **QR-type rank-related retraction I** is defined as

$$\tilde{R}_X(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+, \tag{4.39}$$

where

$$\begin{aligned} \tilde{U}_+ &= qf(U_{\tilde{r}} + \dot{U}_{\tilde{r}})U_S, \\ \tilde{D}_+ &= D_S, \\ \tilde{V}_+ &= qf(V_{\tilde{r}} + \dot{V}_{\tilde{r}})V_S, \\ D_{\tilde{r}} + \dot{D}_{\tilde{r}} &= U_S D_S V_S^T, \end{aligned}$$

where the symbol $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument.

If η^* is given in the form of

$$\eta^* = \dot{U}_{\tilde{r}} D_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} \dot{D}_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} D_{\tilde{r}} \dot{V}_{\tilde{r}}, \tag{4.40}$$

where $\dot{D}_{\tilde{r}}$ is a diagonal matrix, then we have the the **QR-type rank-related retraction II**:

$$\tilde{R}_X(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+, \tag{4.41}$$

where

$$\begin{aligned} \tilde{U}_+ &= qf(U_{\tilde{r}} + \dot{U}_{\tilde{r}}), \\ \tilde{D}_+ &= D_{\tilde{r}} + \dot{D}_{\tilde{r}}, \\ \tilde{V}_+ &= qf(V_{\tilde{r}} + \dot{V}_{\tilde{r}}), \end{aligned}$$

where the symbol $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument. The computation of $\dot{U}, \dot{D}, \dot{V}$ in (4.40) is the same as discussed in Section 4.3.3.

4.3.6 Vector Transport on Fixed-rank Manifold

Vector transport is critical to the success of Riemannian optimization algorithms such as Riemannian quasi-Newton methods. It is used to compare tangent vectors in different tangent spaces and to transport operators on one tangent space to another tangent space. Vector transport can be represented by a matrix. Given two points X_1 and X_2 on a fixed-rank manifold $\mathcal{M}_r$, the corresponding tangent spaces are $\mathrm{T}_{X_1}, \mathrm{T}_{X_2}$. Huang in [Hua13] proposes some methods to construct

isometric vector transports $\mathcal{T}$ from X_1 to X_2 as the direct rotation [DK70] from $\mathrm{T}_{X_1}\mathcal{M}_r$ to $\mathrm{T}_{X_2}\mathcal{M}_r$, restricted to act on $\mathrm{T}_{X_1}\mathcal{M}_r$. This section presents the application of those techniques to $\mathcal{M}_r$.

Note that the tangent space on the fixed-rank manifold has the following structure,

$$\mathrm{T}_X\mathcal{M}_r := \left\{ \begin{array}{c} U_r A V_r^T + U_r B V_{r\perp}^T + U_{r\perp} C V_r^T : \\ A \in \mathbb{R}^{r\times r}, B \in \mathbb{R}^{r\times(n-r)}, C \in \mathbb{R}^{(m-r)\times r}. \end{array} \right\}.$$

It is a $(m+n-r)r$-dimensional subspace of $\mathbb{R}^{mn}$. An orthonormal basis of $\mathrm{T}_X\mathcal{M}_r$ denoted by B_X is given by

$$\begin{aligned} &\{U(e_i e_j^T)V : i = 1, \cdots, r, j = 1, \cdots, r\} \\ &\cup \{U(e_j \tilde{e}_i^T)V_\perp^T : i = 1, \cdots, n-r, j = 1, \cdots, r\} \\ &\cup \{U_\perp(\hat{e}_i e_j^T)V^T : i = 1 \cdots, m-r, j = 1, \cdots, r\} \end{aligned}$$

where $(e_1, \cdots, e_r)$ is the canonical basis of $\mathbb{R}^r$, $(\hat{e}_1, \cdots, \hat{e}_{m-r})$ is the canonical basis of $\mathbb{R}^{m-r}$ and $(\tilde{e}_1, \cdots, \tilde{e}_{n-r})$ is the canonical basis of $\mathbb{R}^{n-r}$. The columns of B_X are thus chosen as the "vec" of the basis elements.

Let B_{X_1} and B_{X_2} be orthonormal bases of $\mathrm{T}_{X_1}\mathcal{M}_r$ and $\mathrm{T}_{X_2}\mathcal{M}_r$. The direct-rotation vector transport from X_1 to X_2 is then given by

$$\mathcal{T} = B_{X_2} U_b^T B_{X_1}^T, \tag{4.42}$$

where $B_{X_1}^T B_{X_2} = U_b P_b$ is the unique polar decomposition. The operator defined by (4.42) is called a vector transport by direct-rotation based on tangent space.

If the codimension, $(m-r)(n-r)$, is sufficiently smaller than the dimension, $(m+n-r)r$, an orthonormal basis for normal space $\mathrm{N}_X\mathcal{M}_r$ can be efficiently constructed. The normal space on the fixed-rank manifold $\mathcal{M}_r$ has the following structure,

$$\mathrm{N}_X\mathcal{M}_r = \left\{ U_{r\perp} E_\perp V_{r\perp}^T : E_\perp \in \mathbb{R}^{(m-r)\times(n-r)} \right\}.$$

Using N_X to denote the orthonormal basis of $\mathrm{N}_X\mathcal{M}_r$, it is given by

$$\{U_\perp \tilde{e}_i \hat{e}_j V_\perp^T : i = 1, \cdots, m-r, j = 1, \cdots, n-r\}.$$

The columns of N_X are thus chosen as the "vec" of the basis elements.

Let N_{X_1}, N_{X_2} be orthonormal basis of $\mathrm{N}_{X_1}\mathcal{M}_r$ and $\mathrm{N}_{X_2}\mathcal{M}_r$. The direct-rotation vector transport from X_1 to X_2 is then given by

$$\mathcal{T} = (I_{mn} - Q_{X_1}Q_{X_1}^T) + Q_{X_2}U_q^T Q_{X_1}^T, \tag{4.43}$$

where $Q_{X_1}^T Q_{X_2} = U_q P_q$ is the unique polar decomposition, Q_{X_1}, Q_{X_2} are orthonormal basis of $\mathrm{T}_{X_1}\mathcal{M}_r \ominus (\mathrm{T}_{X_1}\mathcal{M}_r \cap \mathrm{T}_{X_2}\mathcal{M}_r)$ and $\mathrm{T}_{X_2}\mathcal{M}_r \ominus (\mathrm{T}_{X_2}\mathcal{M}_r \cap \mathrm{T}_{X_2}\mathcal{M}_r)$, which can be obtained by orthonormalizing $(I - N_{X_1}N_{X_1}^T)N_{X_2}$ and $(I - N_{X_2}N_{X_2}^T)N_{X_1}$ respectively. The operator defined by (4.43) is called a vector transport by direct-rotation based on normal space.

For weighted low-rank approximation, it is often assumed $k << \min(m, n)$ and vector transport by direct-rotation based on normal space is not computationally reasonable so direct-rotation by tangent space is the preferred form in the remainder of the discussion.

Vector transport by the differentiated retraction of (4.16) and (4.18) can also be derived. Similar to the idea in [MS13, Section 3.4], the following Proposition states the differentiated retraction of (4.16).

Proposition 25. *Let $X = UDV^T \in \mathcal{M}_r$, $\xi, \eta \in \mathrm{T}_X\mathcal{M}_r$. Assuming ξ and η have the following structure*

$$\xi = \dot{U}_1 D V^T + U\dot{D}_1 V^T + U D \dot{V}_1^T,$$
$$\eta = \dot{U}_2 D V^T + U\dot{D}_2 V^T + U D \dot{V}_2^T$$

then the vector transport by the differentiated retraction of (4.16) *is*

$$\mathcal{T}_\eta \xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1)(D+\dot{D}_2)(uf(V+\dot{V}_2))^T + uf(U+\dot{U}_2)\dot{D}_1 uf(V+\dot{V}_2)^T + uf(U+\dot{U}_2)(D+\dot{D}_2)(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T, \tag{4.44}$$

where $\mathcal{T}_{\dot{U}_2}(\dot{U}_1)$ is a vector transport by differentiated retraction of (4.16) *on the Stiefel manifold [Hua13, Lemma 10.2.1] and $uf(\cdot)$ denotes the orthogonal factor of the polar decomposition.*

Proof. Based on the definition of the vector transport by differentiated retraction and the polar-decomposition-type retraction (4.16), the following hold

$$\begin{aligned}
\mathcal{T}_{\eta}\xi &= \frac{d}{dt}R_X(\eta + t\xi)|_{t=0} \\
&= \frac{d}{dt}[uf(U + \dot{U}_2 + t\dot{U}_1)(D + \dot{D}_2 + t\dot{D}_1)uf(V + \dot{V}_2 + t\dot{V}_1)^T]|_{t=0} \\
&= \frac{d}{dt}[uf(U + \dot{U}_2 + t\dot{U}_1)](D + \dot{D}_2 + t\dot{D}_1)uf(V + \dot{V}_2 + t\dot{V}_1)^T|_{t=0} \\
&+ uf(U + \dot{U}_2 + t\dot{U}_1)\frac{d}{dt}[(D + \dot{D}_2 + t\dot{D}_1)]uf(V + \dot{V}_2 + t\dot{V}_1)^T|_{t=0} \\
&+ uf(U + \dot{U}_2 + t\dot{U}_1)(D + \dot{D}_2 + t\dot{D}_1)\frac{d}{dt}[uf(V + \dot{V}_2 + t\dot{V}_1)]^T|_{t=0}
\end{aligned} \tag{4.45}$$

Since $U \in \mathrm{St}(m,r), V \in \mathrm{St}(n,r)$ according to the vector transport by differentiated retraction on the Stiefel manifold [Hua13, Lemma 10.2.1], for $\dot{U}_1, \dot{U}_2 \in \mathrm{T}_U\mathrm{St}(m,r)$, it follows that

$$\frac{d}{dt}[uf(U + \dot{U}_2 + t\dot{U}_1)]|_{t=0} = \mathcal{T}_{\dot{U}_2}(\dot{U}_1), \tag{4.46}$$

and for $\dot{V}_1, \dot{V}_2 \in \mathrm{T}_V\mathrm{St}(n,r)$,

$$\frac{d}{dt}[uf(V + \dot{V}_2 + t\dot{V}_1)]|_{t=0} = \mathcal{T}_{\dot{V}_2}(\dot{V}_1), \tag{4.47}$$

where

$$\begin{aligned}
T_{\dot{U}_2}(\dot{U}_1) &= \mathrm{D}R_U(\dot{U}_2)[\dot{U}_1] \\
&= \mathrm{D}uf(U + \dot{U}_2)[\dot{U}_1] \\
&= R_U(\dot{U}_2)\Omega + (I - R_U(\dot{U}_2)(R_U(\dot{U}_2))^T)\dot{U}_1((R_U(\dot{U}_2))^T(U + \dot{U}_2))^{-1},
\end{aligned}$$

and R is (4.16), $\mathrm{vec}\{\Omega\} = ((R_U(\dot{U}_2))^T(U + \dot{U}_2) \oplus (R_U(\dot{U}_2))^T(U + \dot{U}_2))^{-1}\mathrm{vec}\{(R_U(\dot{U}_2))^T\dot{U}_1 - \dot{U}_1^T R_U(\dot{U}_2)\}$, $\oplus$ is the Kronecker sum, i.e., $A \oplus B = A \otimes I + I \otimes B$.

Substituting (4.46) and (4.47) into (4.45), yields

$$\mathcal{T}_{\eta}\xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1)(D + \dot{D}_2)uf(V + \dot{V}_2)^T + uf(U + \dot{U}_2)\dot{D}_1uf(V + \dot{V}_2)^T + uf(U + \dot{U}_2)(D + \dot{D}_2)(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T.$$

□

Similarly, the following proposition derives the vector transport by differentiated retraction of (4.18).

Proposition 26. *Let $X = UDV^T \in \mathcal{M}_r$, $\xi, \eta \in \mathrm{T}_X\mathcal{M}_r$. Assuming ξ and η have the following structure*

$$\xi = \dot{U}_1 D V^T + U \dot{D}_1 V^T + U D \dot{V}_1^T,$$
$$\eta = \dot{U}_2 D V^T + U \dot{D}_2 V^T + U D \dot{V}_2^T$$

then the vector transport by the differentiated retraction of (4.18) *is*

$$\mathcal{T}_\eta \xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1)(D + \dot{D}_2) qf(V + \dot{V}_2)^T + qf(U + \dot{U}_2)\dot{D}_1 qf(V + \dot{V}_2)^T + qf(U + \dot{U}_2)(D + \dot{D}_2)(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T, \tag{4.48}$$

where $\mathcal{T}_{\dot{U}_2}(\dot{U}_1)$ is a differentiated retraction on the compact Stiefel manifold [AMS08, Example 8.1.5] and $qf(\cdot)$ denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R.

Proof. Based on the definition of the vector transport by differentiated retraction and the QR-type retraction (4.18), the following hold

$$\begin{aligned}
\mathcal{T}_\eta \xi &= \frac{d}{dt} R_X(\eta + t\xi)|_{t=0} \\
&= \frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)(D + \dot{D}_2 + t\dot{D}_1) qf(V + \dot{V}_2 + t\dot{V}_1)^T]|_{t=0} \\
&= \frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)](D + \dot{D}_2 + t\dot{D}_1) qf(V + \dot{V}_2 + t\dot{V}_1)^T|_{t=0} \\
&+ qf(U + \dot{U}_2 + t\dot{U}_1)\frac{d}{dt}[(D + \dot{D}_2 + t\dot{D}_1)] qf(V + \dot{V}_2 + t\dot{V}_1)^T|_{t=0} \\
&+ qf(U + \dot{U}_2 + t\dot{U}_1)(D + \dot{D}_2 + t\dot{D}_1)\frac{d}{dt}[qf(V + \dot{V}_2 + t\dot{V}_1)]^T|_{t=0}
\end{aligned} \tag{4.49}$$

Since $U \in \mathrm{St}(m, r), V \in \mathrm{St}(n, r)$ according to the vector transport by differentiated retraction on the compact Stiefel manifold [AMS08], for $\dot{U}_1, \dot{U}_2 \in \mathrm{T}_U\mathrm{St}(m, r)$, it follows that

$$\frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)]|_{t=0} = \mathcal{T}_{\dot{U}_2}(\dot{U}_1), \tag{4.50}$$

and for $\dot{V}_1, \dot{V}_2 \in \mathrm{T}_V\mathrm{St}(n, r)$,

$$\frac{d}{dt}[qf(V + \dot{V}_2 + t\dot{V}_1)]|_{t=0} = \mathcal{T}_{\dot{V}_2}(\dot{V}_1), \tag{4.51}$$

where

$$\begin{aligned}
T_{\dot{U}_2}(\dot{U}_1) &= \mathrm{D}R_U(\dot{U}_2)[\dot{U}_1] \\
&= \mathrm{D}qf(U + \dot{U}_2)[\dot{U}_1] \\
&= R_U(\dot{U}_2)\rho_{\mathrm{skew}}(R_U(\dot{U}_2)^T \dot{U}_1 (R_U(\dot{U}_2)^T(U + \dot{U}_2))^{-1}) \\
&+ (I - R_U(\dot{U}_2)R_U(\dot{U}_2)^T)\dot{U}_1 (R_U(\dot{U}_2)^T(U + \dot{U}_2))^{-1},
\end{aligned}$$

and $\rho_{\text{skew}}(B)$ denotes the skew-symmetric term of the decomposition of a square matrix B into the sum of a skew-symmetric term and an upper triangular term, i.e.,

$$(\rho_{\text{skew}}(B))_{i,j} = \begin{cases} B_{i,j} & \text{if } i >, j \\ 0 & \text{if } i = j, \\ -B_{j,i} & \text{if } i < j. \end{cases}$$

Substituting (4.50) and (4.51) into (4.49), yields

$$\mathcal{T}_\eta \xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1)(D+\dot{D}_2)qf(V+\dot{V}_2)^T + qf(U+\dot{U}_2)\dot{D}_1 qf(V+\dot{V}_2)^T + qf(U+\dot{U}_2)(D+\dot{D}_2)(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T.$$

□

4.3.7 Action of the Hessian on a Fixed-rank Manifold

To use the second order information, the action of the Hessian on a vector is required.

Proposition 27. *For any $X = UDV^T \in \mathcal{M}_r$ and $\eta \in \mathrm{T}_X\mathcal{M}_r$, the action of the Riemannian Hessian of a cost function f at X on the direction vector η satisfies*

$$\text{Hess } f_r(X)[\eta] = \bigtriangledown_\eta \text{grad } f(X) = \mathrm{P}_X(\text{Dgrad } f_r(X)[\eta]),$$

where

$$\begin{aligned} \text{Dgrad } f_r(X)[\eta] = &-2[U_\perp(U_\perp^T \eta V D^{-1})U^T + U(U_\perp^T \eta V D^{-1})^T U_\perp^T]\text{vec}^{-1}(W\text{vec}(R-X)) \\ &- 2(UU^T)\text{vec}^{-1}(W\text{vec}(R-\eta)) - 2\text{vec}^{-1}(W\text{vec}(R-\eta))VV^T \\ &- 2\text{vec}^{-1}(W\text{vec}(R-X))[V_\perp(D^{-1}U^T\eta V_\perp)^T V^T + V(D^{-1}U^T\eta V_\perp)V_\perp^T] \\ &+ 2[U_\perp(U_\perp^T \eta V D^{-1})U^T + U(U_\perp^T \eta V D^{-1})^T U_\perp^T]\text{vec}^{-1}(W\text{vec}(R-X))VV^T \\ &+ 2UU^T\text{vec}^{-1}(W\text{vec}(R-\eta))VV^T \\ &+ 2UU^T\text{vec}^{-1}(W\text{vec}(R-X))[V_\perp(D^{-1}U^T\eta V_\perp)^T V^T + V(D^{-1}U^T\eta V_\perp)V_\perp^T]. \end{aligned}$$

Proof. The gradient of f at X on $\mathcal{M}_r$ is given by:

$$\begin{aligned} \text{grad} f_r(X) &= \mathrm{P}_X(-2\text{vec}^{-1}(W\text{vec}\{R-X\})) \\ &= -2UU^T(\text{vec}^{-1}(W\text{vec}\{R-X\})) - 2(\text{vec}^{-1}(W\text{vec}\{R-X\}))VV^T \\ &+ 2UU^T(\text{vec}^{-1}(W\text{vec}\{R-X\}))VV^T \end{aligned} \tag{4.52}$$

where $\text{vec}^{-1}(W\text{vec}\{R-X\})$ represents reshaping the vector $W\text{vec}\{R-X\}$ as an $m \times n$ matrix. Since $\mathcal{M}_r$ is a Riemannian submanifold of a Euclidean space, according to [AMS08, (5.15)] ,

$$\text{Hess} f_r(X)[\eta] = \nabla_\eta \text{grad} f_r(x) = \mathrm{P}_x(\text{Dgrad} f_r(x)[\eta]), \tag{4.53}$$

where $Dg(x)[H]$ is a directional derivative of g at x along H. Differentiating (4.52) according to (4.53) yields a matrix representation of the action of the Hessian of f_r at X along η.

$$\begin{aligned}\mathrm{Dgrad} f_r(x)[\eta] &= -2(UU^T)'(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-X\})) - 2UU^T(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-\eta\}))\\ &\quad - 2(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-\eta\}))VV^T\\ &\quad - 2(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-X\}))(VV^T)' + 2(UU^T)'(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-X\}))VV^T\\ &\quad + 2UU^T(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-\eta\}))VV^T + 2UU^T(\mathrm{vec}^{-1}(W\mathrm{vec}\{R-X\}))(VV^T)'.\end{aligned} \tag{4.54}$$

Next, $(UU^T)'$ and $(VV^T)'$ must be derived. Since $U \in \mathrm{St}(m,r)$, $\dot{U} = U\Omega_U + U_\perp K_U$, where $\Omega_U^T = -\Omega_U, K_U \in \mathbb{R}^{(m-r)\times r}$, it follows that

$$\begin{aligned}(UU^T)' &= \dot{U}U^T + U\dot{U}^T = (U\Omega_U + U_\perp K_U)U^T + U(U\Omega_U + U_\perp K_U)^T\\ &= U(\Omega_U + \Omega_U^T)U^T + U_\perp K_U U^T + UK_U^T + U_\perp^T\\ &= U_\perp K_U U^T + UK_U^T U_\perp^T\\ &= U_\perp(U_\perp^T \dot{X} V D^{-1})U^T + U(U_\perp^T \dot{X} V D^{-1})^T U_\perp^T.\end{aligned} \tag{4.55}$$

Similarly, $(VV^T)'$ is ,

$$(VV^T)' = V(D^{-1}U^T\dot{X}V_\perp)V_\perp^T + V_\perp(D^{-1}U^T\dot{X}V_\perp)^T V^T. \tag{4.56}$$

Substituting (4.55) and (4.56) into (4.54), yields

$$\begin{aligned}\mathrm{Dgrad} f_r(X)[\eta] &= -2[U_\perp(U_\perp^T\eta V D^{-1})U^T + U(U_\perp^T\eta V D^{-1})^T U_\perp^T]\mathrm{vec}^{-1}(W\mathrm{vec}(R-X))\\ &\quad - 2(UU^T)\mathrm{vec}^{-1}(W\mathrm{vec}(R-\eta)) - 2\mathrm{vec}^{-1}(W\mathrm{vec}(R-\eta))VV^T\\ &\quad - 2\mathrm{vec}^{-1}(W\mathrm{vec}(R-X))[V_\perp(D^{-1}U^T\eta V_\perp)^T V^T + V(D^{-1}U^T\eta V_\perp)V_\perp^T]\\ &\quad + 2[U_\perp(U_\perp^T\eta V D^{-1})U^T + U(U_\perp^T\eta V D^{-1})^T U_\perp^T]\mathrm{vec}^{-1}(W\mathrm{vec}(R-X))VV^T\\ &\quad + 2UU^T\mathrm{vec}^{-1}(W\mathrm{vec}(R-\eta))VV^T\\ &\quad + 2UU^T\mathrm{vec}^{-1}(W\mathrm{vec}(R-X))[V_\perp(D^{-1}U^T\eta V_\perp)^T V^T + V(D^{-1}U^T\eta V_\perp)V_\perp^T].\end{aligned}$$

Finally, the action of the Hessian of a cost function f at X in the direction of η satisfies

$$\mathrm{Hess}\ f_r(X)[\eta] = \bigtriangledown_\eta \mathrm{grad}\ f_r(X) = \mathrm{P}_X(\mathrm{Dgrad} f_r(X)[\eta]).$$

□

4.3.8 Some Observations and Improvements on the Methods using the Double Minimization Modification

In Section 4.2.2, the novel reformulation of the weighted low-rank approximation problem (4.1) as a double minimization problem (4.3) by Manton et al. [MMH03] was given. This reformulation allows them to minimize an associated cost function (4.5) on a Grassmann manifold. Two algorithms are discussed in [MMH03]: a linearly convergent steepest descent algorithm [MMH03, Algorithm 11] and a quadratic convergent Newton step algorithm [MMH03, Algorithm 14]; and the Riemannian gradient and Hessian required by the algorithms are also derived. The gradient of the cost function $f(N)$ is

$$\text{grad } f = 2N_\perp^T(R-B)^T A, \tag{4.57}$$

where $A \in \mathbb{R}^{m\times(n-k)}$ and $B \in \mathbb{R}^{m\times n}$ are the unique matrices that satisfy

$$\begin{aligned} \text{vec}\{A\} &= [(N\otimes I_m)^T W^{-1}(N\otimes I_m)]^{-1}\text{vec}\{RN\}, \\ \text{vec}\{B\} &= W^{-1}\text{vec}\{AN^T\}. \end{aligned} \tag{4.58}$$

The Hessian of $f(N)$ is

$$\begin{aligned} H = 2\{&(I_{n-k}\otimes(R-B)N_\perp)^T[(N\otimes I_m)^T W^{-1}(N\otimes I_m)]^{-1}(I_{n-k}\otimes(R-B)N_\perp) \\ &- (I_{n-k}\otimes(R-B)N_\perp)^T[(N\otimes I_m)^T W^{-1}(N\otimes I_m)]^{-1}(N\otimes I_m)^T W^{-1}(N_\perp\otimes A)C \\ &- C^T(N_\perp\otimes A)^T W^{-1}(N\otimes I_m)[(N\otimes I_m)^T W^{-1}(N\otimes I_m)]^{-1}(I_{n-k}\otimes(R-B)N_\perp) \\ &- C^T(N_\perp\otimes A)^T(W^{-1}-W^{-1}(N\otimes I_m)[(N\otimes I_m)^T W^{-1}(N\otimes I_m)]^{-1}(N\otimes I_m)^T W^{-1})(N_\perp\otimes A)C\}, \end{aligned} \tag{4.59}$$

where $C \in \mathbb{R}^{k(n-k)\times k(n-k)}$ is the unique matrix satisfying for all $K \in \mathbb{R}^{k\times(n-k)}$,

$$\text{vec}\{K^T\} = C\text{vec}\{K\}. \tag{4.60}$$

In the numerical study section [MMH03, Section VI], since the Newton method is locally convergent, its initial point is generated by several iterations of the globally convergent steepest descent algorithm. For the steepest descent, they use a Riemannian version of the Armijo step-size rule.

Newton's method requires the solution of a linear equation $H\text{vec}\{K\} = -\text{vec}\{\text{grad}f\}$ for the matrix $K \in \mathbb{R}^{k\times(n-k)}$, where grad$f$ and H are given by (4.57) and (4.59), respectively. No details were given on the method used to solve this system. In order to provide the best possible performance data for the Newton method of Manton et al., a version that employs the "inverse-free"

truncated conjugate gradient method [Ste83] in the experiments presented below. The truncated conjugate gradient method has been used extensively in Riemannian optimization algorithms, for details see [Bak08] and [Hua13].

To use the truncated conjugate gradient, the action of the Hessian on the Grassmann manifold $Gr(n, n-k)$ must be characterized in a computationally efficient manner as done in the following Proposition.

Proposition 28. *For any $N \in Gr(n, n-k)$ and $\eta \in \mathrm{T}_N Gr(n, n-k)$, the action of the Riemannian Hessian of a cost function f at N on the direction vector η satisfies*

$$\begin{aligned}\mathrm{Hess}f(N)[\eta] = (I_n - NN^T)[-2(\eta N^T + N\eta^T)(X-B)^T A - 2(I_n - NN^T)(\mathrm{vec}^{-1}(d\mathrm{vec}\{B\}))A \\ + 2(I_n - NN^T)(X-B)^T(\mathrm{vec}^{-1}(d\mathrm{vec}\{A\}))].\end{aligned}$$

where

$$d\mathrm{vec}\{A\} = [(N \otimes I_m)^T W^{-1}(N \otimes I_m)]^{-1}[\mathrm{vec}\{(R-B)\eta\} - (N \otimes I_m)^T W^{-1}\mathrm{vec}\{A\eta^T\}],$$

$$d\mathrm{vec}\{B\} = W^{-1}(N \otimes I_m)d\mathrm{vec}\{A\} + W^{-1}\mathrm{vec}\{A\eta^T\}.$$

Proof. The vertical space of a Grassmann manifold $Gr(n, n-k)$ is given by [AMS08]

$$\mathrm{T}_N Gr(n, n-k) = \{N_\perp K : K \in \mathbb{R}^{k \times (n-k)}\},$$

and the orthogonal projection onto the vertical space of $Gr(n, n-k)$ at N is

$$P_N Z = N_\perp N_\perp^T Z = (I - NN^T)Z, \quad \forall Z \in \mathbb{R}^{m \times n}.$$

Since the Euclidean gradient of the cost function f is $\bigtriangledown f(N) = 2(R-B)^T A$, where $\mathrm{vec}\{A\}, \mathrm{vec}\{B\}$ are defined in (4.58). The projection onto the vertical space of the Riemannian gradient on $Gr(n, n-k)$ is

$$\mathrm{grad}f = 2(I_n - NN^T)(R-B)^T A. \tag{4.61}$$

The Riemannian Hessian is then computed by

$$\mathrm{Hess}f(N)[\eta] = P_N(\mathrm{Dgrad}f(N)[\eta]). \tag{4.62}$$

Differentiating (4.61) according to (4.62) yields a matrix representation of the action of the Hessian of f at N along η.

$$\begin{aligned}\mathrm{Dgrad}f(N)[\eta] = -2(\eta N^T + N\eta^T)(X-B)^T A - 2(I_n - NN^T)(\mathrm{vec}^{-1}(d\mathrm{vec}\{B\}))A \\ + 2(I_n - NN^T)(X-B)^T(\mathrm{vec}^{-1}(d\mathrm{vec}\{A\})).\end{aligned}$$

where

$$d\text{vec}\{A\} = [(N \otimes I_m)^T W^{-1}(N \otimes I_m)]^{-1}[\text{vec}\{(R-B)\eta\} - (N \otimes I_m)^T W^{-1}\text{vec}\{A\eta^T\}],$$
$$d\text{vec}\{B\} = W^{-1}(N \otimes I_m)d\text{vec}\{A\} + W^{-1}\text{vec}\{A\eta^T\}.$$

□

The efficient version of the Newton algorithm of Manton et al. with truncated conjugate gradient is given in Algorithm 3.

Algorithm 3 Manton's Newton Method with Truncated CG

Require: Data matrix $R \in \mathbb{R}^{m\times n}$, weighting matrix $W \in \mathbb{R}^{mn\times mn}$, rank specification k. Convergence tolerance $\epsilon > 0$. Scalars $\bar{\alpha} > 0$, $c, \beta, \sigma \in (0,1)$.

1: Choose starting point $N_1 \in \mathbb{R}^{n\times(n-k)}$ and $N_{1,\perp} \in \mathbb{R}^{n\times k}$ such that $\begin{bmatrix} N_1 & N_{1,\perp}\end{bmatrix}^T \begin{bmatrix} N_1 & N_{1,\perp}\end{bmatrix} = I$. Set $i = 1$.

2: **while** $\|\text{grad } f_i\| \geq \epsilon$ **do**

3: Obtain $K_i \in \mathbb{R}^{k\times(n-k)}$ by (approximately) solving

$$\text{Hess } f(N_i)K = -\text{grad } f(N_i) \tag{4.63}$$

where f is given by (4.5).

4: Select N_{i+1} such that

$$f(N_i) - f(N_{i+1}) \geq c(f(N_i) - f(R_{N_i}(t_i^A K_i))), \tag{4.64}$$

where t_i^A is the Armijo step-size for the given $\bar{\alpha}, \beta, \sigma$. The retraction R is renormalizing $\begin{bmatrix} N_{i+1} & N_{i+1,\perp}\end{bmatrix}$ by setting $\begin{bmatrix} N_{i+1} & N_{i+1,\perp}\end{bmatrix} := qf(N_i + t_i^A N_{i,\perp}K_i)$, where the symbol $qf(\cdot)$ denotes the Q-factor of the QR decomposition of its matrix argument.

5: **end while**

6: Compute

$$\text{vec}\{X\} = \text{vec}\{R\} - W^{-1}(N \otimes I_m)[(N \otimes I_m)^T W^{-1}(N \otimes I_m)]^{-1}(N \otimes I_m)^T\text{vec}\{R\}.$$

Since the computation of the Riemannian Hessian of this cost function is expensive, a Riemannian BFGS-type algorithm for minimizing $f(N)$ on the Grassmann manifold was derived by Brace and Manton in 2006 [BM06]. As in the Euclidean case, the advantage of a BFGS algorithm is that, compared with the Newton method, is lower computational complexity. While BFGS has superlinear convergence rather than quadratic, it is often significantly better than Newton in terms

of total computation time. An outline of the the Riemannian BFGS algorithm called the Improved BFGS by Brace and Manton [BM06, Algorithm 2] is given in Algorithm 4.

Algorithm 4 Manton's Improved BFGS

Require: data matrix $R \in \mathbb{R}^{m\times n}$, weighting matrix $W \in \mathbb{R}^{mn\times mn}$, rank specification k, convergence tolerance $\epsilon > 0$. Scalars $\bar{\alpha} > 0$, $c, \beta, \sigma \in (0,1)$.

1: Choose starting point $N_1 \in \mathbb{R}^{n\times(n-k)}$ and $N_{1,\perp} \in \mathbb{R}^{n\times k}$ such that $\begin{bmatrix} N_1 & N_{1,\perp} \end{bmatrix}^T \begin{bmatrix} N_1 & N_{,\perp} \end{bmatrix} = I$. Set the initial Hessian estimate to $B_1 = I \in \mathbb{R}^{k(n-k)\times k(n-k)}$. Set $i = 1$.

2: **while** $\|\text{grad } g_i\| > \epsilon$ **do**

3: Obtain p_i by solving $B_i p_i = -\text{grad } g_i$, where grad g_i is defined in (4.57).

4: Select N_{i+1} such that

$$f(N_i) - f(N_{i+1}) \geq c(f(N_i) - f(R_{N_i}(t_i^A p_i))), \tag{4.65}$$

where t_i^A is the Armijo step-size for the given $\bar{\alpha}, \beta, \sigma$. The retraction R is renormalizing $\begin{bmatrix} N_{i+1} & N_{i+1,\perp} \end{bmatrix}$ by setting $\begin{bmatrix} N_{i+1} & N_{i+1,\perp} \end{bmatrix} := qf(N_i + t_i^A N_{i,\perp} p_i)$, where the symbol $qf(\cdot)$ denotes the Q-factor of the QR decomposition of its matrix argument.

5: Set $\Delta g = \text{vec}\{\text{grad } g_{i+1} - N_{i+1,\perp}\text{grad } g_i\}$, $\Delta s = \text{vec}\{N_{i+1,\perp} t_i p_i\}$.

6: Compute the BFGS update

$$B_{i+1} = B_i + \frac{\Delta g \Delta g^T}{\Delta s^T \Delta g} - \frac{B_i \Delta s \Delta s^T B_i^T}{\Delta s^T B_i \Delta s},$$

7: Set $i = i + 1$;

8: **end while**

9: Compute

$$\text{vec}\{X\} = \text{vec}\{R\} - W^{-1}(N \otimes I_m)[(N \otimes I_m)^T W^{-1}(N \otimes I_m)]^{-1}(N \otimes I_m)^T \text{vec}\{R\}.$$

Note that the algorithm does not use any explicit vector transport. This is motivated by Brace and Manton heuristically in order to reduce the complexity of the iteration and its effectiveness was explored only empirically. The Improved BFGS algorithm of Brace and Manton, also only requires satisfying Riemannian Armijo conditions in the line search procedure, which does not guarantee the search directions p_i are descent directions.

Recent advances in the theoretical understanding of vector transport, however, allow an explanation of this heuristic approach. The Improved BFGS algorithm of Brace and Manton is equivalent to the intrinsic dimensional approach discussed in Huang's dissertation [Hua13, Section 9.5], where

the matrix representations of the vector transport in Step 6 of Algorithm 4 is an identity matrix. So, in fact, vector transport is done implicitly since the bases of the tangent spaces are in fact moved continuously and therefore transport by parallelization is performed.

Furthermore, in [Hua13, Algorithm 3], Huang also considered Riemannian Wolfe conditions in the line search procedure, i.e.

$$\frac{d}{dt} f(R(\alpha p_i))|_{\alpha=t_i} \geq c_2 \frac{d}{dt} f(R(\alpha p_i))|_{\alpha=0}, \tag{4.66}$$

where $0 < c_2 < 1$ is a constant, and shows that convergence of the Riemannian Restricted Broyden Family, including RBFGS, can be guaranteed when they are used.

So Brace and Manton's heuristic reduction of the computational complexity of their Riemannain BFGS algorithms can be analyzed rigorously. More importantly for this dissertation, the rigorously analyzed and efficiently implemented RBFGS algorithm of Huang's dissertation [Hua13, Algorithm 3] with intrinsic dimensional approach and the specific "qf" retraction [Hua13, Equation (10.2.3)] in each iteration can be used on the reformulated cost function of Manton et al. in the experiments.

4.4 Experiments

4.4.1 Test Problems

This section provides numerical experiments illustrating the performance of the modified Riemannian optimization algorithm compared with the other approaches. The results presented are obtained by implementing the different algorithms in Matlab (Version 7.10.0) on a Mac platform with 2.4 GHz and 4 GB memory.

In Section 4.4.2, the default values of some parameters are given. In Section 4.4.3, the performance for different values of parameters ϵ_1 and ϵ_2 when approximating matrices with singular values that have a clear gap and those that have an exponential decay are presented. In particular, the choice of the parameters for solving the optimization problem to get close to the true rank of the data matrix and for finding a lower rank but acceptable approximation. In practice, an upper bound, k, on the rank of the approximating matrix is often given. The effect of different choices of k are shown in Section 4.4.4 for low-rank data matrices and the results of different methods are compared. In Section 4.4.5, structured weighting matrices, i.e., diagonal and block-diagonal, are considered. The influence of the retraction and its invariance to the particular factors in the decomposition $X = UDV^T$, or lack of invariance, is presented in Section 4.4.6. Finally, the performance

of different rank reduce methods are shown in Section 4.4.7 and the performance of different inner algorithms are shown in Section 4.4.8.

As reviewed in previous sections, some of the other approaches reformulated the original problem (4.1), the cost function used for each method is listed in Table 4.1. The choice of algorithm for each approach is discussed below.

Table 4.1: Cost function used by the approaches.

Alternating Projections Method (APM)	cost function (4.1)
Double Minimization Method (DMM)	cost function (4.5)
EW-TLS Method	cost function (4.9)
Schneider and Uschmajew's Line-search Method (SULS)	cost function (4.1)
Modified Riemannian Optimization Method (MROM)	cost function (4.1)

4.4.2 Algorithm Parameters and Notations

The parameters in the modified Riemannian optimization method (MROM), i.e., Algorithm 2, are set as follows. The Riemannian trust-region method (RTR-Newton) [Bak08] is used as the inner algorithm in Algorithm 2. Unless stated otherwise, the parameters ϵ_1 and ϵ_2 are $\sqrt{3}$ and 10^{-4} respectively. Parameter ϵ_4 is $\frac{\epsilon_1}{2}$. The initial stopping criterion, ϵ_3, on the fixed-rank manifold is 1 and the initial parameter for the rank detection in Algorithm 1 is $\Delta_0 = 10^{-5}$. Both the parameters τ_1 and τ_2 are 0.1. In order to avoid the effects of noise and get rid of small singular values, 10^{-8} is used as an upper bound of Δ. The polar-decomposition-type retraction and rank-related retraction (4.16), (4.33) are used in the tests in Sections 4.4.3-4.4.5 since they are invariant to the choice of factors in the decomposition of $X = UDV^T$ and efficient in computation.

Manton's Newton method with truncated CG (Algorithm 3) has been observed to be consistently faster than Manton's improved BFGS method (Algorithm 4). Therefore, in the following, Algorithm 3 is used when testing the Double Minimization Method (DMM). Additionally, the evaluation of cost function (4.5) in Algorithm 3 involves the inverse of the weighting matrix W. In order to avoid the computation of the inverse, the Cholesky decomposition of the weighting matrix W is computed in preprocessing and the effect of the inverse determined by the solution of two triangular systems.

The truncated CG iteration [AMS08, Section 7.3.2] algorithm is used in both Algorithm 3 and the inner iteration of the RTR-Newton method in Algorithm 2. The parameters θ and κ in the

truncated CG iteration stopping criteria [AMS08, (7.10)] are 1 and 0.1 respectively. The parameters τ_1 and τ_2 in the trust region update are 0.25 and 2 respectively. The initial trust region radius is 1. The Accelerated line search algorithm [AMS08, Algorithm 1] is used in Algorithm 2, Algorithm 3, and SULS. The constants α and β in the Armijo conditions for the line search in these methods are 0.01 and 0.99 respectively.

For numerical simulated data, two default cases are used: (1) fully random $m \times n$ matrices R of rank r, $R = R_1 R_2^T$, where $R_1 \in \mathbb{R}^{m\times r}$ and $R_2 \in \mathbb{R}^{n\times r}$ are each generated according to a Gaussian distribution with zero mean and unit standard deviation (with Matlab's RANDN); (2) random low-rank matrices with chosen singular values, $R = U_1 S U_2^T$, where S is a diagonal matrix with chosen singular values, $U_1 \in \mathbb{R}^{m\times r}, U_2 \in \mathbb{R}^{n\times r}$ are orthogonal matrices generated by Matlab's ORTH and RANDN. The initializations of each method are as follows. Algorithm 2 and SULS are started with a randomly generated rank-1 matrix defined by $[U_0, D_0, V_0]$, where D_0 is a random number obtained by Matlab's RAND and $U_0 \in \mathbb{R}^{m\times 1}, V_0 \in \mathbb{R}^{n\times 1}$ are obtained by Matlab's ORTH and RAND. Algorithm 3 is started with a random $n \times (n-k)$ matrix and APM is started with a random $m\times(m-k)$ matrix. Both are generated by Matlab's QR and RAND. Other data generation choices are explicitly noted when used.

Algorithm 2 and SULS, are stopped when the norm of the final gradient on the fixed-rank manifold over the norm of initial full gradient is less than 10^{-8} while Algorithm 3 and APM are stopped when the norm of final gradient over the norm of initial gradient is less than 10^{-7}.

The reported time are wall clock times. Some machine independent values are also reported including final value of the cost function (4.1), the relative error, which is computed as $\frac{\|R-X\|_W}{\|R\|_W}$ and the error $\|R - X\|_F$. All computations are in IEEE double precision. The notation used when reporting the experimental results is given in Table 4.2.

4.4.3 Performance of Different Parameters

4.4.3.1 Performance of High Probability of Finding True Rank. The first set of experiments evaluates the ability of MROM, implemented in Algorithm 2, to find the rank of a matrix and an associated approximation for matrices where the spectrum has a clear separation defining the rank. Of particular interest is the effect of the initial matrix on the rank and approximation found.

Table 4.2: Notation for reporting the experimental results.

R_err	relative error $\frac{\|R-X\|_W}{\|R\|_W}$
err	error $\|R-X\|_F$
SVerr	error between the given r singular values and the r singular values found by algorithm
f	final value of the cost function (4.1)
gf_r	The norm of the final gradient on the fixed-rank manifold $\mathcal{M}_r$
gf_F0	The norm of the initial full gradient
nf	number of function evaluations
ng	number of gradient evaluations
nH	number of operations of the form $\mathcal{H}\eta$
nV	number of vector transports
nR	number of retraction evaluations
nR_r	number of rank-related retraction evaluations
time	average time (seconds)

The data matrices R are chosen as random 50×50 matrices with rank 5. The weighting matrices $W = U\Sigma U^T$, where $U \in \mathbb{R}^{2500\times 2500}$ are random orthogonal matrices generated by Matlab's QR and RAND. The 2500 singular values of the weighting matrix are generated by Matlab function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution matrix on the interval $[0.5, 1.5]$. The upper bound on rank, k, is 50. Several initial rank i matrix $X_0^{(i)} = U_0^{(i)} D_0^{(i)} (V_0^{(i)})^T$, where $U_0^{(i)} \in \mathbb{R}^{50\times i}, V_0^{(i)} \in \mathbb{R}^{50\times i}$ are random matrices generated with Matlab's RAND and ORTH, $D_0^{(i)}$ is a random $i \times i$ diagonal matrix with uniformly distributed diagonal elements.

Results reported in Table 4.3 and 4.4 are the average of 50 runs for different data matrices R, weighting matrices W and initial points realizations with upper bound of rank detection $\Delta = 10^{-8}$ and $\Delta = 0$ respectively.

When $\Delta = 10^{-8}$, the results show that for all initial points the true rank is eventually discovered. When $\Delta = 0$, the notion of numerical rank is essentially ignored which is much more restrictive than is done in practice. While not all runs result in a rank of 5, all of the ranks accepted are greater than or equal to 5, as desired when $\Delta = 0$, and a high probability of finding 5 is seen over all of the runs.

Therefore, $\Delta = 10^{-8}$ is a reasonable value when finding the true rank while allowing some influence of numerical rank. Also, although the value of the bound k is 50 (the highest possible),

the rank does not increase to 50 for any iteration of any run. When the initial rank is less than 5, the rank of any iterate usually only increases to 7 (a small number of times to 9 or 10) and then drops back quickly to 5. This implies even the value of k is chosen large, the rank is not destined to increase to the upper bound k if the true rank is small, as often happens with other more simplistic rank selection heuristics, and time and space efficiency is maintained.

Table 4.3: Approximation with different rank initial conditions. $\Delta = 10^{-8}$. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

	rank	R_err	err	f	time(s)	gf_r	gf_r/gf_F0
$X_0^{(1)}$	5	4.872_{-15}	4.494_{-06}	1.153_{-11}	3.445_{+00}	1.289_{-06}	1.640_{-08}
$X_0^{(4)}$	5	1.332_{-17}	2.284_{-06}	3.957_{-12}	3.784_{+00}	6.479_{-07}	8.406_{-09}
$X_0^{(5)}$	5	2.826_{-08}	3.913_{-05}	6.896_{-10}	1.005_{+00}	1.107_{-05}	1.438_{-07}
$X_0^{(6)}$	5	3.351_{-15}	4.646_{-06}	1.148_{-11}	2.735_{+00}	1.294_{-06}	1.680_{-08}
$X_0^{(10)}$	5	1.921_{-09}	5.673_{-06}	1.514_{-11}	4.256_{+00}	1.597_{-06}	1.980_{-08}

Table 4.4: Approximation with different rank initial conditions. $\Delta = 0$. The number in the parenthesis indicates the ratio of the numerical rank equals the true rank. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

	rank	R_err	err	f	time(s)	gf_r	gf_r/gf_F0
$X_0^{(1)}$	5.04 (49/50)	4.872_{-15}	4.496_{-06}	1.153_{-11}	$3.439e+00$	1.290_{-06}	1.641_{-08}
$X_0^{(4)}$	5.1 (47/50)	1.332_{-17}	2.293_{-06}	3.957_{-12}	3.813_{+00}	6.517_{-07}	8.456_{-09}
$X_0^{(5)}$	5	2.826_{-08}	3.913_{-05}	6.896_{-10}	1.000_{+00}	1.107_{-05}	1.438_{-07}
$X_0^{(6)}$	5.06 (47/50)	3.351_{-15}	4.655_{-06}	1.148_{-11}	2.730_{+00}	1.297_{-06}	1.684_{-08}
$X_0^{(10)}$	5.04 (49/50)	1.921_{-09}	5.673_{-06}	1.514_{-11}	4.267_{+00}	1.598_{-06}	1.980_{-08}

Next the advantages of the dynamical rank updating are demonstrated. The data matrices are randomly generated 100×100 matrices with rank 17. The weighting matrix $W = \text{diag}\{W_1, \cdots, W_{100}\}$. Each $W_i = U_i\Sigma_i U_i^T$ is a positive definite symmetric matrix, where $U_i \in \mathbb{R}^{100\times100}$ is random orthogonal matrices generated with Matlab's ORTH and RANDN, Σ_i are diagonal matrices with singular values generated with Matlab's function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution vector on the interval $[0.5, 1.5]$. The rank upper bound, k, is 100. The deterministic rank increment strategy used in many papers alternates between fixed-rank optimization and a fixed rank change. To avoid over-estimating the rank, the fixed rank change

is usually a rank change of 1 or 2. In the following, MROM is compared with fixed rank changes of 1, 2 and 4. All of fixed rank changes hit 17 exactly, so the rank change of 4, usually avoided in practice, is an optimistic approach that is suitable for this problem. If MROM is competitive or superior to the rank change of 4 then it is clear evidence of the robustness and effectiveness of MROM.

For MROM, different values of ϵ_1 are tested. For the fixed rank changes, a local minima on each fixed-rank manifold is sought and the local iteration stopped when the norm of fixed-rank manifold gradient is less than 10^{-5}.

Results reported in Table 4.5 are the average of 50 runs for different data matrices R, weighting matrices W and initial matrices. Figure 4.1 shows the rank changes as a function of the iteration numbers and the computational times for a representative single run. The results demonstrate that different values of ϵ_1 give different rank update patterns for MROM but all reach the same final rank of 17 as desired. When reaching almost the same relative error and error, the computational times of MROM are always less than than the practical fixed rank changes of 1 and 2. Furthermore, when $\epsilon_1 = tan(70^o)$, MROM increases the rank more aggressively than the optimistic fixed change by 4. However, MROM does not require finding the an approximate minimizer on each fixed-rank manifold and therefore does not expend unnecessary computational effort. Table 4.5 shows the relative errors and absolute errors of the different methods are almost the same, but computational cost of MROM with $\epsilon_1 = tan(70^o)$ is significantly less than the practical fixed rank changes of 1 and 2, and less than the optimistic fixed rank change of 4.

Table 4.5: Approximation with different rank update. $\epsilon_2 = 10^{-5}, \Delta = 10^{-8}$. The number in the parenthesis indicates the ratio of the rank increases to 17 is 44 out of 50. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

method	rank	f	R_err	err	time(s)	gf_r	gf_r/gf_F0
$\epsilon_1 = \tan(60^o)$	17.18(44/50)	5.955_{-13}	9.165_{-17}	1.246_{-06}	5.428_{+00}	3.317_{-07}	1.135_{-09}
$\epsilon_1 = \tan(70^o)$	17	7.833_{-14}	3.247_{-13}	6.110_{-07}	2.092_{+00}	1.262_{-07}	4.397_{-10}
$\epsilon_1 = \tan(80^o)$	17	1.613_{-14}	1.347_{-11}	1.971_{-07}	3.343_{+00}	4.114_{-08}	1.430_{-10}
rank-1 update	17	5.724_{-14}	3.299_{-13}	1.676_{-07}	1.316_{+01}	3.854_{-08}	1.354_{-10}
rank-2 update	17	8.459_{-13}	4.455_{-15}	1.917_{-06}	7.497_{+00}	3.966_{-07}	1.374_{-09}
rank-4 update	17	1.148_{-12}	1.246_{-15}	2.975_{-06}	4.259_{+00}	6.992_{-07}	2.431_{-09}

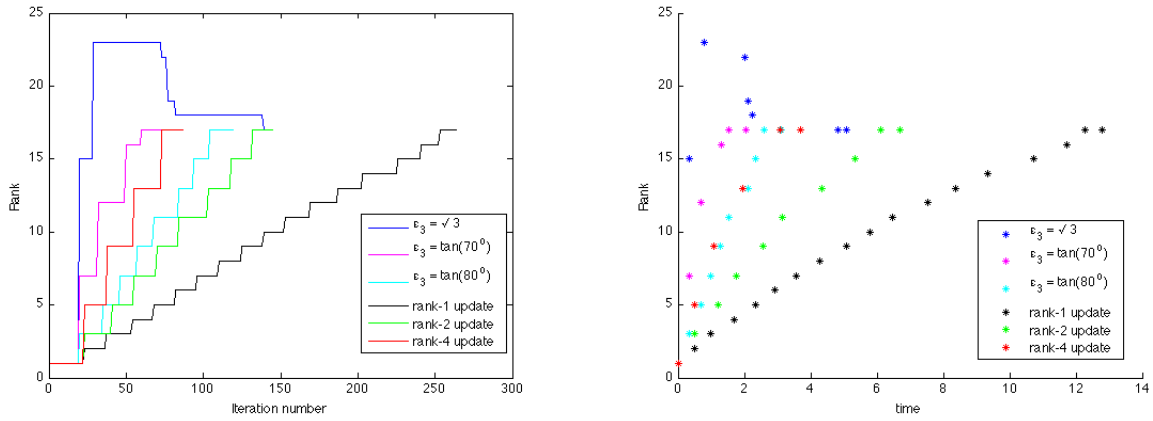

Figure 4.1: Different rank update.

4.4.3.2 Performance of Rank Approximation. In this set of experiments, data matrices R with an exponential decay of singular values are considered. For such matrices, finding the rank of R computationally is challenging. Although it is difficult to find the true rank, MROM, as implemented in Algorithm 2, gives an efficient way to get a rank approximation consistent with the precision specified by the choices of ϵ_1 and ϵ_2. Consider a random 100-by-100 data matrix R with singular values $2^{1-i}, i = 1, 2, \cdots, 100$. This matrix has full rank analytically but it is very ill conditioned and the numerical rank with singular values greater than 10^{-8} is 27. In order to compare the numerical error with the theoretical error, the weighting matrix W is taken as the identity. The value of the upper bound, k, is 100, which is the size of the data matrix R. Since the norm of initial full gradient is always between 2 and 10, ϵ_2 is initialized at 10, and decreased by a factor of 10 for each experiment. For each ϵ_2, three different values of ϵ_1 – 0.5, 1, $\sqrt{3}$ – are tested.

Figures 4.2, 4.3 and Table 4.6 show the average of results for each (ϵ_1, ϵ_2) pair run 100 times with different R, W, and initial X. The true error is $\|R - X_*\|_F = \min_{\text{rank}(X)\leq r} \|R - X\|_F = \sqrt{(2^{-r})^2 + \cdots + (2^{-99})^2}$ and is matched well by the computed errors. Furthermore, the rank estimation is as expected given the design of Algorithm 2. When ϵ_2 is small, the algorithm is expected to determine a rank close to the true numerical rank as determined by Δ. Approximate optimization results when ϵ_2 is increased and an approximation is found that requires less space and time at the cost of an increase, hopefully small, in the approximation error.

In the results, for each ϵ_1 and ϵ_2, the same rank is obtained over the 100 runs except when ϵ_2 is very small. The results demonstrate that as desired when ϵ_2 is large, increasing the rank of the approximation is made difficult. As the value of ϵ_2 decreases, the rank of the approximation increases. Meanwhile, the values of relative error, error and cost function decrease. The rank stops at 27 when ϵ_2 reaches 10^{-8}. Therefore, different values of ϵ_2 can be chosen depending on the accuracy/time/space tradeoffs required in specific applications.

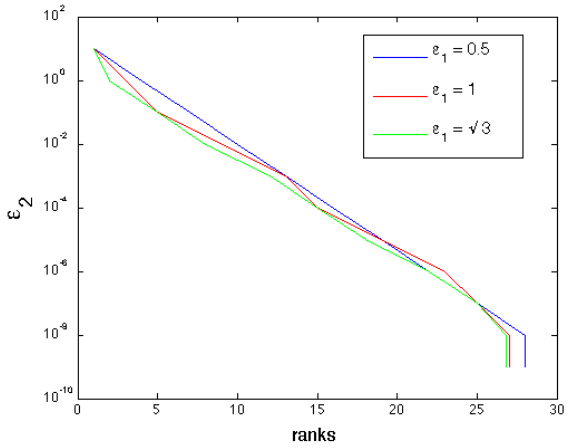

Figure 4.2: ϵ_2 versus rank approximation

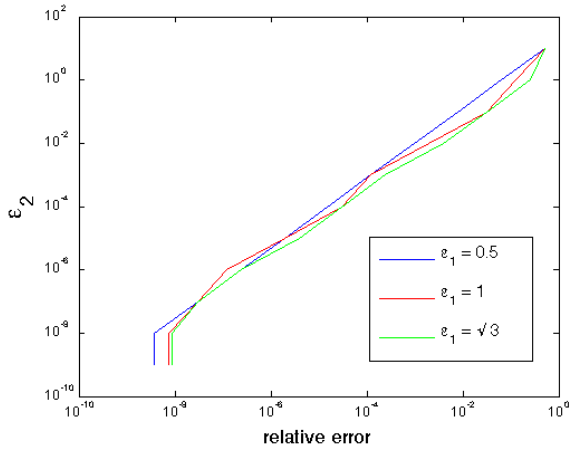

Figure 4.3: ϵ_2 versus relative error

4.4.4 Test of Different Values of the Bound k

Most of the current methods for weighted low-rank approximation are highly dependent on the value of k. In the following, the performance of MROM, DMM as implemented in Algorithm 3, SULS and APM for different values of k.

In [MMH03], one of the advantages of DMM over APM is for low-rank approximation with singular values closely spaced. The first test in this section considers matrices of this type. The data matrix R is chosen as a random 10×10 matrix with chosen singular values $\{1, 1, 1, 1, 1, 0.99, 0.99, 0.99, 0.99, 0.99\}$. The weighting matrix is chosen to be an identity matrix. All algorithms are required to find the best approximation of R with rank $r \leq k = 5$.

The average results of 100 runs with different initial points are reported in Table 4.7. MROM and DMM have similar times and significant time advantages compared with SULS and APM. So MROM is competitive with DMM for such problems.

Table 4.6: Best rank approximation of modified Riemannian optimization method with RTR for different ϵ_1 and ϵ_2. $\epsilon_3 = 10^{-3}, \Delta = 10^{-8}$. The number in the parenthesis indicates the ratio of the rank increases to 27 are 85 out of 100. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

ϵ_2	ϵ_1	rank	R_err	err	SVerr	f	time
10	0.5	1	5.000_{-01}	5.774_{-01}	1.950_{-07}	3.333_{-01}	8.454_{-02}
	1	1	5.000_{-01}	5.774_{-01}	1.950_{-07}	3.333_{-01}	8.436_{-02}
	$\sqrt{3}$	1	5.000_{-01}	5.774_{-01}	1.950_{-07}	3.333_{-01}	8.558_{-01}
1	0.5	4	6.250_{-02}	7.217_{-02}	2.295_{-14}	5.208_{-03}	1.339_{-01}
	1	3	1.250_{-01}	1.443_{-01}	5.633_{-14}	2.083_{-02}	1.160_{-01}
	$\sqrt{3}$	2	2.500_{-01}	2.887_{-01}	7.434_{-13}	8.333_{-01}	1.063_{-01}
10^{-1}	0.5	7	7.813_{-03}	9.021_{-03}	2.645_{-15}	8.138_{-05}	1.737_{-01}
	1	5	3.125_{-02}	3.608_{-02}	2.486_{-15}	1.302_{-03}	1.522_{-01}
	$\sqrt{3}$	5	3.125_{-02}	3.608_{-02}	2.488_{-15}	1.302_{-03}	1.768_{-01}
10^{-2}	0.5	10	9.766_{-04}	1.128_{-03}	2.724_{-15}	1.272_{-06}	2.240_{-01}
	1	9	1.953_{-03}	2.255_{-03}	2.614_{-15}	5.086_{-06}	2.218_{-01}
	$\sqrt{3}$	8	3.906_{-03}	4.511_{-03}	2.618_{-15}	2.035_{-05}	2.539_{-01}
10^{-3}	0.5	13	1.221_{-04}	1.410_{-04}	2.748_{-15}	1.987_{-08}	2.649_{-01}
	1	13	1.221_{-04}	1.410_{-04}	3.228_{-15}	1.987_{-08}	2.979_{-01}
	$\sqrt{3}$	12	2.441_{-04}	2.819_{-04}	3.485_{-15}	7.947_{-08}	3.690_{-01}
10^{-4}	0.5	16	1.526_{-05}	1.762_{-05}	3.196_{-15}	3.104_{-10}	3.102_{-01}
	1	15	3.052_{-05}	3.524_{-05}	3.201_{-15}	1.242_{-09}	3.348_{-01}
	$\sqrt{3}$	15	3.052_{-05}	3.524_{-05}	4.600_{-15}	1.242_{-09}	4.525_{-01}
10^{-5}	0.5	19	1.907_{-06}	2.202_{-06}	3.673_{-15}	4.851_{-12}	4.209_{-01}
	1	19	1.907_{-06}	2.202_{-06}	4.555_{-15}	4.851_{-12}	4.762_{-01}
	$\sqrt{3}$	18	3.815_{-06}	4.405_{-06}	6.433_{-15}	$1.940-11$	6.068_{-01}
10^{-6}	0.5	22	2.384_{-07}	2.753_{-07}	4.506_{-15}	7.579_{-14}	5.238_{-01}
	1	23	1.192_{-07}	1.377_{-07}	5.795_{-15}	1.895_{-14}	6.140_{-01}
	$\sqrt{3}$	22	2.384_{-07}	2.753_{-07}	8.977_{-15}	7.579_{-14}	7.868_{-01}
10^{-7}	0.5	25	2.980_{-08}	3.441_{-08}	4.916_{-15}	1.184_{-15}	6.390_{-01}
	1	25	2.980_{-08}	3.441_{-08}	6.253_{-15}	1.184_{-15}	7.216_{-01}
	$\sqrt{3}$	25	2.980_{-08}	3.441_{-08}	1.126_{-14}	1.184_{-15}	9.448_{-01}
10^{-8}	0.5	28	3.725_{-09}	4.302_{-09}	5.814_{-15}	1.850_{-17}	7.630_{-01}
	1	27	7.451_{-09}	8.603_{-09}	7.246_{-15}	7.401_{-17}	7.707_{-01}
	$\sqrt{3}$	26.85(85/100)	8.568_{-09}	9.894_{-09}	1.207_{-14}	1.073_{-16}	1.021_{+00}
10^{-9}	0.5	28	3.725_{-09}	4.302_{-09}	5.814_{-15}	1.850_{-17}	7.589_{-01}
	1	27	7.451_{-09}	8.603_{-09}	7.246_{-15}	7.401_{-17}	7.673_{-01}
	$\sqrt{3}$	26.85(85/100)	8.568_{-09}	9.894_{-09}	1.207_{-14}	1.073_{-16}	1.019_{+00}
0	0.5	28	3.725_{-09}	4.302_{-09}	5.814_{-15}	1.850_{-17}	7.630_{-01}
	1	27	7.451_{-09}	8.603_{-09}	7.246_{-15}	7.401_{-17}	7.707_{-01}
	$\sqrt{3}$	26.85(85/100)	8.568_{-09}	9.894_{-09}	1.207_{-14}	1.073_{-16}	1.021_{+00}

Table 4.7: Rank 5 approximation of a closely spaced data matrix. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

method	f	R_err	err	t
MROM	4.901_{+00}	7.035_{-01}	2.214_{+00}	6.285_{-02}
DMM	4.901_{+00}	7.035_{-01}	2.214_{+00}	6.029_{-02}
SULS	4.901_{+00}	7.035_{-01}	2.214_{+00}	5.739_{-01}
APM	4.901_{+00}	7.035_{-01}	2.214_{+00}	8.405_{-01}

The next experiment considers 80×10 data matrices with rank 5. The weighting matrices $W = U\Sigma U^T$, where $U \in \mathbb{R}^{mn \times mn}$ is a random orthogonal matrix generated by Matlab's QR and RANDN. The mn singular values of the weighting matrix is generated by Matlab function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution matrix on the interval $[0.5, 1.5]$. Three values of k are considered for each method, one is less than true rank, one equals the true rank and the other is greater than true rank.

Results shown in Table 4.8 are the average of 100 runs for different data matrices R, weighting matrices W and initial points. The data show that when k is chosen less than the true rank, all methods reach almost the same relative error and absolute error, but the time required by MROM is less than the other three. As the value of k increases, MROM shows significant advantages. It achieves good accuracy in the approximation with less computational time. Furthermore, for MROM, as it is iterated based on the three factors U, D, V, the singular values are immediately available while, for the other three methods, an additional SVD is required.

For $k = 7$, the numerical rank of singular values greater than 10^{-8} indicates MROM can obtain the true rank more reliably than the other three methods. Therefore, MROM is more robust finding the true rank with respect to the bound k.

4.4.5 Test of Different Weighting Matrices

Thus far all experiments have considered the weighted low-rank approximation with a full weighting matrix W. In practice, as m and n grow, the complexity of the matrix W must reduce. Two special cases: element-wise weighting, i.e., W is a diagonal, and column-wise weighting, i.e., W is block diagonal with blocks of dimension $m \times m$, are considered.

DMM as implemented in Algorithm 3 does not scale well computationally as m and n grow while k stays relatively small due to the fact that it works in a space of dimension $\min(m-k, n-k)$.

Table 4.8: Approximation of 80-by-10 rank 5 matrices of different k. The number in the parenthesis indicates the ratio of the final rank equals the true rank. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

k	method	rank	f	R_err	err	t
k = 3	MROM	3	8.846_{+01}	3.513_{-01}	3.257_{+01}	4.823_{-01}
	DMM	3	8.846_{+01}	3.513_{-01}	3.257_{+01}	4.706_{+00}
	SULS	3	8.846_{+01}	3.513_{-01}	3.257_{+01}	2.190_{+00}
	APM	3	8.846_{+01}	3.513_{-01}	3.257_{+01}	5.000_{+00}
k = 5	MROM	5	2.191_{-19}	1.557_{-11}	2.304_{-09}	6.890_{-01}
	DMM	5	1.606_{-15}	1.324_{-09}	1.754_{-07}	4.351_{+00}
	SULS	5	2.147_{-12}	4.874_{-08}	9.914_{-06}	1.045_{+00}
	APM	5	7.611_{-09}	2.895_{-06}	4.308_{-04}	3.585_{+00}
k = 7	MROM	5	1.799_{-21}	1.346_{-12}	1.913_{-10}	4.730_{-01}
	DMM	5	1.915_{-18}	4.407_{-11}	7.880_{-09}	2.182_{+00}
	SULS	7(0/100)	1.401_{-12}	3.780_{-08}	1.029_{-05}	2.316_{+00}
	APM	7(0/100)	2.349_{-10}	4.865_{-07}	9.074_{-05}	7.002_{+00}

This is independent of the structure of W. Therefore, DMM is not included in the discussion of these experiments. The algorithm EM-TLS is added to the discussion since it specifically targets block diagonal W.

4.4.5.1 Diagonal Weighting Matrix. The data matrices R are random 80×80 matrices with rank $r = 3, 4, 5$. The weighting matrices W are random diagonal matrices with singular values drawn from a normal distribution having a mean 0 and variance 1. The bound k is set to r. Table 4.9 shows the average results of 50 runs with different data matrices R, weighting matrices W and initial points. The results demonstrate MROM and SULS have significant computational advantages over APM. Furthermore, within almost the same computational time, MROM produces a more accurate approximation than SULS.

Next, the experiments are repeated for data matrices with exponentially decaying singular values. The data matrix R is a random $m \times n$ matrix with rank 10. The chosen singular values are $\{1, 4^{-1}, 4^{-2}, \cdots, 4^{-9}\}$. The weighting matrix W is a diagonal matrix with singular values drawn from a normal distribution having a mean 0 and variance 1. Table 4.10 shows the average results with 100 initial points realizations.

The SULS method is clearly not robust. It cannot reach the stopping criterion for some initial points (the number in the bracket shows the successful runs out of 100). The reported results are

Table 4.9: Approximation of random data matrix with diagonal weighting matrix. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

r	method	f	R_err	err	t
3	MROM	4.800_{-15}	5.410_{-10}	3.842_{-08}	1.196_{-01}
	SULS	1.580_{-13}	7.967_{-09}	5.817_{-07}	1.302_{-01}
	APM	1.954_{-09}	7.640_{-07}	5.619_{-05}	1.935_{+00}
4	MROM	2.334_{-20}	1.992_{-12}	1.912_{-10}	1.335_{-01}
	SULS	3.141_{-13}	8.512_{-09}	8.743_{-07}	1.425_{-01}
	APM	6.136_{-09}	1.059_{-06}	1.033_{-04}	3.088_{+00}
5	MROM	1.924_{-15}	2.814_{-10}	3.475_{-08}	1.499_{-01}
	SULS	5.862_{-13}	9.639_{-09}	1.287_{-06}	1.516_{-01}
	APM	2.545_{-09}	4.499_{-07}	5.579_{-05}	4.722_{+00}

the average of the successful runs. MROM achieves an accurate approximation in less time than the ALS or SULS, especially for large matrices.

4.4.5.2 Block Diagonal Weighting Matrix. The method EW-TLS is specifically designed to efficiently solve the matrix approximation problem with a block diagonal weighting matrix $W = \text{diag}\{W_1, \cdots, W_n\}$. A key part of the success of EW-TLS is the generation of a particular initial condition for the iteration to optimize the transformed cost function. In this section, MROM is compared with EW-TLS and the two general methods SULS and APM.

In the experiments, each $W_i = U_i \Sigma_i U_i^T$ is a positive definite symmetric matrix, where $U_i \in \mathbb{R}^{10\times 10}$ is random orthogonal matrices generated with Matlab's ORTH and RAND, each Σ_i is a diagonal matrix with singular values generated with Matlab's function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution vector on the interval $[0.5, 1.5]$. The data matrix R is a random generated 10×80 matrix with rank 4. The four methods are tested for the values of the upper bound $k = 3, 4, 5$ which are respectively less than, equal to and greater than the true rank. For the EW-TLS approach, the Matlab library function FMINUNC is used to find the local minima. The termination criterion for FMINUNC requires that the norm of final gradient over the norm of initial gradient less than 10^{-7}. For each k and each method, the experiment is repeated 100 times for different initial matrices generated randomly by Matlab's RAND and ORTH. Additionally, each method is run using the custom initial matrix generated by the GTLS approximation [VV89].

Table 4.10: Approximation of data matrix with exponential decay singular values and the weighting matrix is diagonal. $\epsilon_1 = 1, \epsilon_2 = 10^{-6}$. The number in the parenthesis indicates the successful runs out of 100. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

m	n	method	f	R_err	err	t
20	20	MROM	9.935_{-30}	4.370_{-15}	4.343_{-15}	7.628_{-01}
		SULS	8.038_{-16}	3.760_{-08}	1.104_{-07}	9.062_{-01}
		APM	1.596_{-16}	1.568_{-08}	4.503_{-08}	3.038_{-01}
30	20	MROM	3.094_{-29}	7.588_{-15}	7.985_{-15}	6.706_{-01}
		SULS	6.909_{-16}	3.419_{-08}	8.873_{-08}	1.013_{+00}
		APM	5.785_{-17}	8.731_{-09}	2.092_{-08}	6.772_{-01}
30	30	MROM	5.335_{-30}	2.526_{-15}	2.624_{-15}	6.993_{-01}
		SULS	4.804_{-16}	2.937_{-08}	6.683_{-08}	9.606_{-01}
		APM	5.365_{-17}	8.916_{-09}	1.850_{-08}	1.209_{+00}
40	30	MROM	1.591_{-30}	1.325_{-15}	1.320_{-15}	6.320_{-01}
		SULS	5.734_{-16}	3.291_{-08}	7.527_{-08}	8.192_{-01}
		APM	1.314_{-17}	4.426_{-09}	8.535_{-09}	1.950_{+00}
40	40	MROM	4.942_{-30}	1.801_{-15}	1.916_{-15}	8.053_{-01}
		SULS (70/100)	3.537_{-16}	2.492_{-08}	4.948_{-08}	1.147_{+00}
		APM	1.430_{-17}	4.292_{-09}	7.519_{-09}	2.867_{+00}
50	40	MROM	3.388_{-30}	1.890_{-15}	1.946_{-15}	7.021_{-01}
		SULS (93/100)	3.331_{-16}	2.420_{-08}	4.460_{-08}	1.892_{+00}
		APM	7.233_{-18}	3.086_{-09}	4.939_{-09}	4.041_{+00}
50	50	MROM	1.455_{-29}	3.578_{-15}	3.699_{-15}	8.044_{-01}
		SULS (96/100)	2.453_{-16}	2.063_{-08}	3.562_{-08}	1.836_{+00}
		APM	1.260_{-17}	4.131_{-09}	6.318_{-09}	6.159_{+00}
60	50	MROM	3.598_{-29}	8.077_{-15}	8.333_{-15}	7.137_{-01}
		SULS (78/100)	7.917_{-14}	6.493_{-08}	2.738_{-07}	1.852_{+00}
		APM	4.058_{-18}	2.299_{-09}	3.473_{-09}	8.019_{+00}
60	60	MROM	3.117_{-30}	2.043_{-15}	2.121_{-15}	7.504_{-01}
		SULS (93/100)	2.059_{-13}	1.298_{-07}	5.608_{-07}	1.653_{+00}
		APM	5.193_{-18}	2.431_{-09}	3.584_{-09}	1.087_{+01}

Table 4.11: MROM, SULS, EW-TLS and APM for block diagonal weighting matrix W with good initial points and without noise. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

method	k	f	R_err	err	time(s)
MROM	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	1.263_{-01}
	4	3.563_{-29}	9.422_{-16}	2.027_{-14}	1.058_{-02}
	5	7.026_{-29}	1.323_{-15}	2.556_{-14}	9.779_{-03}
SULS	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	8.573_{-01}
	4	3.563_{-29}	9.422_{-16}	2.027_{-14}	2.529_{-02}
	5	4.275_{-29}	1.032_{-15}	2.846_{-14}	2.477_{-02}
EW-TLS	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	8.392_{-01}
	4	3.743_{-26}	3.054_{-14}	1.323_{-12}	3.195_{-02}
	5	1.366_{-22}	1.845_{-12}	1.049_{-10}	3.172_{-02}
APM	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	6.610_{-01}
	4	4.355_{-29}	1.042_{-15}	2.292_{-14}	5.582_{-02}
	5	6.274_{-29}	1.250_{-15}	2.688_{-14}	7.522_{-02}

The average results are presented in Table 4.11 and Table 4.12. From the tables, it is clear that the EW-TLS approach is sensitive to the initial matrix and not only benefits from the use of the GTLS-based initial matrix but often diverges when the GTLS-based initial matrix is not used. Even for the EW-TLS successful runs, for any k, MROM produces as good or better approximations using approximately the same or less computational time and is therefore more robust and as efficient as EW-TLS. Furthermore, since EW-TLS solves the original problem based on the local optimization on a rank-k manifold, it is highly dependent on the upper bound k. If we want to use the algorithm to find the true rank, we need to test different k, which is computationally costly. As with MROM, SULS and APM are not designed specifically for this problem and the consistently produce less accurate approximations and require more computational time than MROM. Clearly, of the four methods MROM using the GTLS-based initial condition is the preferred method.

In practice, a desirable additional property is robustness in the presence of noisy data. The next set of experiments tests the performance of the MROM and EW-TLS in this situation. The data matrices all have 10 rows, with the number of columns n varying from 20 to 200 in increments of 10. Each data matrix R is generated by adding a noise matrix to a rank-4 matrix, i.e., a $10 \times n$ matrix of uniform distributed random numbers with mean 0 and variance 0.0001 is added to the noise-free data matrix R_0 to construct a full rank noisy matrix. The weighting matrix $W = \text{diag}\{W_1, \cdots, W_n\}$. Each $W_i = U_i \Sigma_i U_i^T$, where $U_i \in \mathbb{R}^{10\times 10}$ is a random orthogonal matrix

Table 4.12: MROM, SULS, EW-TLS and APM for block diagonal weighting matrix W with random initial points and without noise. The ratio in the parenthesis indicates the percentage of successful runs. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

method	k	f	R_err	err	time(s)
MROM	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	1.972_{-01}
	4	4.464_{-21}	1.055_{-11}	4.475_{-10}	2.274_{-01}
	5	4.464_{-21}	1.055_{-11}	4.476_{-10}	2.266_{-01}
SULS	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	7.828_{-01}
	4	2.002_{-14}	1.569_{-08}	1.037_{-06}	1.055_{+00}
	5	1.233_{-13}	5.516_{-08}	4.775_{-06}	1.495_{+00}
EW-TLS	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	$7.299_{-01}(84/100)$
	4	6.209_{-12}	2.043_{-07}	5.586_{-06}	$1.181_{+00}(48/100)$
	5	4.899_{-05}	2.430_{-04}	7.168_{-03}	$1.577_{+00}(47/100)$
APM	3	2.931_{-01}	8.545_{-02}	2.534_{+00}	6.208_{-01}
	4	5.295_{-08}	2.979_{-05}	9.242_{-04}	6.641_{-01}
	5	5.402_{-08}	3.101_{-05}	1.021_{-03}	9.447_{-01}

generated by Matlab's QR and RAND. The 10 singular values of W_i are generated by Matlab function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution matrix on the interval $[0.5, 1.5]$. The upper bound on the rank is $k = 4$. The relative error $\frac{\|R_0 - X_*\|_W}{\|R_0\|_W}$ and absolute error $\|R_0 - X_*\|_F$ are presented in Table 4.13. The data are the average execution times over 100 runs for different noise realizations.

The data show MROM to be more robust with respect to the choice of initial approximation. The EW-TLS data all use the good GTLS-based initial approximation matrix with true rank (i.e. rank-4) but a significant number of the runs do not satisfy the stopping criterion while MROM converges for all problems for any initial approximation (i.e. rank-1,2,3,4). The errors are comparable for MROM with different rank initial approximations and EW-TLS when it converges. In general, MROM with the true rank good initial condition results in the best error and computational time combination. Given that the computational complexity of producing the GTLS-based good initial condition is not significant, MROM using it is the robust and efficient method of choice. Its advantage in efficiency is seen, in particular, for the data sets of size $10 \times n$ with $10 << n$.

Table 4.13: GTLS-based initial points. The ratio in the parenthesis indicates the percentage of successful runs. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

				MROM with different rank initial				
m	n	R_err	err	rank-1	rank-2	rank-3	rank-4	EW-TLS
10	20	6.037_{-05}	8.416_{-04}	1.834_{-01}	1.648_{-01}	1.116_{-01}	8.821_{-02}	4.193_{-01}(95/100)
10	30	6.115_{-05}	1.020_{-03}	1.930_{-01}	1.466_{-01}	1.564_{-01}	9.448_{-02}	5.697_{-01}(96/100)
10	40	6.199_{-05}	1.209_{-03}	1.809_{-01}	1.808_{-01}	1.865_{-01}	1.023_{-01}	7.270_{-01}(97/100)
10	50	6.043_{-05}	1.324_{-03}	2.059_{-01}	1.798_{-01}	1.491_{-01}	1.101_{-01}	9.351_{-01}(96/100)
10	60	5.985_{-05}	1.456_{-03}	1.673_{-01}	1.398_{-01}	1.416_{-01}	1.206_{-01}	1.058_{+00}(98/100)
10	70	5.842_{-05}	1.577_{-03}	2.411_{-01}	1.615_{-01}	2.162_{-01}	1.352_{-01}	1.220_{+00}(98/100)
10	80	6.007_{-05}	1.685_{-03}	2.077_{-01}	2.055_{-01}	1.726_{-01}	1.469_{-01}	1.416_{+00}(98/100)
10	90	5.943_{-05}	1.801_{-03}	2.620_{-01}	2.505_{-01}	2.703_{-01}	1.594_{-01}	1.508_{+00}(94/100)
10	100	5.919_{-05}	1.900_{-03}	1.857_{-01}	2.470_{-01}	2.017_{-01}	1.513_{-01}	1.597_{+00}(97/100)
10	110	5.830_{-05}	1.975_{-03}	2.072_{-01}	2.783_{-01}	2.565_{-01}	1.726_{-01}	1.860_{+00}(96/100)
10	120	5.900_{-05}	2.066_{-03}	2.271_{-01}	2.283_{-01}	2.089_{-01}	1.562_{-01}	1.910_{+00}(96/100)
10	130	5.855_{-05}	2.176_{-03}	2.817_{-01}	2.403_{-01}	2.418_{-01}	1.780_{-01}	2.156_{+00}(97/100)
10	140	6.021_{-05}	2.227_{-03}	2.463_{-01}	2.735_{-01}	2.382_{-01}	1.814_{-01}	2.276_{+00}(95/100)
10	150	6.022_{-05}	2.343_{-03}	2.672_{-01}	2.677_{-01}	2.752_{-01}	2.008_{-01}	2.391_{+00}(94/100)
10	160	6.075_{-05}	2.389_{-03}	3.120_{-01}	2.781_{-01}	2.726_{-01}	2.051_{-01}	2.518_{+00}(92/100)
10	170	6.072_{-05}	2.462_{-03}	3.166_{-01}	2.957_{-01}	3.622_{-01}	2.200_{-01}	2.660_{+00}(97/100)
10	180	6.026_{-05}	2.545_{-03}	2.723_{-01}	2.760_{-01}	3.737_{-01}	2.125_{-01}	2.763_{+00}(98/100)
10	190	6.090_{-05}	2.611_{-03}	3.330_{-01}	2.855_{-01}	2.966_{-01}	2.320_{-01}	3.050_{+00}(96/100)
10	200	6.021_{-05}	2.680_{-03}	4.200_{-01}	3.093_{-01}	3.869_{-01}	2.489_{-01}	3.056_{+00}(95/100)

4.4.6 Choice of Retraction and Performance

Four types of retractions on fixed-rank manifolds have been proposed for consideration along with the rank-related retractions: SVD-type retraction (4.15), (4.38), polar-decomposition-type (PD-type) retraction (4.16), (4.33) and QR-type retraction I (4.17), (4.39) and QR-type retraction II (4.18), (4.41). As noted earlier, the last two types are not invariant to factors in the decomposition $X = UDV^T$. They are included in the experiments to produce initial evidence as to whether or not this lack of invariance is important to the effectiveness of the algorithm. Each data matrix R is a random $m \times n$ matrix with rank 4. Each weighting matrix is $W = U\Sigma U^T$, where $U \in \mathbb{R}^{mn \times mn}$ is a random orthogonal matrix generated by Matlab's QR and RAND. The mn singular values of each weighting matrix are generated by Matlab function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution matrix on the interval $[0.5, 1.5]$. The upper bound on rank, k, is 4.

Table 4.14: Rank-4 approximation by different retractions. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

m	n	Retraction	f	R_err	err	time(s)	gf_r	gf_r/gf_F0	nf	ng	nH	nR	nR_r
6	6	SVD-type	8.213_{-17}	2.754_{-10}	2.475_{-08}	1.910_{-01}	2.525_{-09}	1.619_{-09}	28.080	28.080	222.220	21.320	1.820
		PD-type	1.042_{-17}	5.921_{-11}	5.032_{-09}	2.028_{-01}	1.259_{-09}	8.556_{-10}	38.300	38.300	245.300	31.640	1.820
		QR-type I	6.819_{-18}	2.405_{-16}	2.830_{-09}	2.159_{-01}	9.323_{-10}	4.885_{-10}	38.540	38.540	246.440	31.780	1.820
		QR-type II	1.922_{-17}	9.902_{-12}	8.258_{-09}	2.288_{-01}	1.490_{-09}	9.129_{-10}	39.520	39.520	254.160	32.780	1.820
6	12	SVD-type	3.683_{-17}	6.988_{-12}	1.423_{-08}	2.028_{-01}	2.090_{-09}	8.494_{-10}	23.180	23.180	204.860	17.940	1.200
		PD-type	2.619_{-17}	7.210_{-13}	7.045_{-09}	2.113_{-01}	1.847_{-09}	6.716_{-10}	31.260	31.260	216.760	26.000	1.200
		QR-type I	2.669_{-17}	8.605_{-14}	7.946_{-09}	2.224_{-01}	2.152_{-09}	8.224_{-10}	31.500	31.500	216.340	26.280	1.200
		QR-type II	3.820_{-17}	9.028_{-11}	1.288_{-08}	2.312_{-01}	3.116_{-09}	1.311_{-09}	32.120	32.120	215.200	26.920	1.180
12	12	SVD-type	1.513_{-16}	5.755_{-11}	2.201_{-08}	1.813_{-01}	3.802_{-09}	1.038_{-09}	20.400	20.400	155.120	15.800	1.000
		PD-type	5.152_{-17}	8.057_{-12}	9.493_{-09}	1.963_{-01}	2.717_{-09}	7.942_{-10}	26.560	26.560	167.780	22.040	1.000
		QR-type I	8.736_{-17}	2.227_{-10}	1.507_{-08}	1.925_{-01}	3.540_{-09}	1.007_{-09}	26.760	26.760	166.260	22.320	1.000
		QR-type II	2.782_{-17}	3.152_{-14}	6.318_{-09}	2.043_{-01}	2.038_{-09}	6.281_{-10}	28.400	28.400	173.980	23.740	1.000
12	24	SVD-type	9.087_{-17}	4.276_{-15}	1.337_{-08}	1.682_{-01}	2.615_{-09}	5.577_{-10}	19.540	19.540	134.140	15.120	1.000
		PD-type	1.348_{-16}	1.728_{-11}	1.721_{-08}	1.838_{-01}	4.547_{-09}	9.539_{-10}	23.960	23.960	148.300	19.540	1.000
		QR-type I	1.147_{-16}	1.476_{-11}	1.332_{-08}	1.923_{-01}	3.788_{-09}	7.290_{-10}	24.000	24.000	147.700	19.580	1.000
		QR-type II	1.306_{-16}	4.983_{-11}	1.973_{-08}	1.966_{-01}	6.156_{-09}	1.255_{-09}	25.660	25.660	149.280	21.300	1.000
24	24	SVD-type	2.222_{-16}	1.233_{-16}	1.557_{-08}	2.466_{-01}	2.770_{-09}	4.169_{-10}	19.560	19.560	116.980	14.860	1.000
		PD-type	7.721_{-16}	2.799_{-10}	4.576_{-08}	2.458_{-01}	1.209_{-08}	1.664_{-09}	22.460	22.460	119.920	18.060	1.000
		QR-type I	6.310_{-16}	1.726_{-10}	3.880_{-08}	2.632_{-01}	1.026_{-08}	1.338_{-09}	22.860	22.860	121.780	18.420	1.000
		QR-type II	2.465_{-16}	1.470_{-16}	1.614_{-08}	2.810_{-01}	5.159_{-09}	7.350_{-10}	25.080	25.080	128.660	20.680	1.000
24	48	SVD-type	2.672_{-16}	5.506_{-11}	1.449_{-08}	4.603_{-01}	3.851_{-09}	3.942_{-10}	19.440	19.440	99.140	15.320	1.000
		PD-type	4.100_{-16}	1.587_{-10}	2.167_{-08}	4.688_{-01}	5.993_{-09}	5.854_{-10}	21.940	21.940	102.600	17.940	1.000
		QR-type I	3.788_{-16}	1.453_{-10}	2.081_{-08}	4.812_{-01}	5.701_{-09}	5.576_{-10}	21.960	21.960	102.900	17.960	1.000
		QR-type II	4.076_{-16}	1.314_{-16}	2.713_{-08}	4.948_{-01}	8.174_{-09}	8.187_{-10}	22.120	22.120	105.400	17.920	1.000
48	48	SVD-type	6.711_{-19}	4.341_{-12}	9.230_{-10}	1.175_{+00}	2.740_{-10}	1.974_{-11}	20.220	20.220	88.280	16.200	1.000
		PD-type	2.278_{-15}	2.510_{-10}	8.800_{-08}	1.129_{+00}	2.675_{-08}	1.871_{-09}	21.620	21.620	83.080	17.580	1.000
		QR-type I	2.735_{-15}	2.511_{-10}	9.634_{-08}	1.147_{+00}	2.926_{-08}	2.034_{-09}	22.120	22.120	84.000	18.100	1.000
		QR-type II	3.216_{-15}	2.245_{-10}	1.007_{-07}	1.177_{+00}	3.204_{-08}	2.232_{-09}	22.100	22.100	86.360	18.000	1.000
48	96	SVD-type	2.686_{-16}	1.203_{-12}	2.019_{-08}	3.709_{+00}	6.660_{-09}	3.337_{-10}	20.860	20.860	79.180	16.860	1.000
		PD-type	5.878_{-15}	1.070_{-16}	1.094_{-07}	4.202_{+00}	3.343_{-08}	1.626_{-09}	23.560	23.560	89.520	19.140	1.000
		QR-type I	6.809_{-15}	1.055_{-16}	1.182_{-07}	4.168_{+00}	3.628_{-08}	1.764_{-09}	23.560	23.560	89.500	19.140	1.000
		QR-type II	7.385_{-15}	6.091_{-15}	1.219_{-07}	4.204_{+00}	4.022_{-08}	1.963_{-09}	23.800	23.800	89.800	19.480	1.000
96	96	SVD-type	3.367_{-15}	5.028_{-10}	3.720_{-08}	1.485_{+01}	1.358_{-08}	4.568_{-10}	22.130	22.130	81.522	17.870	1.000
		PD-type	5.468_{-18}	3.240_{-13}	2.606_{-09}	1.509_{+01}	9.273_{-10}	3.246_{-11}	23.826	23.826	81.565	19.826	1.000
		QR-type I	5.740_{-18}	3.189_{-13}	2.647_{-09}	1.509_{+01}	9.430_{-10}	3.299_{-11}	23.826	23.826	81.565	19.826	1.000
		QR-type II	3.281_{-16}	1.318_{-10}	1.793_{-08}	1.537_{+01}	6.435_{-09}	2.218_{-10}	24.478	24.478	82.652	20.478	1.000

Table 4.14 shows the average results of 100 runs for each retraction with different data and weighting matrix realizations. All find an approximation with rank the same as the true rank of 4. The final cost function value, relative error and error are all very small. Note that the two QR-type retractions without guaranteed invariance also work well in terms of quality of approximation and computational time.

4.4.7 Performances of Different Rank Reduction Methods

The usual way to reduce the rank is to compute the SVD decomposition of a matrix, set the smallest singular values to zero. This method has been widely used when considering the low-rank

approximation. However, when the sizes of the matrices get large, the computation of the SVD is impractical. We employ the three-factor representation, which avoids the computation. Therefore, the rank reduction can be realized through the truncation of the smallest singular values. However, if we start from a rank-1 matrix, in the process of estimation, the rank might be increased first, then reduce to the true rank. The information obtained in the rank increment can be used to make the rank reduction more efficient as proposed in Algorithm 2.

Table 4.15 shows the average results of rank reduction by truncation compared with the way using the rank increment information. The data matrices R are random $m \times n$ matrices with rank 5. The weighting matrices W are block diagonal matrices. Each block is a positive definite symmetric matrix. The upper bound k is set to be m and the initial starting point is a random generated rank-1matrix. From the table, it is clear that using the rank increment information for rank reduction is more efficient than the simple truncation, especially when the sizes get larger.

Table 4.15: Rank-5 approximation by different rank reduction methods. The subscript $\pm k$ indicates a scale of $10^{\pm k}$.

m	n	Rank Reduce Method	R_err	err	f	time(s)	gf_r	gf_r/gf_F0
10	10	Truncation-type	2.213_{-15}	1.163_{-07}	2.528_{-15}	2.741_{-01}	1.710_{-08}	8.850_{-10}
		New-type	7.833_{-13}	1.104_{-07}	1.673_{-15}	2.712_{-01}	1.606_{-08}	8.783_{-10}
20	20	Truncation-type	8.864_{-12}	3.625_{-07}	1.552_{-14}	3.139_{-01}	4.286_{-08}	1.297_{-09}
		New-type	6.125_{-12}	3.044_{-07}	1.133_{-14}	3.044_{-01}	3.656_{-08}	1.068_{-09}
30	30	Truncation-type	1.500_{-10}	4.053_{-07}	2.402_{-14}	3.131_{-01}	5.231_{-08}	9.753_{-10}
		New-type	1.875_{-11}	5.491_{-07}	3.921_{-14}	3.093_{-01}	6.895_{-08}	1.393_{-09}
40	40	Truncation-type	3.025_{-16}	5.678_{-07}	4.850_{-14}	7.164_{-01}	9.158_{-08}	1.370_{-09}
		New-type	6.121_{-11}	4.909_{-07}	3.428_{-14}	5.022_{-01}	4.136_{-08}	6.576_{-10}
50	50	Truncation-type	1.654_{-13}	2.059_{-07}	1.733_{-14}	1.055_{+00}	1.992_{-08}	2.311_{-10}
		New-type	1.928_{-15}	4.034_{-07}	2.369_{-14}	7.725_{-01}	4.089_{-08}	5.225_{-10}
60	60	Truncation-type	1.060_{-11}	4.717_{-07}	3.790_{-14}	1.746_{+00}	4.204_{-08}	4.204_{-10}
		New-type	4.061_{-17}	5.622_{-07}	4.602_{-14}	1.070_{+00}	7.125_{-08}	8.000_{-10}

4.4.8 Performances of Other General Riemannian Optimization Algorithms

As mentioned earlier, any other optimization methods can be used as the inner algorithm in MROM. In this section, performances of six Riemannian optimization algorithms are illustrated. The six algorithms are Riemannian steepest descent with line search (RTR-SD), Riemannian trust region with symmetric rank-one update (RTR-SR1), Limited-memory RTR-SR1 (LRT-SR1),

general Riemmanian trust-region method (RTR-Newton), Riemannian Broyden-Fletcher-Goldfarb-Shannon (RBFGS) and limited-memory RBFGS (LRBFGS).

Each data matrix R is a random $m \times n$ matrix with rank 4. Each weighting matrix is $W = U\Sigma U^T$, where $U \in \mathbb{R}^{mn \times mn}$ is a random orthogonal matrix generated by Matlab's QR and RAND. The mn singular values of each weighting matrix are generated by Matlab function LOGSPACE with condition number 100 and multiplying, element-wise, by a uniform distribution matrix on the interval $[0.5, 1.5]$. The upper bound on rank, k, is m.

Figure 4.4 shows the average computational time of 5 runs for each method with different data and weighting matrix realizations. All find an approximation with rank the same as the true rank of 4. The figure shows RTR-Newton has time advantages when the sizes of matrices, i.e. $m \times n$, are not too large. As the sizes increase, limited-memory RTR-SR1 and RBFGS show significant time advantages compared with the other methods. Therefore, we can choose different inner algorithm based on the size of the matrices and the efficiency required in specific applications.

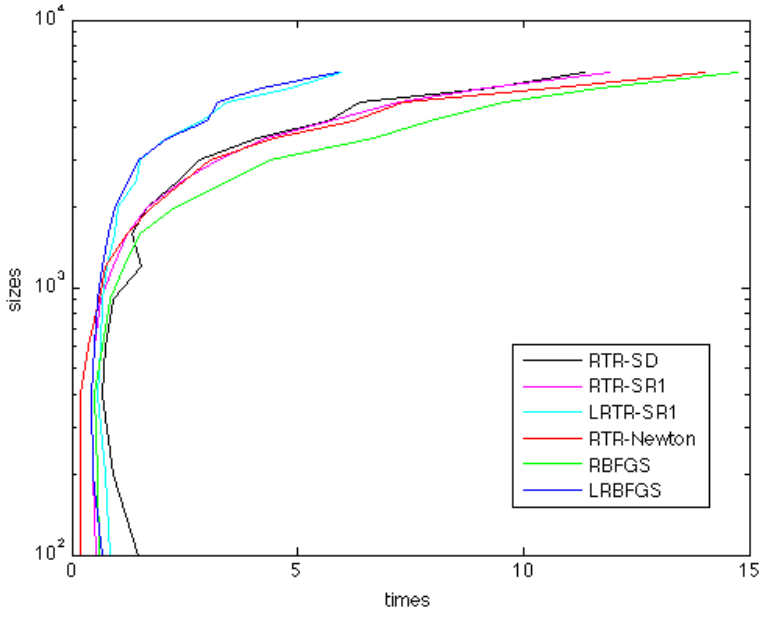

Figure 4.4: Average computational time versus the size of matrix for each method.

4.5 Conclusion

The modified Riemannian optimization algorithm with RTR on the weighted low-rank approximation problem has been explored in this section. First, the effect of the parameters ϵ_1 and ϵ_2 on rank estimation for problems with the difficult exponential decay of singular values was tested. MROM was demonstrated to be effective in computing the appropriate numerical rank approximation of the data matrix given the values of ϵ_1 and ϵ_2 even when the upper bound on the rank, k, was taken to be large and therefore potentially allowing an approximation that was unnecessarily inefficient in terms of space and computational time. For data matrices with a clear gap in their singular values, the algorithm demonstrated a high probability of finding the true rank independently of initial conditions. For the practical situation of a structured weighting matrix and data with and without noise, MROM consistently outperformed, in terms of computational time and approximation quality, the general methods SULS and APM as well as the EW-TLS method that is specifically designed for such problems. Finally, the performance of MROM was seen to be consistent across all choices of the fixed-rank and rank-related retractions independently of the invariance of those retractions with respect to the particular factors in the decomposition of the matrix X. This is for moderate problem sizes and RTR as the inner fixed-rank algorithm. For large problems, limited-memory Riemannian optimization methods as the inner fixed-rank algorithm demonstrated significant time advantages.

CHAPTER 5

LOW-RANK APPROXIMATION ON GRAPH SIMILARITY MATRIX

The node-to-node similarity measure introduced by Blondel et al. [BGH$^+$04] has been used in many practical problems. They defined the similarity matrix as a fixed point of an iterative process, and prove that their measure is equivalent to the solution of an eigenvalue problem of a dimension that is the product of the number of nodes in the two graphs. In this chapter, the efficient determination of a low-rank approximation approximation of the similarity matrix is considered. Section 5.1 reviews the similarity measure of Blondel et al. Two low-rank approximations of the similarity matrix introduced by Cason et al. [CAD13] are discussed in Section 5.2. Some observations are made and new efficient methods of the low-rank approximations of the similarity matrix based on the Riemannian approach to rank inequality constraints are derived in Section 5.3. In Section 5.4, Riemannian optimization methods reviewed in Chapter 2 are compared with Cason's iteration method for low-rank approximation with k identical singular values. Finally, comparisons between the new rank-related algorithm and Cason's iteration method on low-rank approximation with rank at most k are presented in Section 5.5 and the results are summarized in Section 5.6.

5.1 The Similarity Measure of Blondel et al.

In [BGH$^+$04], the node-to-node similarity measure considers two nodes of different graphs "similar" if their neighboring nodes are "similar". For example, the similarity score between node 2 of G_A in Figure 5.1 and node 4 of G_B in Figure 5.2 is determined by the similarity score between their neighbors:

$$s(a_2, b_4) \leftarrow s(a_1, b_1) + s(a_1, b_3) + s(a_3, b_5).$$

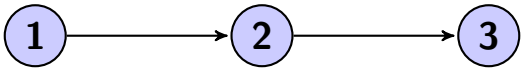

Figure 5.1: Graph G_A with three nodes.

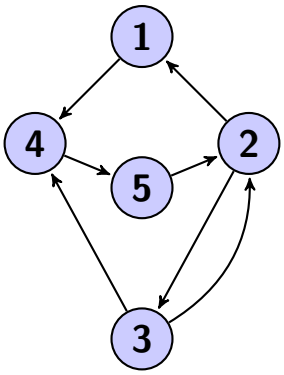

Figure 5.2: Graph G_B with five nodes

In general, given two arbitrary graphs G_A and G_B with n_A and n_B vertices and edge sets E_A and E_B, the similarity score between node i in G_A and node j in G_B is updated according to the following equation:

$$S_{ij} \leftarrow \sum_{r:(r,i)\in E_B, s:(s,j)\in E_A} S_{rs} + \sum_{r:(i,r)\in E_B, s:(j,s)\in E_A} S_{rs}.$$

This can be written in a more compact matrix form

$$S_{k+1} \leftarrow AS_kB^T + A^TS_kB := \mathcal{M}(S_k), \quad k = 0, 1, \cdots \tag{5.1}$$

where A and B are the adjacency matrices of G_A and G_B, and S_k is the $n_A \times n_B$ matrix of entries S_{ij} at iteration k.

Note that the updating function $\mathcal{M}(S_k)$ is a linear map on the matrix S_k. This can be explicitly seen by applying the vec$\{\cdot\}$ operation to the above equation, which concatenates the columns of a matrix into one column vector, to obtain

$$\text{vec}\{S_{k+1}\} \leftarrow (B \otimes A + B^T \otimes A^T)\text{vec}\{S_k\} :\equiv M\text{vec}\{S_k\}. \tag{5.2}$$

Since only the relative score of each pair of nodes is of interest, not the value of S_{ij}, the entire similarity matrix S is normalized using

$$\text{vec}\{S_{k+1}\} = \frac{M\text{vec}\{S_k\}}{\|M\text{vec}\{S_k\}\|_2}, \quad k = 0, 1, \cdots . \tag{5.3}$$

This also avoids over- or under-flow.

The matrix $M := (B \otimes A + B^T \otimes A^T)$ is symmetric and non-negative, and therefore the non-negative vector vec$\{S\}$ is a Perron vector of M, corresponding to the Perron root (i.e. the spectral

radius) $\rho = \max_{\lambda x = Mx} |\lambda|$. Since M is symmetric, its eigenvalues are real and hence it can have only two extremal eigenvalues: ρ and possibly $-\rho$. M^2 is also non-negative and its extremal eigenvalue is ρ^2, which is unique but its geometric multiplicity can be larger than 1. Let Π be the orthogonal projector onto the space of eigenvectors of M^2 with eigenvalues ρ^2, then Π is a non-negative map and any vector $\text{vec}\{S\} = \Pi\text{vec}\{S_0\}$, with $\text{vec}\{S_0\}$ non-negative, is a non-negative solution of $\rho^2\text{vec}\{S\} = M^2\text{vec}\{S\}$.

It was shown in [BGH$^+$04] that the even iterates of the following recurrence

$$\text{vec}\{S_0\} = \mathbf{1}_{mn}, \quad \text{vec}\{S_{k+1}\} = \frac{M\text{vec}\{S_k\}}{\|M\text{vec}\{S_k\}\|_2}, \quad k = 0, 1, \cdots, \tag{5.4}$$

converge to the unique non-negative vector with the largest possible 1-norm (the sum of the magnitudes of all entries), where $\mathbf{1}_{mn}$ denotes a vector whose entries are all equal to 1. Therefore, the definition of the similarity matrix S is the non-negative solution corresponding to $S_0 = \mathbf{1}_{m,n}$ of the following system:

$$\rho^2 S = \mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S)), \quad \mathcal{M}(S) := BSA^T + B^T SA. \tag{5.5}$$

Two additional properties of the matrix to which iteration (5.4) converges are also presented in [BGH$^+$04]:

- The self-similarity matrix of a path graph (a undirected tree with two leaves and internal nodes all with a node degree of 2) is a diagonal matrix.
- When either of the graphs G_A or G_B is regular (a graph is *regular* if the in-degrees of all vertices are equal and the out-degrees of all vertices are equal) or has a normal adjacency matrix (a matrix A is *normal* if it satisfies $AA^T = A^TA$), then the similarity matrix S has rank 1.

The cyclic definition very naturally leads to iterative updates, in which similarity scores between elements propagate along edges to neighboring elements on each iteration.

Algorithm 5 Blondel's Algorithm

Require: Graph G_A and G_B respectively of order m and n

1: $S^0 \leftarrow \mathbf{1}/\|\mathbf{1}\|_F \in \mathbb{R}^{m\times n}$
2: **for** $t = 1, 2, \cdots, t_{max}$ **do**
3: $\quad S^t \leftarrow \frac{AS^{t-1}B^T + A^TS^{t-1}B}{\|AS^{t-1}B^T + A^TS^{t-1}B\|_F}$
4: **end for**
5: $\mathbf{S} \leftarrow S^t$
6: where t_{max} is an even number that is "sufficiently large".

The application of this similarity scoring method to the graphs in Figures 5.3 and 5.4 results in the similarity score shown in Table 5.1.

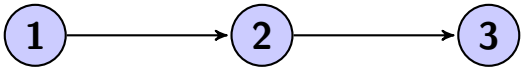

Figure 5.3: Graph G_A with three nodes.

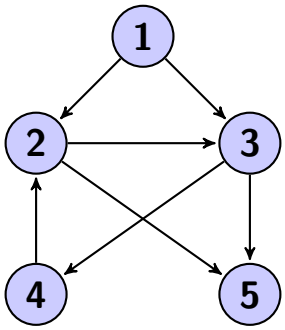

Figure 5.4: Graph G_B with five nodes.

Table 5.1: Similarity Scores between G_A and G_B.

Nodes	1	2	3
1	0.443	0.104	0
2	0.280	0.396	0.086
3	0.086	0.396	0.280
4	0.222	0.049	0.222
5	0	0.104	0.443

5.2 Low-rank Approximation of Similarity Matrix by Cason et al.

As the size of the graphs increases, Algorithm 5 becomes computationally expensive. To save storage space and computation time, a low-rank matrix is considered to approximate the similarity matrix. The low-rank approximation is known to be reasonable for some cases. For example, we know the similarity matrix defined in [BGH$^+$04] can have low-rank structure, see details in Section 5.1. Furthermore, in the expriments, low-rank structure is also observed when considering the similarity between a noisy graph and a given graph, i.e., the similarity between a graph G and $G + \Delta G$, where ΔG represents "noise" in edge weights (which includes adding edges by changing weights with value 0) added to the graph G. It is still an open question as to whether or not a low-rank approximation of a similarity matrix that does not have exact or numerical low-rank contains any useful information.

In [CAD13], Cason et al. proposed two low-rank iterative schemes that converge to two approximations of the Blondel et al. similarity matrix with respectively either k nonzero identical singular values or at most k nonzero (not necessarily identical) singular values. In this section, these methods are reviewed.

Cason et al. first show that the similarity matrix defined by Blondel et al. is the solution of an optimization problem. The iteration in Algorithm 5 is such that

$$S^t \in \underset{\|S\|_F=1}{\operatorname{argmax}} \langle S, \mathcal{M}(S^{t-1})\rangle_F = \operatorname{tr}(S^T \mathcal{M}(S^{t-1})). \tag{5.6}$$

Moreover, they prove that $S^{2\infty}$ is a solution of

$$\max_{S\in\mathcal{S}(m,n)} \Phi(S), \quad \Phi(S) := \langle S, \mathcal{M}^2(S)\rangle_F = \operatorname{tr}(S^T \mathcal{M}^2(S)), \tag{5.7}$$

where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ and $[\mathcal{M}(S)]_{ij} = [ASB^T + A^T SB]_{ij}$, $\mathcal{S}(m,n) := \operatorname{Norm}(1,m,n) = \{S \in \mathbb{R}^{m\times n} : \|S\|_F = 1\}$. This problem maximizes a continous function Φ on a compact domain $\mathcal{S}(m,n)$. Hence, according to the first order optimality condition, if $S^{2\infty}$ is a maximizer, then $S^{2\infty}$ is a stationary point of the iteration. They then proposed the following two low-rank approximations and gave two iterative algorithms.

Approximation with k Identical Singular Values:. In this case, they replace the set $\mathcal{S}(m,n)$ by $\mathcal{S}_k(m,n)$, which is the set of rank-k matrices with Frobenius norm 1 with k identical singular values, i.e.

$$\mathcal{S}_k(m,n) = \left\{ \begin{array}{r} U\hat{I}_k V^T \in \mathbb{R}^{m\times n} : U \in \operatorname{St}(m,k), V \in \operatorname{St}(n,k), \\ \hat{I}_k = I_k/\|I_k\|_F = I_k/\sqrt{k} \end{array} \right\}. \tag{5.8}$$

where $\operatorname{St}(m,k)$ is Stiefel manifold which denotes the set of all $m \times k$ orthonormal matrices, i.e.

$$\operatorname{St}(m,k) := \{X \in \mathbb{R}^{m\times k} : X^T X = I_k\}, \tag{5.9}$$

and I_k denotes the $k \times k$ identity matrix. They propose an iteration algorithm, Algorithm 6, to find an approximation of the similarity matrix defined by Blondel et al. and prove that it converges to a stationary point of maximization problem (5.7) .

Algorithm 6 Cason's Algorithm 1

Require: Graph G_A and G_B respectively of order m and n

1: $S^0 \leftarrow \mathbf{1}/\|\mathbf{1}\|_F \in \mathbb{R}^{m\times n}$
2: **for** $t = 1, 2, \cdots, t_{max}$ **do**
3: Compute $S^t \in \mathcal{S}_k(m,n)$ according to
4: $S^t (= U^t \hat{I}_k [V^t]^T) \leftarrow f(S^{t-1}) := \text{argmax}_{\tilde{S}\in\mathcal{S}_k(m,n)} \langle \tilde{S}, \mathcal{M}^2(S^{t-1})\rangle_F$
5: **end for**
6: $\mathbf{S} \leftarrow S^t$
7: where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ and $[\mathcal{M}(S)]_{ij} = [ASB^T + A^T SB]_{ij}$

Approximation of rank at most k:. The second method they propose is replacing the set $\mathcal{S}(m,n)$ by $\mathcal{S}_{\leq k}(m,n)$, the set of matrices of norm 1 with rank at most k, i.e.

$$\mathcal{S}_{\leq k}(m,n) = \left\{ \begin{array}{c} UDV^T \in \mathbb{R}^{m\times n} : U \in \text{St}(m,k), V \in \text{St}(n,k), \\ D \text{ is a diagonal matrix}, \|D\|_F = 1 \end{array} \right\}. \tag{5.10}$$

They give another iteration algorithm, Algorithm 7, to find an approximation of the similarity matrix defined by Blondel et al. and they prove that it also converges to a stationary point of the maximization problem (5.7).

Algorithm 7 Cason's Algorithm 2

Require: Graph G_A and G_B respectively of order m and n

1: $S^0 \leftarrow \mathbf{1}/\|\mathbf{1}\|_F \in \mathbb{R}^{m\times n}$
2: **for** $t = 1, 2, \cdots, t_{max}$ **do**
3: Compute $S^t \in \mathcal{S}_{\leq k}(m,n)$ according to
4: $S^t (= U^t D^t [V^t]^T) \leftarrow f(S^{t-1}) := \text{argmax}_{\tilde{S}\in\mathcal{S}_{\leq k}(m,n)} \langle \tilde{S}, \mathcal{M}^2(S^{t-1})\rangle_F$
5: **end for**
6: $\mathbf{S} \leftarrow S^t$
7: where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ and $[\mathcal{M}(S)]_{ij} = [ASB^T + A^T SB]_{ij}$

Note that at every iteration Algorithm 7, is exactly the same as Algorithm 5, except taking only the first k dominant singular values, assuming $k < \min(m,n)$. The case when they are equivalent is easy to characterize.

Proposition 29. *If $k = \min(m,n)$, then Algorithm 7 is equivalent to Algorithm 5.*

Proof. Let $\mathcal{M}^2(S)$ have an ordered singular value decomposition

$$\mathcal{M}^2(S) = P\Sigma Q^T = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}, \tag{5.11}$$

with $P_1 \in \mathbb{R}^{m\times k}, P_2 \in \mathbb{R}^{m\times(m-k)}, Q_1 \in \mathbb{R}^{n\times k}, Q_2 \in \mathbb{R}^{n\times(n-k)}, \Sigma_1 \in \mathbb{R}^{k\times k}$ and $\Sigma_2 \in \mathbb{R}^{(m-k)\times(n-k)}$. Then the next iteration is determined by

$$S^t (= U^t D^t [V^t]^T) \leftarrow \Phi(S^{t-1}) := \underset{\tilde{S}\in\mathcal{S}_{\leq k}(m,n)}{\operatorname{argmax}} \langle \tilde{S}, \mathcal{M}^2(S^{t-1})\rangle_F \tag{5.12}$$

and

$$\begin{aligned} \langle \tilde{S}, \mathcal{M}^2(S)\rangle_F &= \operatorname{tr}(\tilde{S}^T \mathcal{M}^2(S)) = \operatorname{tr}(VDU^T \mathcal{M}^2(S)) = \operatorname{tr}(DU^T \mathcal{M}^2(S)V) \\ &\leq \sum_{i=1}^{k} \sigma_i(DU^T \mathcal{M}^2(S)V) \leq \sum_{i=1}^{k} \sigma_i(D)\sigma(U^T \mathcal{M}^2(S)V) \leq \sum_{i=1}^{k} \sigma_i(D)\sigma_i(\Sigma_1) \\ &\leq \operatorname{tr}(\hat{\Sigma}_1 \Sigma_1) \end{aligned} \tag{5.13}$$

where $\hat{\Sigma}_1 := \frac{\Sigma_1}{\|\Sigma_1\|_F}$. Thus,

$$S^t = \frac{1}{\|\Sigma_1\|_F} P_1 \Sigma_1 Q_1^T. \tag{5.14}$$

If $k = \min(m, n)$, without loss of generality, let us assume $m < n$, then $P_1 = P, \Sigma_1 = \Sigma$ and $Q_1 = Q$,

$$S^t = \frac{1}{\|\Sigma\|_F} P\Sigma Q^T = \frac{P\Sigma Q^T}{\|P\Sigma Q^T\|_F} = \frac{\mathcal{M}^2(S)}{\|\mathcal{M}^2(S)\|_F}. \tag{5.15}$$

which is the same as the iteration in Blondel's Algorithm 5 (take even iteration). □

5.3 Some Observations and Proposed Methods

Algorithm 5 is, in fact, a power method, so the rate of convergence depends on the ratio $|\lambda_2|/|\lambda_1|$ the largest two eigenvalues of M^2 with $|\lambda_1| > |\lambda_2|$. Algorithm 7 is equivalent to Blondel's Algorithm 5 when $k = \min(m, n)$. For low-rank approximation, Algorithm 7 exhibits linear convergence as well, although this is not proven in [CAD13]. To avoid this deficiency, second-order information about the cost function can be used to get higher rates of convergence.

The feasible sets (5.8) and (5.10) in the low-rank approximation proposed by Cason et al. have either manifold structure or manifold-like structure. For the approximation with k identical singular values, the set (5.8) has a manifold structure. The general optimization algorithms introduced in Chapter 2 can be used to solve the optimization problem (5.7) on the set (5.8). For the second

kind of approximation with at most k possibly not equal singular values, set (5.10) does not have a manifold structure, but it can be seen as the union of several fixed-rank manifolds. Thus, the modified Riemannian optimization methods (MROM) proposed in Chapter 3 can be applied to solve (5.7) on the set (5.10).

For (5.7) on the set (5.10) when the geometric multiplicity of the extremal eigenvalue of M^2 is more than 1, the eigenspace associated to the extremal eigenvalue has dimension greater than 1. MROM can only guarantee convergence to an eigenvector, not necessarily the unique one with the largest 1-norm.

Since it is not necessarily known a priori if the geometric multiplicity is greater than 1, it is necessary to develop algorithms to handle, in a seamless fashion, problems with geometric multiplicity greater than 1 as well as those with multiplicity 1. The following three modifications to MROM will be investigated:

1. Add a penalty term on the cost function (5.7), i.e., a new cost function
$$\Phi_2(S) := \mathrm{tr}(S^T\mathcal{M}^2(S)) + \lambda \mathbf{1}^T S\mathbf{1}, \tag{5.16}$$
where λ is a penalty coefficient and $\mathbf{1}$ is a vector of all 1 and solve using MROM.

2. Using MROM, solve the following Augmented Lagrangian cost function
$$\Phi_3(S) = \mathbf{1}^T S\mathbf{1} - \lambda\|\mathrm{grad}\Phi_F(S)\|_F^2 + \mu\|\mathrm{grad}\Phi_F(S)\|_F^4, \tag{5.17}$$
where λ is a Lagrange multiplier.

3. Find the optimal solution S_* for the cost function (5.7) using MROM. Then using a second application of MROM with S_* as an initial condition optimize the cost function with an additional penalty term
$$\Phi_4(S) = \mathbf{1}^T S\mathbf{1} - \lambda\|\mathrm{grad}\Phi_F(S)\|_F^2, \quad S_0 = S_*, \tag{5.18}$$
where $\mathrm{grad}\Phi_F(S) = 2\mathcal{M}^2(S) - 2\mathrm{tr}(S^T\mathcal{M}^2(S))S$ is the full gradient of cost function (5.7).

Note that all three default to MROM for problems with geometric multiplicity 1.

A geometric multiplicity greater than 1 appears only under certain conditions and appears to be uncommon. It follows from Perron-Frobenius Theorem that the eigenspaces associated with the Perron-Frobenius eigenvalue is one-dimensional if the non-negative matrix M is primitive, i.e., $(M^h)_{ij} > 0$ for some power h. For (5.7), the following proposition shows if A and B satisfy a certain condition, there exists a pair (i, j) such that $(M^h)_{ij} = 0$ for some power h.

Theorem 30. *Let* $\mathcal{Q} = \{q | a \text{ sequence of length } h \text{ with elements that are either } 1 \text{ or } T\}$ *and* $A^q = \Pi_{i=1}^h A^{q_i}$ *where* $A^{q_i} = A$ *or* A^T. *Let* $M^h = (B \otimes A + B^T \otimes A^T)^h = \sum_{q \in \mathcal{Q}} (B^q \otimes A^q)$. *Then the* $((k-1)n_B + h, (j-1)n_A + i)$*-th element of* M^h *is equal to zero if and only if the product of the* (h, i)*-th element of* A^q *and the* (k, j)*-th element of* B^q *is equal to zero for all* $a \in \mathcal{Q}$.

Proof. The $((k-1)n_B + h, (j-1)n_A + i)$-th element of M^h can be represented by

$$\text{vec}(e_{hk})^T M^h \text{vec}(e_{ij}) = \text{vec}(e_{hk})^T \sum_{q \in \mathcal{Q}} (B^q \otimes A^q) \text{vec}(e_{ij}), \tag{5.19}$$

where $e_{rs} \in \mathbb{R}^{n_A \times n_B}$ represents a basis element matrix with (r, s)-th entry equal to 1, and all others equal to 0.

Since A and B are non-negative matrices, $(B^q \otimes A^q)$ is also a non-negative matrix and therefore, the $((k-1)n_B + h, (j-1)n_A + i)$-th element of M^h is equal to zero if and only if $\forall q \in \mathcal{Q}$,

$$\text{vec}(e_{hk})^T (B^q \otimes A^q) \text{vec}(e_{ij}) = \text{tr}\left(e_{hk}^T A^q e_{ij} (B^q)^T\right) = 0. \tag{5.20}$$

Since $e_{hk}^T A^q$ is an $n_B \times n_A$ matrix with the k-th row equal to the h-th row of A^q and all other rows zero and, similarly, $e_{ij}(B^q)^T$is an $n_A \times n_B$ matrix with the i-th row equal to the j-th column of B^q and all other rows zero, it follows that $\text{tr}\left(e_{hk}^T A^q e_{ij} (B^q)^T\right) = 0$ if and only if the product of the (h, i)-th element of A^q and the (k, j)-th element of B^q is 0. □

5.4 Approximation with k Identical Singular Values

We first look at the feasible set (5.8) in [CAD13] with k identical singular values

$$\mathcal{S}_k(m, n) = \left\{ U \hat{I}_k V^T \in \mathbb{R}^{m \times n} : U \in \text{St}(m, k), V \in \text{St}(n, k), \right. \\ \left. \hat{I}_k = I_k / \|I_k\|_F = I_k / \sqrt{k} \right\}.$$

This set has a manifold structure [CAVD11]. In this case, the general Riemannian optimization algorithms reviewed in Chapter 2 can be used directly to solve problem (5.7). In the following, the crucial ingredients needed in the general Riemannian optimization algorithms to this feasible set are introduced first. Then some experimental results are used to show the efficiency.

5.4.1 Riemannian Gradient

The tangent space to the feasible set $\mathcal{S}_k(m,n)$ at a point $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$ is

$$\begin{aligned} \mathrm{T}_S\mathcal{S}_k(m,n) &:= \{\dot{\gamma}(0) : \gamma \text{ curve on } \mathcal{S}_k(m,n) \textit{ with } \gamma(0) = S\} \\ &= \left\{ \begin{array}{r} U\Omega V^T + UK_V^T V_\perp^T + U_\perp K_U V^T \ s.t. \\ \Omega \in \mathcal{S}_{\mathrm{skew}}(k), K_U \in \mathbb{R}^{(m-k)\times k}, K_V \in \mathbb{R}^{(n-k)\times k} \end{array} \right\}, \end{aligned} \tag{5.21}$$

where $U_\perp, V_\perp$ are any orthogonal complements of U, V and $\mathcal{S}_{skew}(k)$ denotes the set of skew matrices of order k, $\mathrm{skew}(A) = \frac{A-A^*}{2}$.

The normal space to the feasible set $\mathcal{S}_k(m,n)$ at a point $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$ is

$$\begin{aligned} \mathrm{N}_S\mathcal{S}_k(m,n) &:= \{\zeta : \langle \zeta, \xi \rangle_F = 0, \forall \xi \in \mathrm{T}_S\mathcal{S}_k(m,n)\} \\ &= \left\{ UHV^T + U_\perp K V_\perp^T \ s.t. \ H \in \mathcal{S}_{\mathrm{sym}}(k), K \in \mathbb{R}^{(m-k)\times(n-k)} \right\}, \end{aligned} \tag{5.22}$$

with $\mathcal{S}_{\mathrm{sym}}(k)$ denotes the set of symmetric matrices of order k, $\mathrm{sym}(A) = \frac{A+A^*}{2}$.

By restricting the Euclidean inner product on $\mathbb{R}^{m\times n}$,

$$\langle A, B \rangle = \mathrm{tr}(A^T B) \ \text{ with } A, B \in \mathbb{R}^{m\times n},$$

to the tangent space, $\mathcal{S}_k(m,n)$ is a Riemannian manifold with the Riemannian metric

$$g_S(\xi, \eta) := \langle \xi, \eta \rangle = \mathrm{tr}(\xi^T \eta) \ \ \textit{with } S \in \mathcal{S}_k(m,n) \textit{ and } \xi, \eta \in \mathrm{T}_S\mathcal{S}_k(m,n) \tag{5.23}$$

where the tangent vectors ξ, η are seen as matrices in $\mathbb{R}^{m\times n}$.

Once the metric is defined, the notation of gradient of an objective function can be introduced. Since $\mathcal{S}_k(m,n)$ is embedded in $\mathbb{R}^{m\times n}$, the Riemannian gradient is given as the orthogonal projection of the gradient of cost function $\Phi(S)$, which is a function on $\mathbb{R}^{m\times n}$, onto the tangent space at S, given by

$$\begin{aligned} P_{\mathrm{T}_S\mathcal{S}_k(m,n)} &: \mathbb{R}^{m\times n} \to \mathrm{T}_S\mathcal{S}_k(m,n) \\ Z \to P_S Z &= U\mathrm{skew}(U^T Z V)V^T + U(U^T Z V_\perp)V_\perp^T + U_\perp(U_\perp^T Z V)V^T \\ &= -U\mathrm{sys}(U^T Z V)V^T + UU^T Z + ZVV^T - UU^T ZVV^T. \end{aligned} \tag{5.24}$$

Similarly, the orthogonal projection of the gradient of cost function $\Phi(S)$ onto the normal space at S is

$$\begin{aligned} P_{\mathrm{N}_S\mathcal{S}_k(m,n)} &: \mathbb{R}^{m\times n} \to \mathrm{N}_S\mathcal{S}_k(m,n) \\ Z \to P_S^\perp Z &= U\mathrm{sym}(U^T Z V)V^T + (I_m - UU^T)Z(I_n - VV^T). \end{aligned} \tag{5.25}$$

Since the (Euclidean) gradient of the cost function $\Phi(S)$ is $2\mathcal{M}^2(S)$, where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ and $[\mathcal{M}(S)]_{ij} = [ASB^T + A^TSB]_{ij}$, projecting the Euclidean gradient onto tangent space $\mathrm{T}_S\mathcal{S}_k(m,n)$, yields the Riemannian gradient

$$\begin{aligned}
\mathrm{grad}\Phi(S) &:= \mathrm{P}_S 2\mathcal{M}^2(S) \\
&= -U\frac{U^T 2\mathcal{M}^2(S)V + V^T 2\mathcal{M}^2(S)^T U}{2}V^T + 2UU^T\mathcal{M}^2(S) + 2\mathcal{M}^2(S)VV^T - 2UU^T\mathcal{M}^2(S)VV^T \\
&= -3UU^T\mathcal{M}^2(S)VV^T - UV^T\mathcal{M}^2(S)^T UV^T + 2UU^T\mathcal{M}^2(S) + 2\mathcal{M}^2(S)VV^T.
\end{aligned} \tag{5.26}$$

5.4.2 Riemannian Retraction

Given $S \in \mathcal{S}_k(m,n)$ and $\dot{S} \in \mathrm{T}_S\mathcal{S}_k(m,n)$, similar to Section 4.3.3, we give the following three ways of retractiing onto the manifold $\mathcal{S}_k(m,n)$: the **SVD-type retraction**, the **QR-type retraction** and the **polar-type retraction**.

Let $S = UV^T \in \mathcal{S}_k(m,n)$, the $\dot{S} \in \mathrm{T}_S\mathcal{S}_k(m,n)$ can be computed as

$$\dot{S} = \dot{U}V^T + U\dot{V}^T. \tag{5.27}$$

The expression of $(\dot{U}, \dot{V})$ can be derived as follows. Since $U \in \mathrm{St}(m,k), V \in \mathrm{St}(n,k)$, in view of the form of the tangent space to the Stiefel manifold $\mathrm{St}(n,p)$ at a point X,

$$\mathrm{T_X St}(n,p) = \{X\Omega + X_\perp K : \Omega^T = -\Omega, K \in \mathbb{R}^{(n-p)\times p}\}, \tag{5.28}$$

we have

$$\begin{aligned}
\dot{U} &= U\Omega_U + U_\perp K_U, \\
\dot{V} &= V\Omega_V + V_\perp K_V,
\end{aligned} \tag{5.29}$$

where $\Omega_U^T = -\Omega_U, \Omega_V^T = -\Omega_V$, $K_U \in \mathbb{R}^{(m-k)\times k}, K_V \in \mathbb{R}^{(n-k)\times k}$. It follows that

$$\begin{aligned}
\dot{S} &= (U\Omega_U + U_\perp K_U)V^T + U(V\Omega_V + V_\perp K_V)^T \\
&= U(\Omega_U + \Omega_V^T)V^T + U_\perp K_U V^T + U\Omega_V V^T
\end{aligned} \tag{5.30}$$

Multiplying both sides by U^T from the left and V from the right, we get

$$\Omega_U + \Omega_V^T = U^T\dot{S}V. \tag{5.31}$$

Similarly, we have

$$\begin{aligned}
K_U &= U_\perp^T\dot{S}V, \\
K_V^T &= U^T\dot{S}V_\perp.
\end{aligned} \tag{5.32}$$

Since Ω_U, Ω_V are skew matrices, $\Omega_U + \Omega_V^T$ is a skew matrix. There are many possibilities, but for convenience, we take Ω_U and Ω_V as follows:

$$\begin{aligned} \Omega_U &= \frac{1}{2} U^T \dot{S} V, \\ \Omega_V^T &= -\frac{1}{2} (U^T \dot{S} V)^T \end{aligned} \tag{5.33}$$

Therefore, we have the explicit expression of $\dot{U}$ and $\dot{V}$:

$$\dot{U} = U\Omega_U + U_\perp K_U = \frac{1}{2} U U^T \dot{S} V + U_\perp U_\perp^T \dot{S} V = \dot{S} V - \frac{1}{2} U U^T \dot{S} V, \tag{5.34}$$

$$\dot{V} = V\Omega_V + V_\perp K_V = -\frac{1}{2} V V^T \dot{S}^T U + V_\perp V_\perp^T \dot{S}^T U = \dot{S}^T U - \frac{1}{2} V U^T \dot{S} V - V V^T \dot{S}^T U. \tag{5.35}$$

The **SVD-type retraction** is a projective retraction ([AM12]):

$$R_S(\dot{S}) = \frac{1}{\sqrt{k}} U_k V_k^T, \tag{5.36}$$

where $[U, D, V] = \mathrm{svd}(S + \dot{S})$ is (ordered) singular value decomposition (SVD), and U_k, V_k are first k columns of U, V respectively.

The **QR-type retraction** is defined as

$$R_S(\dot{S}) = \frac{1}{\sqrt{k}} U_+ V_+^T, \tag{5.37}$$

where

$$\begin{aligned} U_+ &= qf(U + \dot{U}), \\ V_+ &= qf(V + \dot{V}), \end{aligned} \tag{5.38}$$

where $\dot{U}$, $\dot{V}$ are defined in (5.34), (5.35) and $qf(A)$ denotes the orthogonal Q factor of the QR decomposition of a matrix $A = QR$.

An alternative choice is the **polar-type retraction**:

$$R_S(\dot{S}) = \frac{1}{\sqrt{k}} U_+ V_+^T, \tag{5.39}$$

where

$$\begin{aligned} U_+ &= uf(U + \dot{U}), \\ V_+ &= uf(V + \dot{V}), \end{aligned} \tag{5.40}$$

and the symbol $uf(\cdot)$ denotes the orthogonal component of the polar decomposition.

5.4.3 Vector Transport

In our framework, vector transport can be represented by an m by n matrix. Given two points S_1 and S_2 in $\mathcal{S}_k(m,n)$, the corresponding tangent spaces are denoted T_{S_1}, T_{S_2}. We choose the isometric vector transport $\mathcal{T}$ from S_1 to S_2 to be the direct rotation from $\mathrm{T}_{S_1}\mathcal{S}_k(m,n)$ to $\mathrm{T}_{S_2}\mathcal{S}_k(m,n)$, restricted to act on $\mathrm{T}_{S_1}\mathcal{S}_k(m,n)$.

Efficient implementations of the direct rotation vector transport are constructed following Huang's approach [Hua13]. Let B_{S_1} and B_{S_2} be an orthonormal basis of $\mathrm{T}_{S_1}\mathcal{S}_k(m,n)$ and $\mathrm{T}_{S_2}\mathcal{S}_k(m,n)$ respectively. Hence B_{S_1} and B_{S_2} can be viewed as mn by d matrices (d is the intrinsic dimension) and $B_{S_1}^T B_{S_1} = B_{S_2}^T B_{S_2} = I_d$. The direct-rotation transport from S_1 to S_2 is then given by

$$\mathcal{T} = B_{S_2} V U^T B_{S_1} \tag{5.41}$$

where $B_{S_1}^T B_{S_2} = U\Sigma V^T$ is a singular value decomposition (SVD).

If the codimension, $mn - d$, is sufficiently smaller than the dimension, d, and if, moreover, orthonormal bases N_{S_1} and N_{S_2} are available, then the following vector transport becomes computationally advantageous,

$$\mathcal{T} = (I - Q_{S_1} Q_{S_1}^T) + Q_{S_2} V U^T Q_{S_2}^T, \tag{5.42}$$

where $Q_{S_1}^T Q_{S_2} = U\Sigma V^T$ is an SVD and Q_{S_1}, Q_{S_2} are obtained by orthonormalizing $(I - N_{S_1} N_{S_1}^T) N_{S_2}$ and $(I - N_{S_2} N_{S_2}^T) N_{S_1}$.

If smoothness is imposed, i.e. $B : S \to B_S$ and $N : S \to \mathrm{N}_S$ are smooth functions to build basis of $\mathrm{T}_S\mathcal{S}_k(m,n)$ and $\mathrm{N}_S\mathcal{S}_k(m,n)$, then we have a simpler form of isometric vector transports:

$$\mathcal{T} = B_{S_2} B_{S_1}^T, \tag{5.43}$$

$$\mathcal{T} = I - Q_{S_1} Q_{S_1}^T - Q_{S_2} Q_{S_1}^T. \tag{5.44}$$

Using this idea, we must construct the functions to build the bases. Note that since

$$\mathrm{T}_S\mathcal{S}_k(m,n) = \left\{ \begin{array}{r} U\Omega V^T + U K_V^T V_\perp^T + U_\perp K_U V^T \ s.t. \\ \Omega \in \mathcal{S}_{\mathrm{skew}}(k), K_U \in \mathbb{R}^{(m-k)\times k}, K_V \in \mathbb{R}^{(n-k)\times k} \end{array} \right\},$$

an orthonormal basis of $\mathrm{T}_S\mathcal{S}_k(m,n)$, denoted by B_S, is given by

$$\begin{aligned} &\{\frac{1}{\sqrt{2}} U(e_i e_j^T - e_j e_i^T) V : i = 1, \cdots, k, j = i+1, \cdots, k\} \\ &\cup \{U(e_j \tilde{e}_i^T) V_\perp^T : i = 1, \cdots, n-k, j = 1, \cdots, k\} \\ &\cup \{U_\perp (\hat{e}_i e_j^T) V^T : i = 1 \cdots, m-k, j = 1, \cdots, k\}, \end{aligned} \tag{5.45}$$

where $(e_1, \cdots, e_k)$ is the canonical basis of $\mathbb{R}^k$, $(\hat{e}_1, \cdots, \hat{e}_{m-k})$ is the canonical basis of $\mathbb{R}^{m-k}$ and $(\tilde{e}_1, \cdots, \tilde{e}_{n-k})$ is the canonical basis of $\mathbb{R}^{n-k}$. Similarly, we can construct the basis for normal space

$$\mathrm{N_S}\mathcal{S}_k(m,n) = \left\{ UHV^T + U_\perp K V_\perp^T \ s.t.\ H \in \mathcal{S}_{\mathrm{sym}}(k), K \in \mathbb{R}^{m-k \times n-k} \right\}.$$

Using N_S to denote the basis, which is given by

$$\begin{aligned} &\{Ue_i e_i^T V^T : i = 1, \cdots, k\} \cup \{\frac{1}{\sqrt{2}} U(e_i e_j^T + e_j e_i^T)V^T : i = 1, \cdots, k, j = i+1, \cdots, k\} \\ &\cup \{U_\perp \tilde{e}_i \hat{e}_j V_\perp^T : i = 1, \cdots, m-k, j = 1, \cdots, n-k\}. \end{aligned} \tag{5.46}$$

The columns of B_S and N_S are thus chosen as the "vec" of the basis elements.

We can also derive the vector transport by the differentiated retraction of (5.37).

Proposition 31. *Let $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$, $\xi, \eta \in \mathrm{T}_S\mathcal{S}_k(m,n)$. Assuming ξ and η have the following structure*

$$\begin{aligned} \xi &= \dot{U}_1 V^T + U\dot{V}_1^T, \\ \eta &= \dot{U}_2 V^T + U\dot{V}_2^T. \end{aligned}$$

Then the vector transport by the differentiated retraction of (5.37) *is*

$$\mathcal{T}_\eta \xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1) qf(V + \dot{V}_2)^T + qf(U + \dot{U}_2))(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T, \tag{5.47}$$

where $\mathcal{T}_{\dot{U}_2}(\dot{U}_1)$ is a differentiated retraction on Stiefel manifold [AMS08, Example 8.1.5] and $qf(\cdot)$ denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R.

Proof. Based on the definition of the vector transport by differentiated retraction and the QR-type retraction (5.37), we have

$$\begin{aligned} \mathcal{T}_\eta \xi &= \frac{d}{dt} R_X(\eta + t\xi)\Big|_{t=0} \\ &= \frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1) qf(V + \dot{V}_2 + t\dot{V}_1)^T]\Big|_{t=0} \\ &= \frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)] qf(V + \dot{V}_2 + t\dot{V}_1)^T\Big|_{t=0} \\ &+ qf(U + \dot{U}_2 + t\dot{U}_1) \frac{d}{dt} qf(V + \dot{V}_2 + t\dot{V}_1)^T\Big|_{t=0} \\ &+ qf(U + \dot{U}_2 + t\dot{U}_1) \frac{d}{dt}[qf(V + \dot{V}_2 + t\dot{V}_1)]^T\Big|_{t=0}. \end{aligned} \tag{5.48}$$

Since $U \in \mathrm{St}(m,r), V \in \mathrm{St}(n,r)$, according to the vector transport by differentiated retraction on Stiefel manifold [AMS08], we have for $\dot{U}_1, \dot{U}_2 \in \mathrm{T}_U\mathrm{St}(m,r)$,

$$\frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)]|_{t=0} = \mathcal{T}_{\dot{U}_2}(\dot{U}_1), \tag{5.49}$$

and for $\dot{V}_1, \dot{V}_2 \in \mathrm{T}_V\mathrm{St}(n,r)$,

$$\frac{d}{dt}[qf(V + \dot{V}_2 + t\dot{V}_1)]|_{t=0} = \mathcal{T}_{\dot{V}_2}(\dot{V}_1), \tag{5.50}$$

where

$$\begin{aligned} T_{\dot{U}_2}(\dot{U}_1) &= \mathrm{D}R_U(\dot{U}_2)[\dot{U}_1] \\ &= \mathrm{D}qf(U + \dot{U}_2)[\dot{U}_1] \\ &= R_U(\dot{U}_2)\rho_{\mathrm{skew}}(R_U(\dot{U}_2)^T\dot{U}_1(R_U(\dot{U}_2)^T(U + \dot{U}_2))^{-1}) \\ &+ (I - R_U(\dot{U}_2)R_U(\dot{U}_2)^T)\dot{U}_1(R_U(\dot{U}_2)^T(U + \dot{U}_2))^{-1}, \end{aligned}$$

and $\rho_{\mathrm{skew}}(B)$ denotes the skew-symmetric term of the decomposition of a square matrix B into the sum of a skew-symmetric term and an upper triangular term, i.e.,

$$(\rho_{\mathrm{skew}}(B))_{i,j} = \begin{cases} B_{i,j} & \text{if } i >, j \\ 0 & \text{if } i = j, \\ -B_{j,i} & \text{if } i < j. \end{cases}$$

Substituting (5.49) and (5.50) into (5.48), we have

$$\mathcal{T}_\eta\xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1)qf(V + \dot{V}_2)^T + qf(U + \dot{U}_2)(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T.$$

□

Similarly, vector transport by the differentiated retraction of (5.39) can also be derived and is stated in the following Proposition.

Proposition 32. *Let $S = U\hat{I}_kV^T \in \mathcal{S}_k(m,n)$, $\xi, \eta \in \mathrm{T}_S\mathcal{S}_k(m,n)$. Assuming ξ and η have the following structure*

$$\xi = \dot{U}_1V^T + U\dot{V}_1^T,$$
$$\eta = \dot{U}_2V^T + U\dot{V}_2^T.$$

Then the vector transport by the differentiated retraction of (5.39) *is*

$$\mathcal{T}_\eta\xi = \mathcal{T}_{\dot{U}_2}(\dot{U}_1)uf(V + \dot{V}_2)^T + uf(U + \dot{U}_2))(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T, \tag{5.51}$$

where $\mathcal{T}_{\dot{U}_2}(\dot{U}_1)$ is a vector transport by differentiated retraction of (5.39) *on the Stiefel manifold [Hua13, Lemma 10.2.1] and $uf(\cdot)$ denotes the orthogonal factor of the polar decomposition.*

Proof. Based on the definition of the vector transport by differentiated retraction and the polar-decomposition-type retraction (5.39), we have

$$\begin{aligned}\mathcal{T}_\eta\xi &= \frac{d}{dt}R_X(\eta+t\xi)\Big|_{t=0}\\ &= \frac{d}{dt}[uf(U+\dot{U}_2+t\dot{U}_1)uf(V+\dot{V}_2+t\dot{V}_1)^T]\Big|_{t=0}\\ &= \frac{d}{dt}[uf(U+\dot{U}_2+t\dot{U}_1)]uf(V+\dot{V}_2+t\dot{V}_1)^T\Big|_{t=0}\\ &+ uf(U+\dot{U}_2+t\dot{U}_1)\frac{d}{dt}uf(V+\dot{V}_2+t\dot{V}_1)^T\Big|_{t=0}\\ &+ uf(U+\dot{U}_2+t\dot{U}_1)\frac{d}{dt}[uf(V+\dot{V}_2+t\dot{V}_1)]^T\Big|_{t=0}.\end{aligned} \tag{5.52}$$

Since $U\in\mathrm{St}(m,r), V\in\mathrm{St}(n,r)$, according to the vector transport by differentiated retraction on the Stiefel manifold [Hua13, Lemma 10.2.1], for $\dot{U}_1,\dot{U}_2\in\mathrm{T}_U\mathrm{St}(m,r)$, it follows that

$$\frac{d}{dt}[uf(U+\dot{U}_2+t\dot{U}_1)]|_{t=0}=\mathcal{T}_{\dot{U}_2}(\dot{U}_1), \tag{5.53}$$

and for $\dot{V}_1,\dot{V}_2\in\mathrm{T}_V\mathrm{St}(n,r)$,

$$\frac{d}{dt}[uf(V+\dot{V}_2+t\dot{V}_1)]|_{t=0}=\mathcal{T}_{\dot{V}_2}(\dot{V}_1), \tag{5.54}$$

where

$$\begin{aligned}T_{\dot{U}_2}(\dot{U}_1) &= \mathrm{D}R_U(\dot{U}_2)[\dot{U}_1]\\ &= \mathrm{D}uf(U+\dot{U}_2)[\dot{U}_1]\\ &= R_U(\dot{U}_2)\Omega+(I-R_U(\dot{U}_2)(R_U(\dot{U}_2))^T)\dot{U}_1((R_U(\dot{U}_2))^T(U+\dot{U}_2))^{-1},\end{aligned}$$

and R is (5.39), $\mathrm{vec}\{\Omega\}=((R_U(\dot{U}_2))^T(U+\dot{U}_2)\oplus(R_U(\dot{U}_2))^T(U+\dot{U}_2))^{-1}\mathrm{vec}\{(R_U(\dot{U}_2))^T\dot{U}_1-\dot{U}_1^TR_U(\dot{U}_2)\}$, $\oplus$ is the Kronecker sum, i.e., $A\oplus B=A\otimes I+I\otimes B$.

Substituting (5.53) and (5.54) into (5.52), we have

$$\mathcal{T}_\eta\xi=\mathcal{T}_{\dot{U}_2}(\dot{U}_1)uf(V+\dot{V}_2)^T+uf(U+\dot{U}_2)(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T.$$

□

5.4.4 The Action of Riemannian Hessian

In order to exploit second-order information, we give the following proposition with an expression of the action of Riemannian Hessian.

Proposition 33. *For any $S = U\hat{I}_k V^T \in \mathcal{S}_k(m,n)$ and $\eta \in \mathrm{T_S}\mathcal{S}_k(m,n)$, the Riemannian Hessian of Φ at S in the direction of η satisfies*

$$\mathrm{Hess}\Phi(S)[\eta] = \bigtriangledown_\eta \mathrm{grad}\Phi(S) = P_S(\mathrm{Dgrad}\Phi(S)[\eta])$$

where

$$\begin{aligned}\mathrm{Dgrad}\Phi(S)[\eta] = &-3k^2[\eta S^T\mathcal{M}^2(S)S^TS + S\eta^T\mathcal{M}^2(S)S^TS + SS^T\mathcal{M}^2(\eta)S^TS\\ &+ SS^T\mathcal{M}^2(S)\eta^TS + SS^T\mathcal{M}^2(S)S^T\eta]\\ &- k[\eta\mathcal{M}^2(S)^TS + S\mathcal{M}^2(\eta)^TS + S\mathcal{M}^2(S)^T\eta]\\ &+ 2k[\eta S^T\mathcal{M}^2(S) + S\eta^T\mathcal{M}^2(S) + SS^T\mathcal{M}^2(\eta)]\\ &+ 2k[\mathcal{M}^2(\eta)S^TS + \mathcal{M}^2(S)\eta^TS + \mathcal{M}^2(S)S^T\eta].\end{aligned}$$

Proof. We already have the Riemannian gradient of the cost function $\Phi(S)$ on $\mathcal{S}_k(m,n)$ is given by:

$$\begin{aligned}&\mathrm{grad}\Phi(S) = 2\mathrm{P}_S\mathcal{M}^2(S)\\ &= -3UU^T\mathcal{M}^2(S)VV^T - UV^T\mathcal{M}^2(S)^TUV^T + 2UU^T\mathcal{M}^2(S) + 2\mathcal{M}^2(S)VV^T \qquad (5.55)\\ &= -3k^2SS^T\mathcal{M}^2(S)S^TS - kS\mathcal{M}^2(S)^TS + 2kSS^T\mathcal{M}^2(S) + 2k\mathcal{M}^2(S)S^TS.\end{aligned}$$

Since $\mathcal{S}_k(m,n)$ is a Riemannian submanifold of a Euclidean space, according to [AMS08, Equation (5.15)] ,

$$\mathrm{Hess}\Phi(S)[\eta] = \nabla_\eta \mathrm{grad}\Phi(S) = \mathrm{P}_S(\mathrm{Dgrad}\Phi(S)[\eta]), \qquad (5.56)$$

where $Dg(x)[H]$ is a directional derivative of g at x along H. We now differentiate (5.55) to get a matrix representation of the directional derivative of Riemannian gradient, gradΦ, at S along η.

$$\begin{aligned}D\mathrm{grad}\Phi(S)[\eta] = &-3k^2[\eta S^T\mathcal{M}^2(S)S^TS + S\eta^T\mathcal{M}^2(S)S^TS + SS^T\mathcal{M}^2(\eta)S^TS\\ &+ SS^T\mathcal{M}^2(S)\eta^TS + SS^T\mathcal{M}^2(S)S^T\eta]\\ &- k[\eta\mathcal{M}^2(S)^TS + S\mathcal{M}^2(\eta)^TS + S\mathcal{M}^2(S)^T\eta]\\ &+ 2k[\eta S^T\mathcal{M}^2(S) + S\eta^T\mathcal{M}^2(S) + SS^T\mathcal{M}^2(\eta)]\\ &+ 2k[\mathcal{M}^2(\eta)S^TS + \mathcal{M}^2(S)\eta^TS + \mathcal{M}^2(S)S^T\eta].\end{aligned}$$

Finally, the Hessian of a cost function Φ at S in the direction of η satisfies

$$\mathrm{Hess}\Phi(S)[\eta] = \bigtriangledown_\eta \mathrm{grad}\Phi(S) = \mathrm{P}_S(D\mathrm{grad}\Phi(S)[\eta]).$$

□

5.4.5 Experiments

In this section, we compare the performance of Cason's iteration method with those of the general Riemannian manifold methods introduced in Chapter 2. Six Riemannian algorithms are used, i.e., Riemannian steepest descent with line search (RTR-SD), Riemannian trust region with symmetric rank-one update (RTR-SR1), limited-memory RTR-SR1 (LRTR-SR1), general Riemannian trust-region method (RTR-Newton), Riemnnian Broyden-Fletcher-Goldfarb-Shannon (RBFGS) and limited-memory RBFGS (LRBFGS). Four of them are combined with a trust region: RTR-SD, RTR-SR1, LRTR-SR1, RTR-Newton. The rest are combined with a line search algorithm, i.e., RBFGS with inversion Hessian approximation $\mathcal{H}_k$, LRBFGS (limited-memory RBFGS). The inner iteration algorithm of trust region is the truncated CG inner iteration [AMS08, Section 7.3.2]. The θ, κ parameters in the inner iteration stopping criteria [AMS08, (7.10)] are set to $1, 0.1$. τ_1, τ_2 in trust region are 0.25 and 2 respectively. The initial radius Δ_0 is 1. c in RTR-SR1 and LRTR-SR1 is set to 0.1, ν is the square root of machine epsilon. The constants c_1, c_2 used in the Wolfe conditions are $1e-04$ and 0.999 respectively.

The results presented are obtained by implementing the different algorithms in Matlab (Version 7.10.0) on a Mac platform with 2.4 GHz and 4 GB memory.

Unless otherwise indicated in the description of the experiments, the following test data parameters are used. The test graph is a random graph based on Erdós-Rényi model with 100 nodes and average of outgoing edges of each node is 10 and the self-similarity matrix is computed. The initial iterate S_0 of Riemannian algorithms is composed by two parts U_0, V_0, where U_0, V_0 are the first k columns of U and V generated by applying Matlab's function SVD on an all 1 matrix. The initial iterates S_0 of Cason's iteration method, i.e. Algorithm 6, is an all 1 matrix, which is setting in Step 1 of Algorithm 7. The stopping criterion required the ratio of the norm of final gradient and the norm of initial gradient is less than 10^{-7} for all methods. To obtain sufficiently stable timing results, an average time is taken of five runs with identical parameters.The notation used when reporting the experimental results is given in Table 5.2.

In Section 5.4.5.2, different retractions are compared. Section 5.4.5.2 compares the performances of Algorithm 7 and RTR-Newton method. In Section 5.4.5.3, the performances of different Riemannian algorithms are compared with Algorithm 7.

Table 5.2: Notation for reporting the experimental results.

Rerr	relative error $\frac{\|S-S^B\|_F}{\|S^B\|_F}$, where S^B is the full rank matrix obtained by Blondel's algorithm
f	final value of the cost function (5.7)
gf_0	Riemannian metric value of the initial gradient
gf_f	Riemannian metric value of the final gradient
iter	number of iterations
nf	number of function evaluations
nf	number of function evaluations
ng	number of gradient evaluations
nH	number of operations of the form $\mathcal{H}\eta$
nV	number of vector transports
nR	number of retraction evaluations
t	average time (seconds)

5.4.5.1 Performance of different retractions. Three types of retractions are proposed in Section 5.4.2. In this section, the results of RTR-Newton with different retractions are compared. Table 5.3 shows the results of different retractions for different k in RTR-Newton. From the table, we observe as k increases, all three retractions get almost the same relative error and the same final value of the cost function (5.7). The computation time of QR-type retraction and polar-decomposition type retraction are almost the same. But the computation time of SVD-type retraction is more than the other two types of retraction. Table 5.3 also shows the number of operations of the form $\mathcal{H}\eta$ in SVD-type retraction is more than that in the other two types of retractions when k is small hence the difference in computational times.

5.4.5.2 Comparision of Cason's Iteration algorithm and RTR-Newton. From the comparison of different retractions, we observe polar-decomposition-type retraction has time advantages compared to the other two. In the following experiments, this retraction is always used. In this section, the results of low-rank approximation generated by Cason's iteration method and RTR-Newton are compared. Table 5.4 shows the results. It shows RTR-Newton method has noticeable time advantages compared with iteration method, especially when k gets large. It can also be observed that when the rank of the approximation increases i.e. as k increases, the relative error ($\frac{\|S-S^B\|_F}{\|S^B\|_F}$, where S^B is the full rank matrix obtained by Blondel's algorithm 5), also increases, although the values of the cost function decrease. These counterintuitive results occur because similarity matrices do not usually have identical eigenvalues.

Table 5.3: Comparison of different retractions for approximation with k identical singular values. The subscript $\pm z$ indicates a scale of $10^{\pm z}$.

k	Rerr	Retraction	f	time(s)	gf_f/gf_0	gf_f	nF	nG	nH	nR
1	1.875_{-02}	SVD-type	4.404_{+04}	1.704_{-01}	1.578_{-10}	2.846_{-06}	7	7	23	6
		QR-type	4.404_{+04}	1.058_{+00}	1.995_{-09}	3.599_{-05}	7	7	21	6
		PD-type	4.404_{+04}	1.332_{-01}	1.995_{-09}	3.599_{-05}	7	7	21	6
2	7.655_{-01}	SVD-type	3.203_{+04}	4.457_{-01}	2.108_{-10}	2.691_{-06}	11	11	67	10
		QR-type	3.203_{+04}	4.026_{-01}	7.468_{-09}	9.531_{-05}	11	11	64	10
		PD-type	3.203_{+04}	4.184_{-01}	7.468_{-09}	9.531_{-05}	11	11	64	10
3	9.195_{-01}	SVD-type	2.706_{+04}	6.205_{-01}	5.055_{-10}	5.303_{-06}	17	17	100	16
		QR-type	2.706_{+04}	6.200_{-01}	1.346_{-09}	1.412_{-05}	18	18	103	17
		PD-type	2.706_{+04}	6.087_{-01}	1.346_{-09}	1.412_{-05}	18	18	103	17
4	9.996_{-01}	SVD-type	2.359_{+04}	5.773_{-01}	2.383_{-08}	2.184_{-04}	15	15	93	14
		QR-type	2.359_{+04}	8.849_{-01}	1.796_{-08}	1.646_{-04}	17	17	96	16
		PD-type	2.359_{+04}	6.001_{-01}	1.796_{-08}	1.646_{-04}	17	17	96	16
5	1.052_{+00}	SVD-type	2.113_{+04}	1.304_{+00}	8.339_{-08}	6.837_{-04}	24	24	218	23
		QR-type	2.116_{+04}	7.262_{-01}	2.126_{-09}	1.743_{-05}	22	22	119	21
		PD-type	2.116_{+04}	7.081_{-01}	2.126_{-09}	1.743_{-05}	22	22	119	21
6	1.088_{+00}	SVD-type	1.916_{+04}	1.729_{+00}	7.720_{-11}	5.844_{-07}	30	30	295	29
		QR-type	1.918_{+04}	7.965_{-01}	1.090_{-09}	8.253_{-06}	22	22	134	21
		PD-type	1.918_{+04}	7.885_{-01}	1.090_{-09}	8.253_{-06}	22	22	134	21
10	1.165_{+00}	SVD-type	1.331_{+04}	3.162_{+00}	2.361_{-09}	1.365_{-05}	31	31	418	30
		QR-type	1.331_{+04}	2.934_{+00}	3.044_{-09}	1.760_{-05}	33	33	463	32
		PD-type	1.331_{+04}	2.924_{+00}	2.942_{-09}	1.701_{-05}	31	31	480	30

Table 5.4: Comparison of Cason's iteration method and RTR-Newton for approximation with k identical singular values. The subscript $\pm z$ indicates a scale of $10^{\pm z}$.

		Iteration Method				RTR-Newton Method			
k	Rerr	f	t	gf_f/gf_0	gf_f	f	t	gf_f/gf_0	gf_f
1	3.48_{-02}	4.69_{+04}	0.67	9.03_{-09}	2.15_{-03}	4.69_{+04}	0.18	1.84_{-09}	4.38_{-05}
2	7.66_{-01}	2.30_{+04}	1.69	9.57_{-08}	1.61_{-03}	3.26_{+04}	0.41	2.20_{-08}	3.73_{-04}
3	9.20_{-01}	2.62_{+04}	1.13	9.99_{-08}	1.37_{-03}	2.62_{+04}	0.90	6.38_{-09}	8.68_{-05}
4	1.00_{+00}	2.29_{+04}	10.83	9.99_{-08}	1.18_{-03}	2.29_{+04}	0.49	2.26_{-08}	2.67_{-04}
5	1.05_{+00}	1.95_{+04}	12.33	9.99_{-08}	1.02_{-03}	1.95_{+04}	1.35	7.42_{-09}	7.74_{-05}
10	1.16_{+00}	1.15_{+04}	31.09	9.94_{-08}	7.33_{-04}	1.15_{+04}	1.79	4.16_{-08}	3.05_{-04}

5.4.5.3 Comparison of other Riemannian algorithms. In this section, six Riemannian optimization methods are compared with Algorithm 7. Results are shown in Table 5.4.5.3, where the values in brackets show the iteration numbers and the missing values $(-)$ mean the result needs more time to reach the stop criteria or has exceeded the allowable amount of memory. From the results, we observe limited-memory RBFGS method is comparable with RTR-Newton method. What is more, the Riemannian optimization methods, except RBFGS method, have significant time advantages compared with iteration method, especially when k is small.

Table 5.5: Computation time with iteration numbers in brackets for approximation of similarity matrix with k identical singular values using different methods. RTR-SD stands for Riemannian trust region-steepest descend method, RTR-SR1 stands for Riemannian trust region-SR1 method, LRTR-SR1 stands for limited memory RTR-SR1, RTR stands for RTR-Newton method, LRBFGS stands for limited memory BFGS method.

k	rel_err	Iteration method	RTR-SD	RTR-SR1	LRTR-SR1	RTR	LRBFGS	RBFGS
1	0.03	0.68(37)	0.23(29)	0.43(27)	0.32(21)	0.26(6)	0.27(12)	1.53(36)
2	0.77	1.54(83)	0.51(148)	1.03(60)	0.52(61)	0.54(10)	0.46(49)	11.77(318)
3	0.92	1.46(73)	1.37(399)	4.63(124)	1.62(184)	0.92(17)	0.79(71)	20.60(287)
4	0.99	0.53(26)	2.01(624)	11.17(175)	1.84(215)	0.93(15)	0.84(79)	-
5	1.05	30.93(1868)	1.79(494)	26.34(262)	3.52(280)	1.30(29)	0.84(79)	-
6	1.09	44.63(2152)	3.22(881)	41.53(293)	34.22(444)	1.33(28)	1.15(96)	-
10	1.16	47.33(2739)	13.36(3206)	–	12.61(1196)	2.55(31)	5.78(464)	-
15	1.21	123.17(6764)	–	–	–	4.29(33)	7.24(498)	-
50	1.31	154.39(8424)	–	–	–	8.22(29)	45.13(1599)	-

5.5 Approximation of rank at most k

Since the relative error of the approximation with k identical singular values increases as k increases, we enhance our method with a diagonal positive scaling D_k. However, there is no rigorous way of how to choose k, i.e., if k is chosen "too small", the result may not be a good approximation of the similarity matrix; if k is chosen "too large", the algorithm may require excessive computation. In this section, we consider the set of rank at most k, i.e. (5.10) in [CAD13]:

$$\mathcal{S}_{\leq k}(m,n) = \left\{ \begin{array}{r} UDV^T \in \mathbb{R}^{m\times n} : U \in \mathrm{St}(m,k), V \in \mathrm{St}(n,k), \\ D \text{ diagonal}, \|D\|_F = 1 \end{array} \right\}.$$

The set $\mathcal{S}_{\leq k}(m,n)$ is not a manifold, it can be written as

$$\mathcal{S}_{\leq k}(m,n) = \bigcup_{r\leq k} \mathcal{S}_r, \tag{5.57}$$

where

$$\mathcal{S}_r(m,n) = \left\{ \begin{array}{r} UD_rV^T \in \mathbb{R}^{m\times n} : U \in \mathrm{St}(m,r), V \in \mathrm{St}(n,r), \\ D_r \text{ is a diagonal matrix }, \|D_r\|_F = 1 \end{array} \right\} \tag{5.58}$$

is a fixed-rank manifold with r nonzero singular values. Thus, the modified Riemannian optimization method can be applied once the required differential geometric objects (e.g. Riemannian gradient, full gradient, retraction, rank-related retraction etc.) are defined.

5.5.1 Gradients of Interest

Following Cason et al. [CAD13], the tangent space to $\mathcal{S}_r(m,n)$ at a point $S = UD_rV^T \in \mathcal{S}_r(m,n)$ is

$$\begin{aligned} \mathrm{T}_S\mathcal{S}_r(m,n) :&= \left\{ \begin{array}{l} UAV^T + UBV_\perp^T + U_\perp CV^T : \\ B, C \text{ arbitrary}, \ \mathrm{tr}(AD_r) = 0 \end{array} \right\} \\ &= \left\{ \begin{array}{l} \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} A & B \\ C & 0 \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix} : \\ B, C \text{ arbitrary}, \mathrm{tr}(AD_r) = 0 \end{array} \right\}, \end{aligned} \tag{5.59}$$

where $U_\perp, V_\perp$ are any orthogonal complements of U, V. There is an extra condition on$A$ such that $\mathrm{tr}(AD_r) = 0$, due to the requirement on matrix D_r, i.e., it satisfies $\|D_r\|_F = 1$.

The normal space to $\mathcal{S}_r(m,n)$ at the point $S = UD_rV^T \in \mathcal{S}_r(m,n)$ is

$$\begin{aligned} \mathrm{N}_S\mathcal{S}_r(m,n) :&= \left\{ \begin{bmatrix} U & U_\perp \end{bmatrix} \begin{bmatrix} \alpha D_r & 0 \\ 0 & R_\perp \end{bmatrix} \begin{bmatrix} V^T \\ V_\perp^T \end{bmatrix} : \right. \\ &\qquad \left. \alpha \in \mathbb{R}, R_\perp \in \mathbb{R}^{(m-r)\times(n-r)} \right\} \\ &= \left\{ \alpha S + U_\perp R_\perp V_\perp^T : \right. \\ &\qquad \left. \alpha \in \mathbb{R}, R_\perp \in \mathbb{R}^{(m-r)\times(n-r)} \right\}. \end{aligned} \tag{5.60}$$

By restricting the Euclidean inner product on $\mathbb{R}^{m\times n}$,

$$\langle A, B\rangle = \mathrm{tr}(A^TB) \ \text{ with } A, B \in \mathbb{R}^{m\times n},$$

to the tangent space, we turn $\mathcal{S}_r(m,n)$ into a Riemannian manifold with Riemannian metric

$$g_S(\xi,\eta) := \langle \xi, \eta\rangle = \mathrm{tr}(\xi^T\eta) \ \text{ with } S \in \mathcal{S}_k(m,n) \text{ and } \xi, \eta \in \mathrm{T}_S\mathcal{S}_k(m,n), \tag{5.61}$$

where the tangent vectors ξ, η are seen as matrices in $\mathbb{R}^{m\times n}$.

Once the metric is defined, the Riemannian gradient can be determined which in turn requires projection. The orthogonal projection onto the tangent space and the normal space at $S = UD_rV^T$ are

$$\begin{aligned} P_{\mathrm{T}_S\mathcal{S}_r(m,n)} &: \mathbb{R}^{m\times n} \to \mathrm{T}_S\mathcal{S}_r(m,n) \\ Z \to P_S Z &= UU^TZVV^T - \alpha S + UU^TZV_\perp V_\perp^T + U_\perp U_\perp^T ZVV^T \\ &= UU^TZ + ZVV^T - UU^TZVV^T - \alpha S, \end{aligned} \tag{5.62}$$

$$\begin{aligned} P_{\mathrm{N}_S\mathcal{S}_r(m,n)} &: \mathbb{R}^{m\times n} \to \mathrm{N}_S\mathcal{S}_r(m,n) \\ Z \to P_S^\perp Z &= \alpha S + (I_m - UU^T)Z(I_n - VV^T). \end{aligned} \tag{5.63}$$

In order to get the explicit form of projection, an explicit expression of α is needed. We have any $Z \in \mathcal{R}^{m\times n}$ can be rewritten into the following form

$$Z = UKV^T + UBV_\perp^T + U_\perp CV^T + U_\perp EV_\perp, \tag{5.64}$$

where $K \in \mathbb{R}^{r\times r}, B \in \mathbb{R}^{r\times(n-r)}, C \in \mathbb{R}^{(m-r)\times r}, E \in \mathbb{R}^{(m-r)\times(n-r)}$. Projecting Z onto the tangent and normal spaces, yields

$$Z = P_S Z + P_S^\perp Z. \tag{5.65}$$

From the two terms in the expression for Z, the tangent and normal spaces are seen to be $\mathrm{T}_S\mathcal{S}_r = \{UAV^T + UBV_\perp^T + U_\perp CV^T\}$ and $\mathrm{N}_S\mathcal{S}_r = \{\alpha S + U_\perp R_\perp V_\perp^T\}$, where $B, C, R_\perp$ are arbitrary matrices and A is any matrix that satisfies $\mathrm{tr}(AD_r) = 0$.

To get a computationally useful form of elements of these spaces, suitable expressions for A and α are required. Since $\alpha S = \alpha U D_r V^T$, we seek A in the form $A = K - \alpha D_r$. Multiplying Z by U^T from the left and V from the right, eliminates the last three terms and gives $K = U^T Z V$. Thus $A = U^T Z V - \alpha D_r$. Combining this with the constraint $\mathrm{tr}(AD_r) = 0$, we get $\mathrm{tr}((K - \alpha D_r)D_r) = 0$. Therefore, the form of α can be obtained:

$$\alpha = \frac{\mathrm{tr}(KD_r)}{\mathrm{tr}(D_r D_r)} = \mathrm{tr}(U_r^T Z V_r D_r) = \mathrm{tr}(Z V_r D_r U_r^T) = \mathrm{tr}(Z(U_r^T D_r V_r^T)^T) = \mathrm{tr}(ZS^T). \tag{5.66}$$

Given the explicit form of α, we obtain the formula for projection onto tangent and normal space as follows:

$$\begin{aligned} P_{\mathrm{T}_S\mathcal{S}_r(m,n)} &: \mathbb{R}^{m\times n} \to \mathrm{T}_S\mathcal{S}_r(m,n) \\ Z \to P_S Z &= UU^T ZVV^T - \mathrm{tr}(ZS^T)S + UU^T ZV_\perp V_\perp^T + U_\perp U_\perp^T ZVV^T \\ &= UU^T Z + ZVV^T - UU^T ZVV^T - \mathrm{tr}(ZS^T)S, \\ P_{\mathrm{N}_S\mathcal{S}_k(m,n)} &: \mathbb{R}^{m\times n} \to \mathrm{N}_S\mathcal{S}_k(m,n) \end{aligned} \tag{5.67}$$

$$Z \to P_S^\perp Z = \mathrm{tr}(ZS^T)S + (I_m - UU^T)Z(I_n - VV^T). \tag{5.68}$$

Since the Euclidean gradient of cost function $\Phi(S)$ is $2\mathcal{M}^2(S)$, projecting the gradient onto tangent space, we obtain the Riemannian gradient

$$\begin{aligned} \mathrm{grad}\Phi(S) &:= P_S 2\mathcal{M}^2(S) = 2P_S\mathcal{M}^2(S) \\ &= 2UU^T\mathcal{M}^2(S) + 2\mathcal{M}^2(S)VV^T - 2UU^T\mathcal{M}^2(S)VV^T - 2\mathrm{tr}(\mathcal{M}^2(S)S^T)S, \end{aligned} \tag{5.69}$$

where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M}(S))$ and $[\mathcal{M}(S)]_{ij} = [ASB^T + A^T SB]_{ij}$.

In order to apply MROM, a Riemannian submanifold $\mathcal{M}$ is needed such that the cost function can be extended. Since for each $S = UDV^T \in \mathcal{S}_{\leq k}(m,n)$, we require $\|D\|_F = 1$, which implies $\|S\|_F = 1$, the submanifold $\mathcal{M}$ can be treated as a unit sphere S^{mn-1}. Consider the following two functions Φ_F and Φ_r:

$$\begin{aligned} &\Phi_{\mathrm{F}} : \mathcal{M} = S^{mn-1} \to \mathbb{R} : S \mapsto \mathrm{tr}(S^T\mathcal{M}^2(S)), \\ &\Phi_r : \mathcal{S}_r(m,n) \to \mathbb{R} : S \mapsto \mathrm{tr}(S^T\mathcal{M}^2(S)). \end{aligned}$$

The cost function for the rank inequality constrained problem is then $\Phi = \Phi_{\mathrm{F}}|_{\mathcal{S}_{\leq k}}$ and $\Phi_r = \Phi_{\mathrm{F}}|_{\mathcal{S}_r} = \Phi|_{\mathcal{S}_r}$.

The tangent space and normal space to a sphere S^{mn-1} at a point $S \in S^{mn-1}$ are given in [AMS08, Example 3.6.1]:

$$\mathrm{T}_S S^{mn-1} = \{Z \in \mathbb{R}^{m\times n} : S^T Z = 0\}, \tag{5.70}$$

$$\mathrm{N}_S S^{mn-1} = \{\alpha S : \alpha \in \mathbb{R}\}, \tag{5.71}$$

and the projections are

$$\begin{aligned} P_{\mathrm{T}_S S^{mn-1}} :&\mathbb{R}^{m\times n} \to \mathrm{T}_S S^{mn-1} \\ &Z \mapsto P_S Z = Z - \alpha S, \end{aligned} \tag{5.72}$$

$$\begin{aligned} P_{\mathrm{N}_S S^{mn-1}} :&\mathbb{R}^{m\times n} \to \mathrm{N}_S S^{mn-1} \\ &Z \mapsto P_S^{\perp} Z = \alpha S, \end{aligned} \tag{5.73}$$

where $\alpha = \mathrm{tr}(ZS^T)$. Thus, the full gradient on the submanifold $\mathcal{M}$ can be obtained by projecting the Euclidean gradient of cost function $\Phi_{\mathrm{F}}(S)$ onto the tangent space $\mathrm{T}_S\mathcal{M}$

$$\mathrm{grad}\Phi_{\mathrm{F}}(S) := 2P_S\mathcal{M}^2(S) = 2\mathcal{M}^2(S) - 2\mathrm{tr}(\mathcal{M}^2(S)S^T)S, \tag{5.74}$$

where $\mathcal{M}^2(S) = \mathcal{M}(\mathcal{M})$ and $[\mathcal{M}(S)]_{ij} = [ASB^T + A^T SB]_{ij}$.

5.5.2 Retractions of Interest

Two kinds of retractions are required for MROM: retraction onto the fixed-rank manifolds and rank-related retractions.

Given the triple (U, D, V) such that $S = UDV^T$, the computation of the triple $(\dot{U}, \dot{D}, \dot{V})$ is similar to the discussions in Section 4.3.4. For $\dot{U}$ and $\dot{V}$, they are the same as shown in Section 4.3.4. However, there is an additional restriction on D such that $\|D\|_F = 1$. Thus, $\dot{D}$ is computed as

$$\dot{D} = U^T \dot{S} V - \alpha D = U^T \dot{S} V - \mathrm{tr}(\dot{S}S^T)D. \tag{5.75}$$

The three choices of retractions on the fixed-rank manifold $\mathcal{S}_r$ are considered in the following:

- **three-factor SVD-type retraction**:

$$R_S(\dot{S}) = U_+ D_+ V_+^T \tag{5.76}$$

where

$$\begin{aligned}
&\dot{U}D = Q_u R_u, \\
&\dot{V}D = Q_v R_v, \\
&U_s D_s V_s = \begin{bmatrix} D + \dot{D} & R_v^T \\ R_u & 0 \end{bmatrix}, \\
&U_+ = \begin{bmatrix} U & Q_u \end{bmatrix} U_s(:, 1:r), \\
&D_+ = \frac{D_s(1:r, 1:r)}{\|D_s(1:r, 1:r)\|_F}, \\
&V_+ = \begin{bmatrix} V & Q_v \end{bmatrix} V_s(:, 1:r),
\end{aligned}$$

- **three-factor polar-type retraction**

$$R_S(\dot{S}) = U_+ D_+ V_+^T \tag{5.77}$$

where

$$\begin{aligned}
&U_s D_s V_s^T = D + \dot{D} \quad \text{using SVD}, \\
&U_+ = uf(U + \dot{U})U_s, \\
&D_+ = \frac{D_s}{\|D_s\|_F}, \\
&V_+ = uf(V + \dot{V})V_s
\end{aligned}$$

and $uf(\cdot)$ denotes the orthogonal component of the polar decomposition.

- **three-factor QR-type retraction I**

$$R_S(\dot{S}) = U_+ D_+ V_+^T. \tag{5.78}$$

where

$$\begin{aligned}
&U_s D_s V_s^T = D + \dot{D} \quad \text{using SVD}, \\
&U_+ = qf(U + \dot{U})U_s, \\
&D_+ = \frac{D_s}{\|D_s\|_F}, \\
&V_+ = qf(V + \dot{V})V_s,
\end{aligned}$$

and $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument.

- **three-factor QR-type retraction II**

$$R_S(\dot{S}) = U_+ D_+ V_+^T. \tag{5.79}$$

where

$$U_+ = qf(U + \dot{U}),$$
$$D_+ = \frac{D + \dot{D}}{\|D + \dot{D}\|_F},$$
$$V_+ = qf(V + \dot{V}),$$

and $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument.

In order to apply Algorithm 2, a rank-related retraction is also needed. It must satisfy certain properties as in Definition 8 in Chapter 3. Given $S \in \mathcal{S}_r$, based on Definition 8 in Chapter 3, the following three types of rank-related retractions are constructed:

- **SVD-type rank-related retraction**

$$\tilde{R}_S(\eta^*) = U_{\tilde{r}} \hat{D}_{\tilde{r}} V_{\tilde{r}}^T, \tag{5.80}$$

where $[U, D, V] = \text{svd}(S + \eta^*)$ is (ordered) singular value decomposition (SVD), $U_{\tilde{r}}, V_{\tilde{r}}$ are first $\tilde{r}$ columns of U, V respectively, $D_{\tilde{r}}$ are the upper $\tilde{r}$ by $\tilde{r}$ block of matrix D, $\hat{D}_{\tilde{r}} = \frac{D_{\tilde{r}}}{\|D_{\tilde{r}}\|_F}$. This can be computed more efficiently by

$$\tilde{R}_S(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+^T, \tag{5.81}$$

where

$$\begin{aligned}
&Q_u R_u = \tilde{U}_p, \\
&Q_v R_v = \tilde{V}_p, \\
&U_s D_s V_s = \begin{bmatrix} D_{\tilde{r}} + \dot{D}_{\tilde{r}} & R_v^T \\ R_u & 0 \end{bmatrix}, \\
&\tilde{U}_+ = \begin{bmatrix} U_{\tilde{r}} & Q_u \end{bmatrix} U_S(:, 1:\tilde{r}), \\
&\tilde{D}_+ = \frac{D_s(1:\tilde{r}, 1:\tilde{r})}{\|D_s(1:\tilde{r}, 1:\tilde{r})\|_F}, \\
&\tilde{V}_+ = \begin{bmatrix} V_{\tilde{r}} & Q_v \end{bmatrix} V_S(:, 1:\tilde{r}),
\end{aligned}$$

and $\tilde{r} = r + \Delta r$, $U_{\tilde{r}} = \begin{bmatrix} U_r & U_{\Delta r} \end{bmatrix}$, $D_{\tilde{r}} = \begin{bmatrix} D_r & 0^{r \times \Delta r} \\ 0^{\Delta r \times r} & 0^{\Delta r \times \Delta r} \end{bmatrix}$, $V_{\tilde{r}} = \begin{bmatrix} V_r & V_{\Delta r} \end{bmatrix}$.

- **Polar-type rank-related retraction**

$$\tilde{R}_S(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+^T, \tag{5.82}$$

where

$$\begin{aligned}
\tilde{U}_+ &= uf(U_{\tilde{r}} + \dot{U}_{\tilde{r}})U_s, \\
\tilde{D}_+ &= \frac{D_s}{\|D_s\|_F}, \\
\tilde{V}_+ &= uf(V_{\tilde{r}} + \dot{V}_{\tilde{r}})V_s, \\
D_{\tilde{r}} + \dot{D}_{\tilde{r}} &= U_s D_s V_s^T,
\end{aligned}$$

where $uf(\cdot)$ denotes the orthogonal component of the polar decomposition.

- **QR-type rank-related retraction I**

$$\tilde{R}_S(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+^T, \tag{5.83}$$

where

$$\begin{aligned}
\tilde{U}_+ &= qf(U_{\tilde{r}} + \dot{U}_{\tilde{r}})U_s, \\
\tilde{D}_+ &= \frac{D_s}{\|D_s\|_F}, \\
\tilde{V}_+ &= qf(V_{\tilde{r}} + \dot{V}_{\tilde{r}})V_s, \\
D_{\tilde{r}} + \dot{D}_{\tilde{r}} &= U_s D_s V_s^T,
\end{aligned}$$

where $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument.

For the three types of rank-related retraction above, the rank-related vector η^* has the form of $\eta^* = \dot{U}_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} \dot{D}_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} \dot{V}_{\tilde{r}}$ and $\dot{D} = U^T \dot{S} V - \mathrm{tr}(\dot{S} S^T) D$. If $\dot{D}_{\tilde{r}}$ is assumed to be a diagonal matrix and $\eta^* = \dot{U}_{\tilde{r}} D_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} \dot{D}_{\tilde{r}} V_{\tilde{r}}^T + U_{\tilde{r}} D_{\tilde{r}} \dot{V}_{\tilde{r}}$, then similar to Section 4.3.5, we have the **QR-type rank-related retraction II**:

$$\tilde{R}_X(\eta^*) = \tilde{U}_+ \tilde{D}_+ \tilde{V}_+^T, \tag{5.84}$$

where

$$\begin{aligned}
\tilde{U}_+ &= qf(U_{\tilde{r}} + \dot{U}_{\tilde{r}}), \\
\tilde{D}_+ &= \frac{D_{\tilde{r}} + \dot{D}_{\tilde{r}}}{\|D_{\tilde{r}} + \dot{D}_{\tilde{r}}\|_F}, \\
\tilde{V}_+ &= qf(V_{\tilde{r}} + \dot{V}_{\tilde{r}}),
\end{aligned}$$

where $qf(\cdot)$ denotes the Q-factor of the thin QR decomposition of its matrix argument.

5.5.3 Vector Transport

In our framework, vector transport can be represented by an m by n matrix. Given two points S_1 and S_2 in $\mathcal{S}_k(m, n)$, the corresponding tangent spaces are T_{S_1}, T_{S_2}. We choose the isometric

vector transport $\mathcal{T}$ from S_1 to S_2 as the direct rotation from $\mathrm{T}_{\mathrm{S}_1}\mathcal{S}_k(m,n)$ to $\mathrm{T}_{\mathrm{S}_2}\mathcal{S}_k(m,n)$, restricted to act on $\mathrm{T}_{\mathrm{S}_1}\mathcal{S}_k(m,n)$. Note that the tangent space has the following structure

$$\mathrm{T}_S\mathcal{S}_k(m,n) = \left\{ \begin{array}{r} UAV^T + UBV_\perp^T + U_\perp CV^T : \\ B, C \text{ arbitrary, } \mathrm{tr}(AD_k) = 0, \end{array} \right\}.$$

An orthonormal basis of $\mathrm{T}_S\mathcal{S}_k(m,n)$, denoted by B_S, is given by

$$\begin{aligned} &\{U(e_i e_j^T)V : i = 1, \cdots, k, j = 1, \cdots, k, i, j \text{ can not both equal to k}\} \\ &\cup \{U(e_j \tilde{e}_i^T)V_\perp^T : i = 1, \cdots, n-k, j = 1, \cdots, k\} \\ &\cup \{U_\perp(\hat{e}_i e_j^T)V^T : i = 1 \cdots, m-k, j = 1, \cdots, k\} \end{aligned} \tag{5.85}$$

where $(e_1, \cdots, e_k)$ is the canonical basis of $\mathbb{R}^k$, $(\hat{e}_1, \cdots, \hat{e}_{m-k})$ is the canonical basis of $\mathbb{R}^{m-k}$ and $(\tilde{e}_1, \cdots, \tilde{e}_{n-k})$ is the canonical basis of $\mathbb{R}^{n-k}$. Similarly, we can construct the basis for normal space

$$\mathrm{N}_S\mathcal{S}_k(m,n) = \left\{ \begin{array}{r} \alpha S + U_\perp R_\perp V_\perp^T : \\ \alpha \in \mathbb{R}, R_\perp \in \mathbb{R}^{(m-k)\times(n-k)} \end{array} \right\}.$$

Using N_S to denote, which is given by

$$\{UDV^T\} \cup \{U_\perp \tilde{e}_i \hat{e}_j V_\perp^T : i = 1, \cdots, m-k, j = 1, \cdots, n-k\}. \tag{5.86}$$

The columns of B_S and N_S are thus chosen as the "vec" of the basis elements.

We can also derive the vector transport by the differentiated retraction of (5.78) and (5.79).

Proposition 34. *Let $S = UD_kV^T \in \mathcal{S}_k(m,n)$, $\xi, \eta \in \mathrm{T}_S\mathcal{S}_k(m,n)$. Assuming ξ and η have the following structure*

$$\begin{aligned} \xi &= \dot{U}_1 D_k V^T + U\dot{D}_1 V^T + UD_k\dot{V}_1^T, \\ \eta &= \dot{U}_2 D_k V^T + U\dot{D}_2 V^T + UD_k\dot{V}_2^T. \end{aligned}$$

Then the vector transport by the differentiated retraction of (5.78) *is*

$$\begin{aligned} \mathcal{T}_\eta\xi = {} & \mathcal{T}_{\dot{U}_2}(\dot{U}_1)\frac{D_k + \dot{D}_2}{\|D_k + \dot{D}_2\|_F} qf(V + \dot{V}_2)^T + qf(U + \dot{U}_2)\frac{D_k + \dot{D}_2}{\|D_k + \dot{D}_2\|_F}(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T \\ & + qf(U + \dot{U}_2))\left(\frac{\dot{D}_1}{\|D_k + \dot{D}_2\|_F} - \frac{D_k + \dot{D}_2}{\|D_k + \dot{D}_2\|_F^2}\frac{(D_k + \dot{D}_2)^T\dot{D}_1}{\|D_k + \dot{D}_2\|_F}\right) qf(V + \dot{V}_2)^T, \end{aligned} \tag{5.87}$$

where $\mathcal{T}_{\dot{U}_2}(\dot{U}_1)$ is a differentiated retraction on Stiefel manifold [AMS08, Example 8.1.5] and $qf(\cdot)$ denotes the Q factor of the QR decomposition with nonnegative elements on the diagonal of R.

Proof. Based on the definition of the vector transport by differentiated retraction and the QR-type I retraction (5.78), we have

$$\begin{aligned}
\mathcal{T}_\eta \xi &= \frac{d}{dt} R_X(\eta + t\xi)\Big|_{t=0} \\
&= \frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)\left(\frac{D_k + \dot{D}_2 + t\dot{D}_1}{\|D_k + \dot{D}_2 + t\dot{D}_1\|_F}\right) qf(V + \dot{V}_2 + t\dot{V}_1)^T]\Big|_{t=0} \\
&= \frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)]\left(\frac{D_k + \dot{D}_2 + t\dot{D}_1}{\|D_k + \dot{D}_2 + t\dot{D}_1\|_F}\right) qf(V + \dot{V}_2 + t\dot{V}_1)^T\Big|_{t=0} \\
&+ qf(U + \dot{U}_2 + t\dot{U}_1)\frac{d}{dt}\left[\left(\frac{D_k + \dot{D}_2 + t\dot{D}_1}{\|D_k + \dot{D}_2 + t\dot{D}_1\|_F}\right)\right] qf(V + \dot{V}_2 + t\dot{V}_1)^T\Big|_{t=0} \\
&+ qf(U + \dot{U}_2 + t\dot{U}_1)\left(\frac{D_k + \dot{D}_2}{\|D_k + \dot{D}_2\|_F} + t\dot{D}_1\right)\frac{d}{dt}[qf(V + \dot{V}_2 + t\dot{V}_1)]^T\Big|_{t=0}.
\end{aligned} \tag{5.88}$$

Since $U \in \mathrm{St}(m,r), V \in \mathrm{St}(n,r)$, according to the vector transport by differentiated retraction on the Stiefel manifold [AMS08], we have for $\dot{U}_1, \dot{U}_2 \in \mathrm{T}_U\mathrm{St}(m,r)$,

$$\frac{d}{dt}[qf(U + \dot{U}_2 + t\dot{U}_1)]|_{t=0} = \mathcal{T}_{\dot{U}_2}(\dot{U}_1), \tag{5.89}$$

and for $\dot{V}_1, \dot{V}_2 \in \mathrm{T}_V\mathrm{St}(n,r)$,

$$\frac{d}{dt}[qf(V + \dot{V}_2 + t\dot{V}_1)]|_{t=0} = \mathcal{T}_{\dot{V}_2}(\dot{V}_1). \tag{5.90}$$

where

$$\begin{aligned}
T_{\dot{U}_2}(\dot{U}_1) &= \mathrm{D}R_U(\dot{U}_2)[\dot{U}_1] \\
&= \mathrm{D}qf(U + \dot{U}_2)[\dot{U}_1] \\
&= R_U(\dot{U}_2)\rho_{\mathrm{skew}}(R_U(\dot{U}_2)^T\dot{U}_1(R_U(\dot{U}_2)^T(U + \dot{U}_2))^{-1}) \\
&+ (I - R_U(\dot{U}_2)R_U(\dot{U}_2)^T)\dot{U}_1(R_U(\dot{U}_2)^T(U + \dot{U}_2))^{-1},
\end{aligned}$$

and $\rho_{\mathrm{skew}}(B)$ denotes the skew-symmetric term of the decomposition of a square matrix B into the sum of a skew-symmetric term and an upper triangular term, i.e.,

$$(\rho_{\mathrm{skew}}(B))_{i,j} = \begin{cases} B_{i,j} & \text{if } i >, j \\ 0 & \text{if } i = j, \\ -B_{j,i} & \text{if } i < j. \end{cases}$$

The derivative $\frac{d}{dt}\left(\frac{D_k+\dot{D}_2+t\dot{D}_1}{\|D_k+\dot{D}_2+t\dot{D}_1\|_F}\right)\Big|_{t=0}$ is computed as follows:

$$\begin{aligned}
&\frac{d}{dt}\left(\frac{D_k+\dot{D}_2+t\dot{D}_1}{\|D_k+\dot{D}_2+t\dot{D}_1\|_F}\right)\Big|_{t=0}\\
&=\frac{\frac{d}{dt}(D_k+\dot{D}_2+t\dot{D}_1)\|D_k+\dot{D}_2+t\dot{D}_1\|_F-(D_k+\dot{D}_2+t\dot{D}_1)\frac{d}{dt}(\|D_k+\dot{D}_2+t\dot{D}_1\|_F)}{\|D_k+\dot{D}_2+t\dot{D}_1\|_F^2}\Big|_{t=0}\\
&=\frac{\dot{D}_1}{\|D_k+\dot{D}_2\|_F}-\frac{D_k+\dot{D}_2}{\|D_k+\dot{D}_2\|_F^2}\frac{(D_k+\dot{D}_2)^T\dot{D}_1}{\|D_k+\dot{D}_2\|_F}
\end{aligned}\tag{5.91}$$

Substituting (5.89), (5.90) and (5.91) into (5.88), we have the vector transport by differentiated retraction (5.78) are following

$$\begin{aligned}
\mathcal{T}_\eta\xi=&\mathcal{T}_{\dot{U}_2}(\dot{U}_1)\frac{D_k+\dot{D}_2}{\|D_k+\dot{D}_2\|_F}qf(V+\dot{V}_2)^T+qf(U+\dot{U}_2))\frac{D_k+\dot{D}_2}{\|D_k+\dot{D}_2\|_F}(\mathcal{T}_{\dot{V}_2}(\dot{V}_1))^T\\
&+qf(U+\dot{U}_2))\left(\frac{\dot{D}_1}{\|D_k+\dot{D}_2\|_F}-\frac{D_k+\dot{D}_2}{\|D_k+\dot{D}_2\|_F^2}\frac{(D_k+\dot{D}_2)^T\dot{D}_1}{\|D_k+\dot{D}_2\|_F}\right)qf(V+\dot{V}_2)^T.
\end{aligned}\tag{5.92}$$

□

5.5.4 Action of the Hessian on Fixed-rank Manifold

Proposition 35. *For any $S=UD_kV^T\in\mathcal{S}_k(m,n)$ and $\eta\in \mathrm{T}_S\mathcal{S}_k(m,n)$, the Riemannian Hessian of Φ at S in the direction of η satisfies*

$$\mathrm{Hess}\Phi(S)[\eta]=\bigtriangledown_\eta\mathrm{grad}\Phi(S)=P_S(D\mathrm{grad}\Phi(S)[\eta]),$$

where

$$\begin{aligned}
D\mathrm{grad}\Phi(S)[\eta]=&2[U_\perp(U_\perp^T\eta VD_k^{-1})U^T+U(U_\perp^T\eta VD_k^{-1})^TU_\perp^T]\mathcal{M}^2(S)+2(UU^T)\mathcal{M}^2(\eta)\\
&+2\mathcal{M}^2(\eta)VV^T+2\mathcal{M}^2(S)[V_\perp(D_k^{-1}U^T\eta V_\perp)^TV^T+V(D_k^{-1}U^T\eta V_\perp)V_\perp^T]\\
&-2[U_\perp(U_\perp^T\eta VD_k^{-1})U^T+U(U_\perp^T\eta VD_k^{-1})^TU_\perp^T]\mathcal{M}^2(S)VV^T-2UU^T\mathcal{M}^2(\eta)VV^T\\
&-2UU^T\mathcal{M}^2(S)[V_\perp(D_k^{-1}U^T\eta V_\perp)^TV^T+V(D_k^{-1}U^T\eta V_\perp)V_\perp^T]\\
&-2\mathrm{tr}(\mathcal{M}^2(\eta)S^T)S-2\mathrm{tr}(\mathcal{M}^2(S)\eta^T)S-2\mathrm{tr}(\mathcal{M}^2(S)S^T)\eta.
\end{aligned}$$

Proof. As the Riemannian gradient of $\Phi(S)$ at a point $S=UD_kV^T$ is

$$\begin{aligned}
\mathrm{grad}\Phi(S)=&P_S2\mathcal{M}^2(S)=2P_S\mathcal{M}^2(S)\\
=&2UU^T\mathcal{M}^2(S)+2\mathcal{M}^2(S)VV^T-2UU^T\mathcal{M}^2(S)VV^T-2\mathrm{tr}(\mathcal{M}^2(S)S^T)S.
\end{aligned}\tag{5.93}$$

In order to find the Hessian, we need the derivative of UU^T and the derivative of VV^T. Since $U \in \mathrm{St}(m,k)$, its derivative has the form $\dot{U} = U\Omega_1 + U_\perp K_1$, where Ω_1 is a skew matrix, $K_1 \in \mathbb{R}^{(m-k)\times k}$. Similarly for $V \in \mathrm{St}(n,k)$, the derivative of V has the form $\dot{V} = V\Omega_2 + V_\perp K_2$, where Ω_2 is a skew matrix, $K_2 \in \mathbb{R}^{(n-k)\times k}$. Thus, for any $S = UD_kV^T \in \mathcal{S}_k(m,n)$, its derivative can be written into the following form:

$$\dot{S} = UAV^T + U_\perp K_1 D_k V^T + UD_k K_2^T V_\perp^T, \tag{5.94}$$

where $A \in \mathbb{R}^{k\times k}, K_1 \in \mathbb{R}^{(m-k)\times k}, K_2 \in \mathbb{R}^{(n-k)\times k}$, $\mathrm{tr}(AD_k) = 0$. Multiplying $\dot{S}$ by $U_\perp^T$ from the left and V from the right, we get

$$K_1 = U_\perp^T \dot{S} V D_k^{-1}. \tag{5.95}$$

Similarly, we can get

$$K_2^T = D_k^{-1} U^T \dot{S} V_\perp. \tag{5.96}$$

The derivative of UU^T is computed as

$$\begin{aligned}(UU^T)' &= \dot{U}U^T + U\dot{U}^T = (U\Omega_1 + U_\perp K_1)U^T + U(U\Omega_1 + U_\perp K_1)^T \\ &= U(\Omega_1 + \Omega_1^T)U^T + U_\perp K_1 U^T + UK_1^T U_\perp^T \\ &= U_\perp K_1 U^T + UK_1^T U_\perp^T \\ &= U_\perp (U_\perp^T \dot{S} V D_k^{-1})U^T + U(U_\perp^T \dot{S} V D_k^{-1})^T U_\perp^T.\end{aligned} \tag{5.97}$$

Similarly, we can obtain the derivative of VV^T:

$$(VV^T)' = V_\perp (D_k^{-1} U^T \dot{S} V_\perp)^T V^T + V(D_k^{-1} U^T \dot{S} V_\perp) V_\perp^T. \tag{5.98}$$

Therefore, the directional derivative of $\mathrm{grad}\Phi$ at S along η is

$$\begin{aligned}D\mathrm{grad}\Phi(S)[\eta] &= 2(UU^T)'\mathcal{M}^2(S) + 2(UU^T)\mathcal{M}^2(\eta) + 2\mathcal{M}^2(\eta)VV^T + 2\mathcal{M}^2(S)(VV^T)' \\ &\quad - 2(UU^T)'\mathcal{M}^2(S)VV^T - 2UU^T\mathcal{M}^2(\eta)VV^T - 2UU^T\mathcal{M}^2(S)(VV^T)' \\ &\quad - 2\mathrm{tr}(\mathcal{M}^2(\eta)S^T)S - 2\mathrm{tr}(\mathcal{M}^2(S)\eta^T)S - 2\mathrm{tr}(\mathcal{M}^2(S)S^T)\eta \\ &= 2[U_\perp(U_\perp^T\eta V D_k^{-1})U^T + U(U_\perp^T\eta V D_k^{-1})^T U_\perp^T]\mathcal{M}^2(S) + 2(UU^T)\mathcal{M}^2(\eta) \\ &\quad + 2\mathcal{M}^2(\eta)VV^T + 2\mathcal{M}^2(S)[V_\perp(D_k^{-1}U^T\eta V_\perp)^T V^T + V(D_k^{-1}U^T\eta V_\perp)V_\perp^T] \\ &\quad - 2[U_\perp(U_\perp^T\eta V D_k^{-1})U^T + U(U_\perp^T\eta V D_k^{-1})^T U_\perp^T]\mathcal{M}^2(S)VV^T - 2UU^T\mathcal{M}^2(\eta)VV^T \\ &\quad - 2UU^T\mathcal{M}^2(S)[V_\perp(D_k^{-1}U^T\eta V_\perp)^T V^T + V(D_k^{-1}U^T\eta V_\perp)V_\perp^T] \\ &\quad - 2\mathrm{tr}(\mathcal{M}^2(\eta)S^T)S - 2\mathrm{tr}(\mathcal{M}^2(S)\eta^T)S - 2\mathrm{tr}(\mathcal{M}^2(S)S^T)\eta.\end{aligned} \tag{5.99}$$

Finally, the Riemannian Hessian of Φ at S in the direction of η can be computed by

$$\text{Hess}\Phi(S)[\eta] = \bigtriangledown_\eta \text{grad}\Phi(S) = P_S(D\text{grad}\Phi(S)[\eta]).$$

□

5.5.5 Some Observations of Cason's Algorithm

To reduce the complexity, Cason et al. do not use Algorithm 7, i.e., they do not work with $S \in \mathbb{R}^{m\times n}$ itself but with its singular value decomposition $(U, D, V) \in \mathbb{R}^{m\times k} \times \text{Diag}(k, k, k) \times \mathbb{R}^{n\times k}$, where $\text{Diag}(k, k, k) := \{D \in \mathbb{R}^{k\times k} : D \text{ diagonal}, D_{ii} = 0 \text{ for all } i > k\}$. Similarly, in practice, they do not compute $\mathcal{M}^2(S) \in \mathbb{R}^{m\times n}$ itself but its singular value decomposition. Algorithm 7 is rewritten as following

Algorithm 8 Cason's Algorithm 3

Require: Graph G_A and G_B respectively of order m and n

1: $(U^0, D^0, V^0) \leftarrow SVD_k(\mathbf{1}/\|\mathbf{1}\|_F)$
2: **for** $t = 1, 2, \cdots, t_{max}$ **do**
3: $\quad U' \leftarrow \left[AU^{t-1}D^{t-1}, \quad A^TU^{t-1}D^{t-1}\right] \in \mathbb{R}^{m\times 2k}$;
4: $\quad U'' \leftarrow \left[AU', \quad A^TU'\right] \in \mathbb{R}^{m\times 4k}$;
5: $\quad V' \leftarrow \left[BV^{t-1}, \quad B^TV^{t-1}\right] \in \mathbb{R}^{n\times 2k}$;
6: $\quad V'' \leftarrow \left[BV', \quad B^TV'\right] \in \mathbb{R}^{n\times 4k}$;
7: $\quad (Q_U, R_U) \leftarrow QR(U'') \in \mathbb{R}^{m\times 4k} \times \mathbb{R}^{4k\times 4k}$;
8: $\quad (Q_V, R_V) \leftarrow QR(V'') \in \mathbb{R}^{n\times 4k} \times \mathbb{R}^{4k\times 4k}$;
9: $\quad (U''', D''', V''') \leftarrow SVD_k(R_UR_V^T) \in \mathbb{R}^{m\times k} \times \mathbb{R}^{k\times k} \times \mathbb{R}^{n\times k}$;
10: $\quad (U^t, D^t, V^t) \leftarrow (Q_UU''', \frac{D'''}{\|D'''\|}, Q_VV''')$;
11: **end for**

Note that Algorithm 8 does not use any Riemannian objects. It is an update of the form

$$S_{t+1} = P(S_t + s_t(\mathcal{M}^2(S_t) - S_t)), \tag{5.100}$$

with step size $s_t = 1$ and P is a projection using the SVD and a normalization, that projects a proposed iterate onto the feasible set $\mathcal{S}_k$.

The update of the Riemannian steepest descent algorithm with line search on the fixed rank manifold $\mathcal{S}_k(m, n)$ is given by

$$S_{t+1} = R_{S_t}(-s_tP_{S_t}\nabla\Phi(S_t)), \tag{5.101}$$

where s_t is a step size, $P_{S_t}\nabla\Phi(S_t)$ is the Riemannian gradient obtained by projecting the Euclidean gradient $\nabla\Phi(S_t)$ onto the tangent space $\mathrm{T}_{S_t}\mathcal{S}_k(m,n)$, and R_{S_t} is the retraction from the tangent space of S_t to the manifold, e.g., a projection of the matrix $S_t - s_t P_{S_t}\nabla\Phi(S_t)$. Cason's iteration (5.100) is not exactly Riemannian steepest descent on $\mathcal{S}_k(m,n)$ but it is related to Riemannian steepest descent.

Comparing the two forms and reviewing the Riemannian gradient discussed in Section 5.5.1, we observe the update in (5.100) is equivalent to the update of Riemannian steepest descent method with Riemannian gradient of submanifold $\mathcal{M}$, i.e., the full gradient (5.74) on the sphere of dimension $mn-1$, and a particular step size $s_t = \frac{1}{\alpha} = \frac{1}{2\mathrm{tr}(\mathcal{M}^2(S_t)S_t^T)}$ mapped to the feasible set $\mathcal{S}_k(m,n)$ using the SVD-type retraction (5.76) discussed in Section 5.5.2.

Therefore, Cason's improved Algorithm 8 is a generalization of the well-known Euclidean gradient projection method for constrained optimization that maps every point of the line $x_i + \alpha d_i$ defined by the unconstrained line search method to the nearest point of the, typically convex, feasible set. In this case, the "line" is defined by a step of Riemannian steepest descent on the sphere retracted by simple scaling of the norm followed by a rank-k approximation of the point on the sphere and a second normalization to a point on $\mathcal{S}_k(m,n)$. Additionally, we have seen empirically that this fixed step size of Cason's in this form satisfies the Riemannian Armijo condition that is one of the line search termination criteria that is used to guarantee convergence.

5.5.6 Experiments

In this section, Algorithm 8 is used in all comparisons. Given the relationship of Algorithm 8 to a fixed step size gradient projection algorithm we compare it to the Riemannian steepest descent method on rank-k manifold first. Then, Algorithm 2 is compared to Algorithm 8. Section 5.5.6.2 shows the performance of both methods on approximating a rank-1 similarity matrix. The performance when approximating a similarity matrix for random graphs is shown in Section 5.5.6.3. Finally, in Section 5.5.6.4, the difficulties for Algorithm 8 and MROM when the two dominant eigenvalues of $M^2 = (A \otimes B + A^T \otimes B^T)^2$ are close, where A, B are adjacency matrices, are illustrated and the performance improvement of a modification to MROM is demonstrated.

The parameters in Algorithm 2 are set to be the same as they were in Section 4.4.2. The initial point in Algorithm 8 is given in Step 1. The initial point in Algorithm 2 is a rank-1 matrix defined by $[U_0, D_0, V_0]$, where $U_0 = \frac{1}{\sqrt{m}}\left[1,\cdots,1\right]^T$, $D_0 = 1$, $V_0 = \frac{1}{\sqrt{n}}\left[1,\cdots,1\right]^T$. The stopping criteria of

Algorithm 2 and Algorithm 8 are set to be the norm of final gradient on the fixed-rank manifold over the norm of initial full gradient is less than 10^{-7}.

5.5.6.1 Comparision of Approximation with k Nonzero Singular Values. In this section, the performance of Cason's method is compared with the performance of Riemannian steepest descent method (RSD) on rank-k manifold. Note that the rank-k manifold is a noncompact manifold since the limit may be not in the feasible set. The projection in Algorithm 8 is equivalent to an SVD-type retraction. In this section, the SVD-type retraction and the orthographic retraction [AO13, Section 3.2] on the fixed-rank manifold is used for RSD. Both of them are second-order retraction. Furthermore, like Algorithm 8, the step size chosen in RSD is fixed, which is $\frac{1}{2\mathrm{tr}(\mathcal{M}^2(S_k)S_k^T)}$.

The random generated graph is a directed graph with 1000 nodes and the probability of adding a new edge for each node is 0.01. We look at the performances of both methods to compute low-rank approximations of self-similarity matrices, i.e., $A = B$. The initial point in RSD is a rank-k matrix defined by $[U_0, D_0, V_0]$, where $U_0 \in \mathbb{R}^{m\times k}, V_0 \in \mathbb{R}^{n\times k}$ are orthogonal matrices generated by Matlab's ORTH and RANDN, $D_0 = \frac{D}{\|D\|_F}$, where D is a diagonal matrix with diagonal elements from 1 to k. The stopping criterion for both methods are $\|\Delta_S\|_F \leq 10^{-6}\|S\|_F$. The relative error is computed by $\frac{\|S-S^B\|_F}{\|S^B\|_F}$, where S^B is the true similarity matrix obtained from the Blondel's algorithm. The numerical rank of the true similarity matrix is small. There are 8 singular values greater than 10^{-5} and 9 of them greater than 10^{-6}.

The average computational times and the average relative errors with respect to the true self-similarity matrices for exactly k nonzero eigenvalues are shown in Figure 5.5 and Figure 5.6. From the two figures, we observe that when reaching almost the same relative error, the computational time of RSD is much smaller than Cason's method, especially when k gets larger. This shows that even for fixed k the combination of the somewhat lower computational complexity per step of RSD and its choice of direction are a clear improvement over the choices of direction and projection in Algorithm 8

Figure 5.6 shows when k reaches 8, the relative error is already small, which matches the singular values we observed in true similarity matrix. After that, increasing k makes no big difference on the relative error. This means the value k greater than 8 may be "too large" and can bring unnecessary

operations. This motivates using Algorithm 2 rather than a fixed rank k to exploit its ability to find a suitable numerical rank independent of the upper bound k.

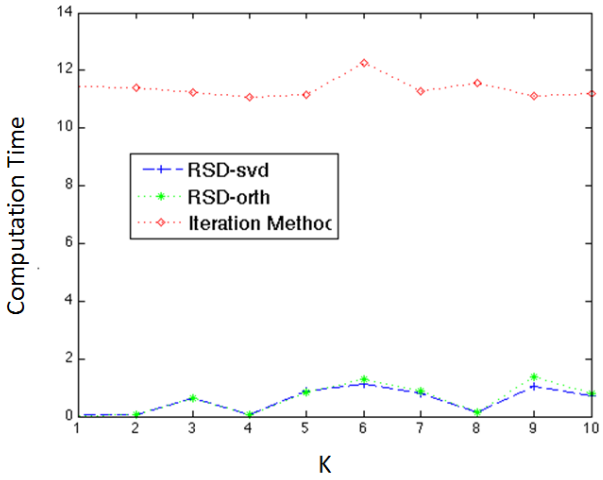

Figure 5.5: Comparison of compuational time between Riemannian Steepest Descent method and Cason's iteration method with different k.

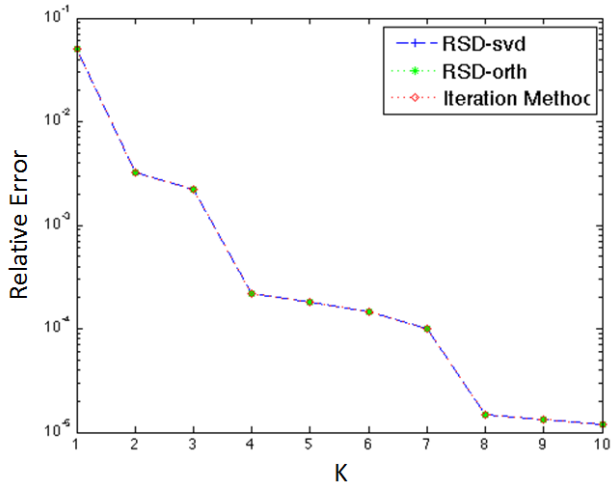

Figure 5.6: Comparison of relative error between Riemannian Steepest Descent method and Cason's iteration method with different k.

5.5.6.2 Comparison of approximation to a rank-1 similarity matrix. As mentioned in Section 5.3, when one of the graphs is symmetric, the similarity matrix is a rank-1 matrix. In the following, we compare Cason's method with MROM for different bounds k. The inner algorithm in MROM is taken to be the Riemannian steepest descent method (RSD) with fixed step size in each iteration discussed earlier.

The adjacency matrix of a random symmetric graph with N nodes is generated by $A = A_0 + A_0^T$, where A_0 is generated by Matlab's RANDINT with seed 1. The graph B is also generated by Matlab's RANDINT. Since the rank of the true similarity matrix is 1, the bound on rank is considered with two values $k = 1$ and $k = 5$. The relative error is computed by $\frac{\|S-S^B\|_F}{\|S^B\|_F}$, where S^B is true similarity matrix obtained from Blondel's algorithm. The initial point in MROM is a rank-k matrix defined by $[U_0, D_0, V_0]$, where $U_0 \in \mathbb{R}^{m\times k}, V_0 \in \mathbb{R}^{n\times k}$ are orthogonal matrices generated by Matlab's ORTH and RANDN, $D_0 = \frac{D}{\|D\|_F}$, where D is a diagonal matrix with diagonal elements from 1 to k. The initial point in Algorithm 8 is a rank-1 matrix given in Step 1 independent of the value of the bound k.

Results are shown below in Figures 5.7 and 5.8. For MROM, we observe, independently of the rank of the initial point, always adjusts the rank of the similarity matrix to the correct value of 1. Figure 5.7 shows the computational time of both methods with different k. MROM has significant time advantages as N (the size of graph) increases, independently of the initial point. The computational time of Algorithm 8 are almost the same for different k. The algorithm works with $m \times 4k$ and $m \times 2k$ matrices which since $m \gg k$ here yields mild dependence on k in complexity per step. MROM has a similar mild dependence per step. Figure 5.8 shows although the relative error achieved by both methods is small that of MROM is near numerical roundoff and noticeably smaller than that of Cason's method.

Therefore, the rank adjustment and efficient optimization on each fixed rank manifold clearly provide a significant benefit compared to Cason's method. Furthermore, figure 5.7 indicates that MROM starting from a rank-1 matrix is more efficient than starting from a higher rank matrix. In the all experiments in the remainder of this chapter, we start from a rank-1 matrix and let the rank adjust automatically using our rank control strategy.

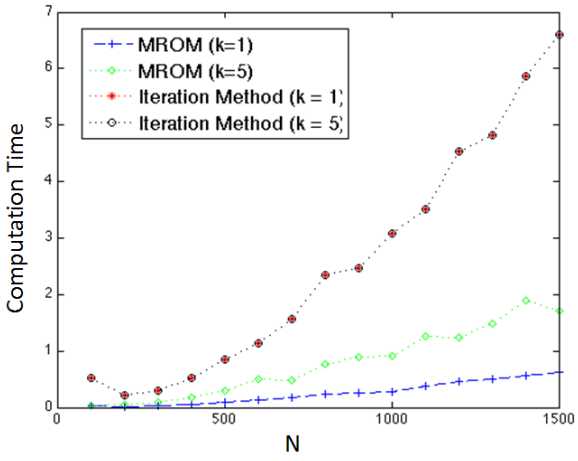

Figure 5.7: Comparison of computational time between MROM and Cason's iteration method for $k = 1$ and $k = 5$.

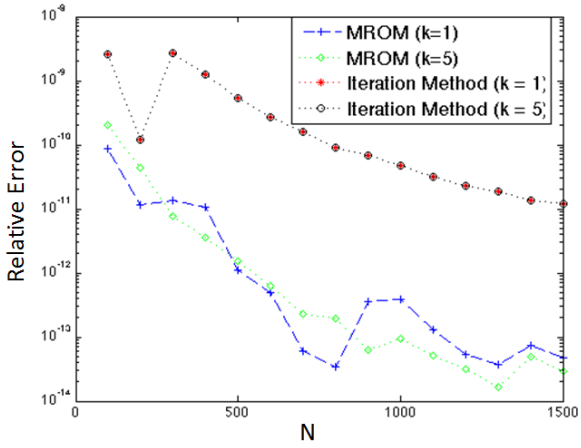

Figure 5.8: Comparison of relative error between MROM and Cason's iteration method for $k = 1$ and $k = 5$.

5.5.6.3 Comparison of approximation of a self-similarity matrix of a random graph.

In this section, the performances of MROM and Cason's method when computing a low-rank approximation of the self-similarity matrix of a randomly generated graph are compared. The random

graph is a directed graph with 500 nodes and, since the probability of adding a new edge for each node is 0.003, it has a sparse adjacency matrix. The relative error is computed by $\frac{\|S-S^B\|_F}{\|S^B\|_F}$, where S^B is true similarity matrix obtained from the Blondel's algorithm. The numerical rank of the true similarity matrix is not large compared to the size of the matrix. There are 182 singular values greater than 10^{-5} and 311 of them greater than 10^{-6}.

The comparison in Section 4.4.8 shows for matrix with large size, limited-memory RBFGS method has significant time advantage. Therefore, it is used as the inner algorithm in MROM. Since the numerical rank is large, ϵ_1 is chosen to be $\frac{\sqrt{3}}{3}$ in Algorithm 2 in order to allow a more rapid increase of rank at each step.

Figures 5.9 and 5.10 show the comparison of relative error and computational time for the two methods with different values of k. It can be observed for each k, that the relative error of both methods are almost the same, however, the computational time cost by MROM is much less than Algorithm 8. Clearly, by allowing an approximate minimization and starting with a rank-1 similarity matrix, MROM's rank adjustment and efficient Riemannian optimization on each fixed rank manifold makes it the preferred method is the numerical rank of the low-rank approximation is not very small.

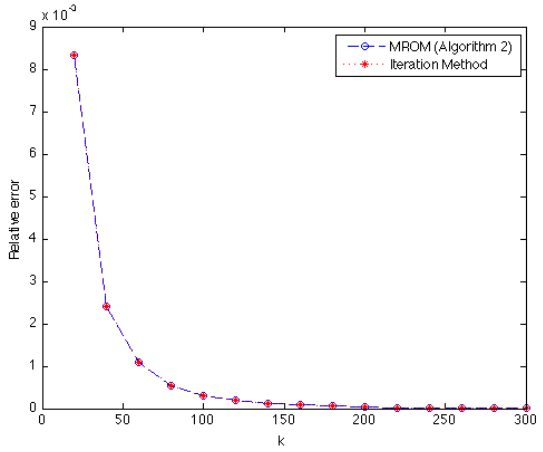

Figure 5.9: Comparison of relative error between MROM and Cason's Iteration Method on random generated graph

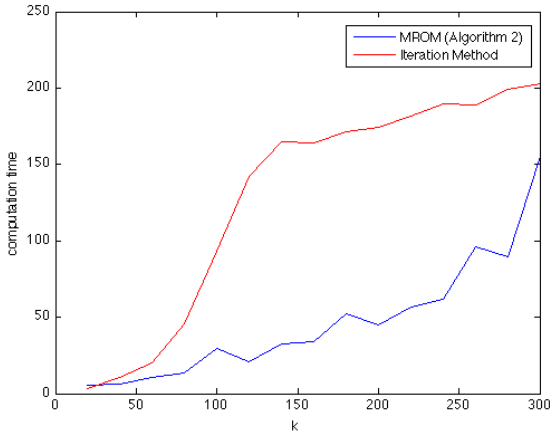

Figure 5.10: Comparison of cost time between MROM and Cason's Iteration Method on random generated graph

5.5.6.4 Comparision when the largest two eigenvalues are close. In this section, situations that cause difficulties for both Cason's Algorithm 8 and MROM are considered. First, note that, due to its relationship to the power method, when the two dominant eigenvalues of M^2, where $M = A \otimes B + A^T \otimes B^T$, are very close in magnitude, the rate of the convergence of Algorithm 8 to the desired dominant eigenvector can be very slow. The limiting case when the dominant eigenvalue of M^2 has geometric multiplicity greater than 1, requires that the similarity matrix S is the eigenvector of M^2 with the largest 1-norm. Given an appropriately large bound k, Cason's Algorithm 8 will converge to this eigenvector by design, possibly very slow. Unfortunately, MROM, while converging to an eigenvector of M^2, does not necessarily converge to the desired one.

Fortunately, in general, it does not appear to be a common situation to have large graphs for comparison where M^2 has two such dominant eigenvalues, either analytically or numerically. However, it is possible to construct a family of graphs to illustrate the effect on the algorithms of interest and to provide some basis for the expectation that MROM with some modification can maintain its efficiency and effectiveness.

We consider computing the self-similarity matrix of a graph G defined by a unidirectional cycle and an additional source node. The graph with 10 nodes and uniform edge weights of 1 is shown in Figure 5.11.

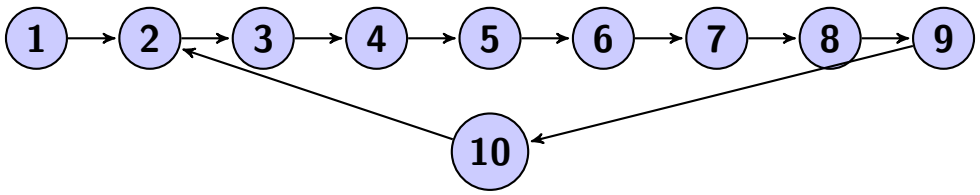

Figure 5.11: A sparse Graph G with 10 nodes.

The adjacency matrix A of graph G in Figure 5.11 is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The two dominant eigenvalues of the matrix M^2, where $M = A \otimes A + A^T \otimes A^T$, are 5.6280 and 5.5818 yielding a ratio of approximately 0.9918 which is close enough to 1 to expect a degradation in convergence.

Table 5.6 shows the relative error ($\frac{\|S-S^B\|_F}{\|S^B\|_F}$, S^B is true similarity matrix obtained from the dominant eigenvector associated with matrix M^2), the final value of the cost function (5.7) and the computational time. With the bound $k = 10$, both methods produce a similarity matrix with rank 10, which is the true rank. The relative error achieved by MROM is much smaller than Algorithm 8, and is essentially at double precision roundoff. Furthermore, although the modified Riemannian optimization method starts from a rank-1 matrix and adjusts rank automatically, the computational time cost is less than Cason's method. This shows the degradation of convergence in Cason's algorithm and that, despite the fact that the eigenvalues are close, they are not close enough to affect MROM's performance.

Table 5.6: Rank $\leq k$ approximation of self-similarity matrix of graph 5.11. The subscript $\pm z$ indicates a scale of $10^{\pm z}$.

	Iteration Method			MROM		
k	relative err	f	time(s)	relative err	f	time(s)
10	2.005_{-06}	5.6280	0.359	3.049_{-14}	5.6280	0.167

A larger member of the family of graphs can be used to illustrate the need for a modification to MROM. Consider the extending the graph above from 10 to 41 nodes and adding a tiered set of

edge weights as seen in the adjacency matrix

$$A = \begin{bmatrix} 0 & 5 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 10^{-5} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 10^{-5} \\ 0 & 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}.$$

.

Checking the singular values of the true rank similarity matrix S^B, we observe that the weighting has caused the singular values to become small after the first six singular values

$$\begin{bmatrix} 0.81700 & 0.57577 & 0.03147 & 0.00085 & 4.66e-05 & 1.26e-06 \end{bmatrix}.$$

Clearly, a low-rank approximation of S^B is a reasonable goal. The two computed dominant eigenvalues of M^2 are equal to double precision accuracy with a value approximately 677.00148 and there is a significant gap to the next eigenvalue at approximately 27. Therefore, the invariant subspace associated with the dominant eigenvalue has dimension 2 and according to [BGH$^+$04], the true rank similarity matrix S^B can be provided by the normalized projection of a vector $\mathbf{1}$ on the dominant invariant subspaces.

Different values for upper bound k ($k = 1, 2, 3, 4, 5, 6$) are used in the experiments. Figure 5.12 shows the results of $\log|\text{grad}\Phi|$ versus number of iterations in Cason's method. It is clear that for $k = 2, 3, 4$ the algorithm has significant difficulties reaching the desired stopping criterion. The rapid convergence for $k = 1$ does not yield the desired eigenvector of M^2 since the bound is too small compared to the numerical rank of 6 of S^B. However, when k is taken large enough at 6 a good approximation of S^B is computed. The first six singular values of the computed similarity matrix, listed in the second column of Table 5.7, are very close to those of S^B given in the first column of the same table.

When MROM with RTR-Newton as inner algorithm is applied to the problem, we observed that, while convergence is reasonable, the method simply converges to an eigenvector associated with the largest eigenvalues. It does not satisfy the property of the desired similarity matrix, i.e., it does not have the largest $\mathbf{1}$ norm. In order to get the approximation of similarity matrix with largest $\mathbf{1}$ norm, a modification such as those proposed in Section 5.3 must be made.

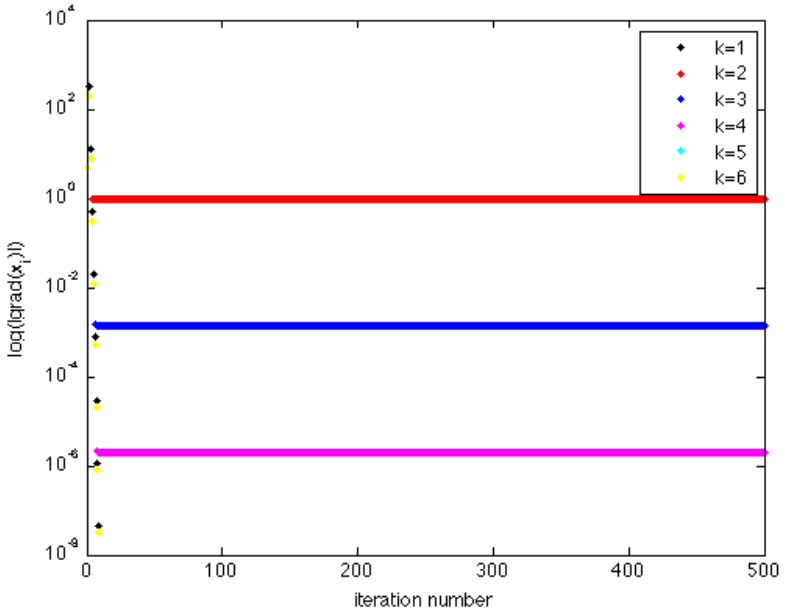

Figure 5.12: Low rank approximation with rank at most k by Cason's Iteration Method.

We consider the first modification proposed in Section 5.3. It requires adding a penalty term on the cost function (5.7), i.e. the following new cost function is considered:

$$\Phi_2(S) := \mathrm{tr}(S^T \mathcal{M}^2(S)) + \lambda \mathbf{1}^T S \mathbf{1}, \tag{5.102}$$

where λ is the penalty coefficient, $\mathbf{1}$ is an all one vector.

The Euclidean gradient of the new cost function $\Phi_2(S)$ is $2\mathcal{M}^2(S) + \frac{\lambda}{\sqrt{mn}} \mathbf{1}^T \mathbf{1}$. As before, projecting the Euclidean gradient onto the tangent space of submanifold $\mathcal{M}$ (the sphere S^{mn-1}) and the fixed-rank manifold $\mathcal{S}_r$ yields the full Riemannian gradient and Riemannian gradient on the fixed-rank manifold.

All the parameters are set to be the same as before. In addition, the new parameter λ in the penalty term is set to be $\frac{\sqrt{mn}}{1000mn}$. The value of the quantity $\log |\mathrm{grad}\Phi|$ as a function of the iteration is shown in Figure 5.13.

For all values of the bound k, MROM on the modified cost function achieves the stopping criterion. The characteristic behavior of changing rank from the rank-1 initial guess at the similarity matrix to the appropriate rank final approximation - in this case always the bound k – is observed.

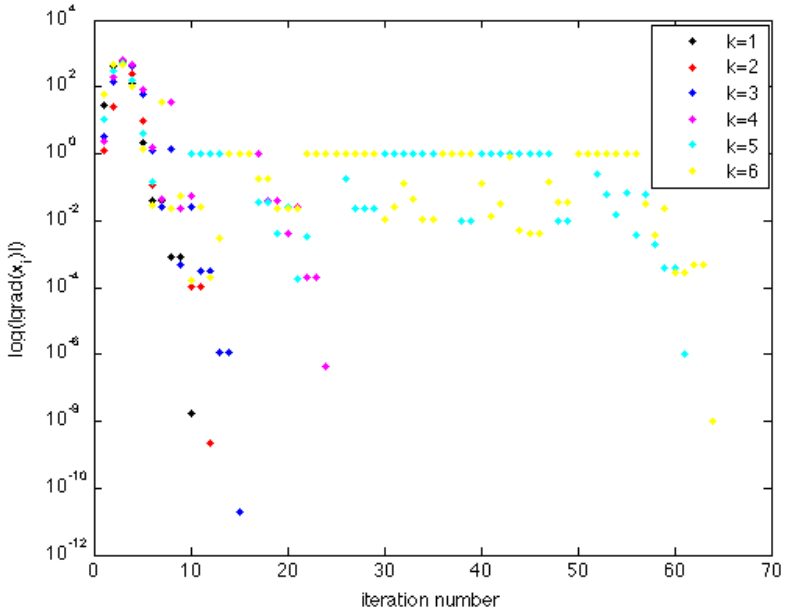

Figure 5.13: Low rank approximation with rank at most k by applying MROM on new cost function.

For a given rank, the value of $\log|\text{grad}\Phi|$ decreases rapidly as the Riemannian optimization on the fixed-rank submanifold $\mathcal{S}_r$ of the sphere converges. After a rank change the value increases as expected and then once again decreases rapidly due to the fixed-rank optimization. Eventually, the final rank is chosen and a value of $\log|\text{grad}\Phi|$ satisfying the stopping criterion is achieved. The third column of Table 5.7 contains the relevant singular values of the similarity matrix computed by MROM on the modified cost function with $k = 6$. Clearly, they are good approximations to those of S^B and they have been computed efficiently.

5.6 Conclusion

The adaptation and application of MROM to generating a low-rank approximation of a graph similarity matrix has been explored in this chapter. While the utility of a low-rank approximation matrix when the similarity matrix is not low-rank either numerically or exactly is still an open question, it is reasonable to consider the efficient computation of such a low-rank approximation, especially for those cases where the true similarity matrix has low-rank structure.

Table 5.7: The first 6 singular values of true similarity matrix S^B, low rank approximation got by Cason's iteration algorithm S^C and low rank approximation got by modified Riemannian optimization algorithm S^M.

S^B	S^C	S^M
0.817005719797966	0.817005719800206	0.817006681732839
0.57576954816249	0.575769548159327	0.575768137078938
0.0314698499477739	0.0314698499477084	0.0314705899851615
0.000852991923203862	0.000852991923194901	0.000853796152030672
4.6621999922645e-05	4.66218975973872e-05	7.2624727193948e-05
1.26369173803024e-06	1.26368896454642e-06	4.53024313970956e-05

Two types of low-rank approximation have been considered. For the first type, the feasible set of low-rank approximations with k identical singular values is a manifold. Therefore, the general Riemannian optimization methods can be applied directly. The second type, approximation with rank at most k, is more general and its feasible set does not have a manifold structure. The algorithm proposed by Cason et al. is essentially equivalent to Blondel's algorithm (Algorithm 5) when $k = \min(m, n)$. What is more, we observe the more efficient version of their algorithm, i.e., Algorithm 8, can be analyzed in terms of a Riemannian manifold rather than a heuristic Euclidean algorithm.

Support for the following conclusions was demonstrated. First, the performances of the Riemannian optimization algorithms on fixed-rank manifold show that working on rank-k manifold is more efficient in terms of space and computational time than using Algorithm 8. Next, the most significant advantage of MROM is its efficient and effective updating of the rank of the approximation to the similarity matrix. Additionally, the performance of MROM was seen to be more efficient for most cases especially the practical case when the numerical rank is not very small and an approximate optimization yields a useful low-rank approximation for the particular application. For the special rank-1 similarity matrix, MROM has significant time advantages compared with Cason's iteration method. This advantage holds even if the starting point has rank greater than 1. MROM can decrease the rank to 1 efficiently to save time and space. MROM's overall robustness was clearly demonstrated.

MROM failed to compute a good approximation to the similarity matrix when the geometric multiplicity of the extremal eigenvalue of M^2 is greater than 1, i.e., the eigenspace associated to the extremal eigenvalue is not one-dimensional. While this situation appears to be uncommon,

MROM can only guarantee convergence to an eigenvector, not necessarily the unique one with the largest 1-norm required for this special case. Applying MROM to one of the proposed modified cost functions has been shown to address the problem for a family of example graphs. We have also introduced other modified cost functions in Section 5.3. They may prove useful for this special case.

CHAPTER 6

CONCLUSION AND FUTURE RESEARCH

In this dissertation, we present new algorithms that solve optimization problems on a matrix manifold $\mathcal{M} \subseteq \mathbb{R}^{m \times n}$ with an additional rank inequality constraint. New geometric objects are defined to facilitate efficiently finding a suitable rank. The convergence properties of the algorithms are analyzed and empirically verified. Experiments and applications are used to illustrate the efficiency and effectiveness of the algorithm.

The major contributions of this dissertation are:

1. Developed a rank-related vector that defines a search direction on the tangent cones.
2. Developed a rank-related retraction that facilitates the change from one fixed-rank manifold to another that is more flexible and effective than fixed-increment updating.
3. Developed a general algorithm with flexible fixed-rank inner algorithm choice to solve optimization problems with rank inequality constraints.
4. Completed the convergence theory for the new algorithms.
5. Empirically evaluated the algorithms for two important applications: weighted low-rank matrix approximation problems and low-rank approximation of a graph similarity matrix.
6. For weighted low-rank matrix approximation problems:
   (a) Empirically evaluated the ability of the new algorithms to determine a space efficient approximation when the singular value profile is gapless and strongly gapped for general weighting matrix for a range of problem sizes and choice of fixed-rank inner algorithm.
   (b) Empirically evaluated the performance of the new algorithms for problems with a structured weighting matrix for a range of problem sizes and choice of fixed-rank inner algorithm.
   (c) Empirically evaluated the influence of the retraction and its factor (in)variance for a range of problem sizes and choice of fixed-rank inner algorithm.
   (d) Empirically evaluated the performance of different inner algorithms and sizes. For large size matrix, the limited-memory algorithms were shown to be preferable.

7. For low-rank approximation of a graph similarity matrix:

   (a) Developed algorithms with flexible fixed-rank inner algorithm choice for geometric multiplicity of 1 and greater than 1.

   (b) Empirically evaluated the new algorithms for problems with geometric multiplicity of 1 for a range of problem sizes and choice of fixed-rank inner algorithm.

   (c) Characterized some problems that yield geometric multiplicity greater than 1.

   (d) Considered the case when the similarity matrix defined by the full-rank iteration does not have a good low-rank approximation in the sense of the Frobenius norm and determined if the low-rank approximations that are produced by the proposed algorithms have any useful information for the related graph problems.

There are several avenues of future research in both algorithms and their applications. For algorithms, we will consider developing heuristics to choose/adapt parameters ϵ_1 and ϵ_2 in Algorithm 2. To avoid the singular value decomposition for a large matrix, we implement three-factor representations of each matrix. Further analysis is needed on the invariance and variance of retractions with respect to the three-factor representations.

For applications, there are other constraints on the approximating matrix of interest in the literature apart from the rank constraint, e.g., non-negativity and Hankel structure. We will continue to adapt and improve the Riemannian methods and our understanding of their behavior and its relationship to application characteristics and constraints. The adapted and improved methods will be systematically compared with current state-of-the-art methods in each application area.

BIBLIOGRAPHY

[AAM14] P.-A. Absil, Luca Amodei, and Gilles Meyer, *Two Newton methods on the manifold of fixed-rank matrices endowed with riemannian quotient geometries*, Computational Statistics **29** (2014), no. 3-4, 569–590.

[ABG07] Pierre-Antoine Absil, C. G. Baker, and Kyle A. Gallivan, *Trust-region methods on Riemannian manifolds*, Foundations of Computational Mathematics **7** (2007), no. 3, 303–330.

[ADM+02] Roy L. Adler, Jean-Pierre Dedieu, Joseph Y. Margulies, Marco Martens, and Michael Shub, *Newton's method on Riemannian manifolds and a geometric model for the human spine*, IMA Journal of Numerical Analysis **22** (2002), no. 3, 359–390.

[AM07] Dimitris Achlioptas and Frank McSherry, *Fast computation of low-rank matrix approximations*, J. ACM **54** (2007), no. 2, –1–1.

[AM12] Pierre-Antoine Absil and Jerome Malick, *Projection-like retractions on matrix manifolds.*, SIAM Journal on Optimization **22** (2012), no. 1, 135–158.

[AMS08] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, Princeton, NJ, 2008.

[AO13] P.-A. Absil and I. V. Oseledets, *Low-rank retractions: a survey and new results*, Tech. Report UCL-INMA-2013.04-v1, Catholic University of Louvain, October 2013.

[Bak08] C. G. Baker, *Riemannian manifold trust-region methods with applications to eigenproblems*, Ph.D. thesis, School of Computational Science, Florida State University, Summer Semester 2008.

[BGH+04] Vincent D. Blondel, Anahí Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren, *A measure of similarity between graph vertices: Applications to synonym extraction and web searching*, SIAM Review **46** (2004), no. 4, 647–666.

[BGV12] C. G. Baker, K. A. Gallivan, and P. Van Dooren, *Low-rank incremental methods for computing dominant singular subspaces*, Linear Algebra and Its Applications **436** (2012), no. 8, 2866–2888.

[BM06] I. Brace and J. H. Manton, *An improved BFGS-on-manifold algorithm for computing low-rank weighted approximations*, Proceedings of 17th International Symposium on Mathematical Theory of Networks and Systems, 2006, pp. 1735–1738.

[CAD13] T.P. Cason, P.A. Absil, and P. Van Dooren, *Iterative methods for low rank approximation of graph similarity matrices*, Linear Algebra and its Applications **438** (2013), no. 4, 1863 – 1882, 16th ILAS Conference Proceedings, Pisa 2010.

[CAVD11] T.P. Cason, P.-A. Absil, and P. Van Dooren, *Comparing two matrices by means of isometric projections*, Numerical Linear Algebra in Signals, Systems and Control (Paul Van Dooren, Shankar P. Bhattacharyya, Raymond H. Chan, Vadim Olshevsky, and Aurobinda Routray, eds.), Lecture Notes in Electrical Engineering, vol. 80, Springer Netherlands, 2011, pp. 77–93 (English).

[DK70] Chandler Davis and W. M. Kahan, *The rotation of eigenvectors by a perturbation. III*, SIAM Journal on Numerical Analysis **7** (1970), no. 1, pp. 1–46.

[DKM06] Petros Drineas, Ravi Kannan, and Michael W. Mahoney, *Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix*, SIAM Journal on Computing **36** (2006), no. 1, 158–183.

[DM05] Petros Drineas and Michael W. Mahoney, *On the Nyström method for approximating a Gram matrix for improved kernel-based learning*, Journal of Machine Learning Research **6** (2005), 2153–2175.

[DPM02] J.-P. Dedieu, P. Priouret, and G. Malajovich, *Newton method on Riemannian manifolds: Covariant alpha-theory*, ArXiv Mathematics e-prints (2002), –1–1.

[DV06] Amit Deshpande and Santosh Vempala, *Adaptive sampling and fast low-rank matrix approximation*, Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (Josep Daz, Klaus Jansen, JosD.P. Rolim, and Uri Zwick, eds.), Lecture Notes in Computer Science, vol. 4110, Springer Berlin Heidelberg, 2006, pp. 292–303.

[EAS98] Alan Edelman, Toms A. Arias, and Steven T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM Journal on Matrix Analysis and Applications **20** (1998), no. 2, 303–353.

[EY36] Carl Eckart and Gale Young, *The approximation of one matrix by another of lower rank*, Psychometrika **1** (1936), no. 3, 211–218 (English).

[FHB04] M. Fazel, H. Hindi, and S. Boyd, *Rank minimization and applications in system theory*, Proceedings of American Control Conference, AACC, 2004, pp. 3273–3278.

[FND08] Catherine Fraikin, Yurii E. Nesterov, and Paul Van Dooren, *Optimizing the coupling between two isometric projections of matrices.*, SIAM Journal on Matrix Analysis and Applications **30** (2008), no. 1, 324–345.

[Gab82] D. Gabay, *Minimizing a differentiable function over a differential manifold*, Journal of Optimization Theory and Applications **37** (1982), no. 2, 177–219 (English).

[GP10] V. Guillemin and A. Pollack, *Differential topology*, AMS Chelsea Publishing Series, AMS Chelsea Pub, 2010.

[GQA12] Kyle A. Gallivan, Chunhong Qi, and P.-A. Absil, *A Riemannian Dennis-Moré condition*, High-Performance Scientific Computing (Michael W. Berry, Kyle A. Gallivan, Efstratios Gallopoulos, Ananth Grama, Bernard Philippe, Yousef Saad, and Faisal Saied, eds.), Springer London, 2012, pp. 281–293.

[GZ13] Jonathan Gillard and Anatoly A. Zhigljavsky, *Optimization challenges in the structured low rank approximation problem*, J. Global Optimization **57** (2013), no. 3, 733–751.

[GZ14] J. Gillard and A. Zhigljavsky, *Stochastic methods for Hankel structured low rank approximation*, Proceedings of 21th International Symposium on Mathematical Theory of Networks and Systems, 2014, pp. 961–964.

[Hai01] E. Hairer, *Geometric integration of ordinary differential equations on manifolds*, BIT **41** (2001), no. 5, 996–1007.

[Hai11] ———, *Solving differential equations on manifolds*, 2011, Lecture Notes.

[HJ90] Roger A. Horn and Charles R. Johnson, *Matrix Analysis*, Cambridge University Press, 1990.

[HMT11] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabalistic algorithms for computing matrix decompositions*, SIAM Review **53** (2011), no. 2, 217–288.

[HT04] K. Huper and J. Trumpf, *Newton-like methods for numerical optimization on manifolds*, Signals, Systems and Computers, 2004. Conference Record of the Thirty-Eighth Asilomar Conference on, vol. 1, Nov 2004, pp. 136–139 Vol.1.

[Hua13] W. Huang, *Optimization algorithms on Riemannian manifolds with applications*, Ph.D. thesis, Florida State University, 2013.

[JBAS10a] M. Journe, F. Bach, P. Absil, and R. Sepulchre, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM Journal on Optimization **20** (2010), no. 5, 2327–2351.

[JBAS10b] Michel Journe, Francis R. Bach, Pierre-Antoine Absil, and Rodolphe Sepulchre, *Low-rank optimization on the cone of positive semidefinite matrices*, SIAM Journal on Optimization **20** (2010), no. 5, 2327–2351.

[JHSX11] Hui Ji, Si-Bin Huang, Zuowei Shen, and Yuhong Xu, *Robust video restoration by joint sparse and low rank matrix approximation.*, SIAM Journal on Imaging Sciences **4** (2011), no. 4, 1122–1142.

[KL07] Othmar Koch and Christian Lubich, *Dynamical low-rank approximation.*, SIAM Journal on Matrix Analysis and Applications **29** (2007), no. 2, 434–454.

[Kri06] Wim P. Krijnen, *Convergence of the sequence of parameters generated by alternating least squares algorithms*, Computational Statistics and Data Analysis **51** (2006), no. 2, 481 – 489.

[LKLS13] Joonseok Lee, Seungyeon Kim, Guy Lebanon, and Yoram Singer, *Matrix approximation under local low-rank assumption*, CoRR **abs/1301.3192** (2013), –1 – 1.

[LLR95] Nathan Linial, Eran London, and Yuri Rabinovich, *The geometry of graphs and some of its algorithmic applications*, Combinatorica **15** (1995), no. 2, 215–245.

[LPW97] W.-S. Lu, S.-C. Pei, and P.-H. Wang, *Weighted low-rank approximation of general complex matrices and its application in the design of 2-d digital filters*, IEEE Transactions on Circuits and SystemsI **44** (1997), 650–655.

[Lue72] David G. Luenberger, *The gradient projection method along geodesics*, Management Science **18** (1972), no. 11, 620–631.

[Lue73] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.

[Man02] J.H. Manton, *Optimization algorithms exploiting unitary constraints*, IEEE Transactions on Signal Processing **50** (2002), no. 3, 635–650.

[Mar11] I. Markovsky, *Low rank approximation: Algorithms, implementation, applications*, Communications and Control Engineering, Springer, 2011.

[Mes98] Mehran Mesbahi, *On the rank minimization problem and its control applications*, Systems and Control Letters **33** (1998), no. 1, 31 – 36.

[MK97] Toshihiko Morita and Takeo Kanade, *A sequential factorization method for recovering shape and motion from image streams*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), no. 8, 858–867.

[MM02] Robert Mahony and JonathanH. Manton, *The geometry of the newton method on non-compact lie groups*, Journal of Global Optimization **23** (2002), no. 3-4, 309–327.

[MMBS13] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre, *Low-rank optimization with trace norm penalty*, SIAM Journal on Optimization **23** (2013), no. 4, 2124–2149.

[MMH03] Jonathan H. Manton, Robert Mahony, and Yingbo Hua, *The geometry of weighted low-rank approximations*, IEEE Transactions on Signal Processing **51** (2003), no. 2, 500–514.

[MRP+06] Ivan Markovsky, Maria Luisa Rastello, Amedeo Premoli, Alexander Kukush, and Sabine Van Huffel, *The element-wise weighted total least-squares problem*, Computational Statistics and Data Analysis **50** (2006), no. 1, 181 – 209, 2nd Special issue on Matrix Computations and Statistics.

[MS13] B. Mishra and R. Sepulchre, *R3MC: A Riemannian three-factor algorithm for low-rank matrix completion*, ArXiv e-prints (2013), –1–1.

[Mun00] J.R. Munkres, *Topology*, Prentice Hall, 2000.

[MV14] B. Mishra and B. Vandereycken, *A Riemannian approach to low-rank algebraic Riccati equations*, Proceedings of 21th International Symposium on Mathematical Theory of Networks and Systems, 2014, pp. 965–968.

[NW06] Jorge Nocedal and Stephen J. Wright, *Numerical Optimization*, 2. ed. ed., Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, 2006.

[OW00] B. Owren and B. Welfert, *The Newton iteration on Lie groups*, BIT **40** (2000), no. 1, 121–145.

[OW04] Donal B. O'Shea and Leslie C. Wilson, *Limits of tangent spaces to real surfaces*, American Journal of Mathematics **126** (2004), no. 5, pp. 951–980 (English).

[PDGM10] Panagiotis Papadimitriou, Ali Dasdan, and Hector Garcia-Molina, *Web graph similarity for anomaly detection*, Journal of Internet Services and Applications **Volume 1** (2010), no. 1, 19–30.

[PR02a] Amedeo Premoli and Maria Luisa Rastello, *The parametric quadratic form method for solving TLS problems with elementwise weighting*, Van Huffel, Sabine (ed.) et al., Total least squares and errors-in-variables modeling. Analysis, algorithms and applications. Dordrecht: Kluwer Academic Publishers. 67-76 (2002)., 2002.

[PR02b] Amedeo Premoli and MariaLuisa Rastello, *The parametric quadratic form method for solving tls problems with elementwise weighting*, Total Least Squares and Errors-in-Variables Modeling (Sabine Huffel and Philippe Lemmerling, eds.), Springer Netherlands, 2002, pp. 67–76.

[Qi11] C. Qi, *Numerical optimization methods on Riemannian manifolds*, Ph.D. thesis, Florida State University, 2011.

[RW12] W. Ring and B. Wirth, *Optimization methods on Riemannian manifolds and their application to shape space*, SIAM Journal on Optimization **22** (2012), no. 2, 596–627.

[Shu86] M. Shub, *Some remarks on dynamical systems and numerical analysis*, Proceedings VII ELAM (L. Lara-Carrero and J. Lewowicz, eds.), 1986, pp. 69–92.

[Smi93] Steven Thomas Smith, *Geometric optimization methods for adaptive filtering*, Ph.D. thesis, Harvard University, Cambridge, MA, USA, 1993, UMI Order No. GAX93-31032.

[Ste83] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM Journal on Numerical Analysis **20** (1983), no. 3, 626–637.

[SU14] R. Schneider and A. Uschmajew, *Convergence results for projected line-search methods on varieties of low-rank matrices via Łojasiewicz inequality*, ArXiv e-prints (2014), –1–1.

[UV14] A. Uschmajew and B. Vandereycken, *Line-search methods and rank increase on low-rank matrix varieties*, Proceedings of the 2014 International Symposium on Nonlinear Theory and its Applications (NOLTA2014), 2014.

[Van13] B. Vandereycken, *Low-rank matrix completion by Riemannian optimization*, SIAM Journal on Optimization **23** (2013), no. 2, 1214—1236.

[VB94] Lieven Vandenberghe and Stephen Boyd, *Semidefinite programming*, SIAM Review **38** (1994), 49–95.

[VV89] Sabine Van Huffel and Joos Vandewalle, *Analysis and properties of the generalized total least squares problem $AX \approx B$ when some or all columns in a are subject to error*, SIAM Journal on Matrix Analysis and Applications **10** (1989), no. 3, 294–315.

[WAK97] Peter D. Wentzell, Darren T. Andrews, and Bruce R. Kowalski, *Maximum likelihood multivariate calibration*, Analytical Chemistry **69** (1997), no. 13, 2299–2311.

[Whi92] Hassler Whitney, *Local properties of analytic varieties*, Hassler Whitney Collected Papers (James Eells and Domingo Toledo, eds.), Contemporary Mathematicians, Birkhuser Boston, 1992, pp. 497–536.

[Wu02] Lixin Wu, *Fast at-the-money calibration of the Libor market model using lagrange multipliers*, Journal of Computational Finance (2002), 39–77.

[Ye05] Jieping Ye, *Generalized low rank approximations of matrices*, Machine Learning **61** (2005), no. 1-3, 167–191.

[ZSJC12] D. Zachariah, M. Sundin, M. Jansson, and S. Chatterjee, *Alternating least-squares for low-rank matrix reconstruction*, IEEE Signal Processing Letters **19** (2012), no. 4, 231–234.

[ZW03] Zhenyue Zhang and Lixin Wu, *Optimal low-rank approximation to a correlation matrix*, Linear Algebra and its Applications **364** (2003), 161 – 187.

BIOGRAPHICAL SKETCH

Guifang Zhou, daughter of Daoping Zhou and Suzhen Gui, was born on January 7th, 1985 in Ningguo, Anhui province of P.R. China. She completed her Bachelor degree in Mathematics and Applied Mathematics in 2006 and her Master degree in Applied Mathematics in 2009, both from Anhui University in China. She enrolled in the doctoral program at the Florida State University in 2009. After obtaining her Master degree in Applied and Computational Mathematics in 2012, she is currently under the advisement of Prof. Kyle A. Gallivan and Prof. Paul Van Dooren.

Guifang Zhou's research interests include optimization methods on manifolds with rank restriction and their application to problems such as weighted low-rank approximation and low-rank approximation of similarity matrix. Besides the dissertation research, she also contribute to develop software, called TreeScaper, for phylogenetic analysis.

After her Ph.D., Guifang will start her post doctoral research position in the Department of Biological Sciences at Louisiana State University.