UNIVERSITY OF LIÈGE

A Riemannian approach to large-scale constrained least-squares with symmetries

by

Bamdev Mishra

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy
in the
Faculty of Applied Sciences
Department of Electrical Engineering and Computer Science

October 2014

Jury:

Prof. Louis Wehenkel (President)	University of Liège
Prof. Rodolphe Sepulchre (Supervisor)	University of Liège
Prof. Francis Bach (Co-supervisor)	Ecole Normale Supérieure, Paris
Prof. Quentin Louveaux	University of Liège
Prof. Lieven De Lathauwer	Katholieke Universiteit Leuven
Prof. Philippe Toint	Université de Namur

UNIVERSITY OF LIÈGE

Abstract

Faculty of Applied Sciences
Department of Electrical Engineering and Computer Science

by Bamdev Mishra

This thesis deals with least-squares optimization on a manifold of equivalence relations, e.g., in the presence of *symmetries* which arise frequently in many applications. While least-squares cost functions remain a popular way to model large-scale problems, the additional symmetry constraint should be interpreted as a way to make the modeling robust. Two fundamental examples are the *matrix completion problem*, a least-squares problem with *rank* constraints and the *generalized eigenvalue problem*, a least-squares problem with *orthogonality* constraints. The possible large-scale nature of these problems demands to exploit the problem structure as much as possible in order to design numerically efficient algorithms.

The constrained least-squares problems are tackled in the framework of Riemannian optimization that has gained much popularity in recent years because of the special nature of orthogonality and rank constraints that have particular symmetries. Previous work on Riemannian optimization has mostly focused on the search space, exploiting the differential geometry of the constraint but disregarding the role of the cost function. We, on the other hand, propose to take both cost and constraints into account to propose a *tailored* Riemannian geometry. This is achieved by proposing novel *Riemannian metrics*. To this end, we show a basic connection between sequential quadratic programming and Riemannian gradient optimization and address the general question of selecting a metric in Riemannian optimization. We revisit quadratic optimization problems with orthogonality and rank constraints by generalizing various existing methods, like power, inverse and Rayleigh quotient iterations, and proposing novel ones that empirically compete with state-of-the-art algorithms.

Overall, this thesis deals with exploiting two fundamental structures, least-squares and symmetry, in nonlinear optimization.

UNIVERSITY OF LIÈGE

Resumé

Faculty of Applied Sciences
Department of Electrical Engineering and Computer Science

by Bamdev Mishra

Cette thèse de doctorat traite de l'optimisation au sens des moindres carrés sur une variété de relations d'équivalence, i.e., en présence de *symétries* qui apparaissent fréquemment dans plusieurs applications. Alors que les fonctions de coût de moindres carrés restent un moyen répandu de modéliser les problèmes à grande dimension, la contrainte additionnelle de symétrie peut être interprétée comme un moyen de rendre la modélisation robuste. Deux exemples fondamentaux sont le *problème de complétion de matrices*, un problème aux moindres carrés où des contraintes de *rang* sont présentes, et le *problème des valeurs propres généralisées*, un problème aux moindres carrés soumis à des contraintes d'*orthogonalité*. Les dimensions potentiellement importantes de ces problèmes appellent à en exploiter au maximum la structure , dans le but de développer des algorithmes numériquement efficients.

Les problèmes aux moindres carrés contraints sont ici résolus en utilisant l'ensemble des outils de l'optimisation Riemannienne, qui a récemment gagné en popularité grâce à la possibilité qu'elle offre de gérer la nature spéciale des contraintes d'orthogonalité et de rang aux symétries particulières. Les précédents travaux d'optimisation Riemannienne se sont principalement attachés à l'espace de recherche, en exploitant la géométrie différentielle des contraintes, mais sans s'intéresser au rôle de la fonction de coût. Dans ce travail, au contraire, nous proposons de prendre en compte les contraintes et le coût, dans le but de proposer une géométrie Riemannienne ajustée. Ceci est rendu possible par l'introduction de nouvelles *métriques Riemanniennes*. Dans ce but, nous montrons une connexion basique entre la programmation quadratique séquentielle et l'algorithme du gradient Riemannien. La question générale de la sélection d'une métrique pour l'optimisation Riemannienne est aussi abordée. L'optimisation quadratique pour les problèmes aux contraintes de rang et d'orthogonalité est revisitée, en généralisant plusieurs méthodes existantes telles que les méthodes itératives utilisant la puissance, l'inverse, ou le quotient de Rayleigh. De nouvelles méthodes sont aussi proposées, qui sont empiriquement compétitives avec les techniques de l'état de l'art.

D'une manière générale, cette thèse de doctorat concerne l'exploitation de deux structures fondamentales, les moindres carrés et la symétrie, pour l'optimisation non-linéaire.

To my parents

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisor, Prof. Rodolphe Sepulchre (University of Liège), for his constant guidance, support, and encouragement throughout my PhD study. I am indebted to him for introducing me to research in general and to the field of optimization on matrix manifolds in particular. I am also grateful to him for hosting my research visit to the University of Cambridge.

I would also like to express my sincere gratitude to my co-supervisor, Prof. Francis Bach (Ecole Normale Supérieure, Paris), for his constant guidance, support, and encouragement. I am also grateful to him for his hospitality during my research visits to INRIA Paris-Rocquencourt.

I wish to express my sincere gratitude to my collaborators, Prof. Pierre-Antoine Absil (Université Catholique de Louvain), Prof. Silvère Bonnabel (Mines ParisTech), Dr. Nicolas Boumal (Université Catholique de Louvain), Adithya Apuroop Karavadi, Dr. Gilles Meyer, and Dr. Bart Vandereycken (Princeton University), for their encouragement and feedback that constantly helped me in my learning process and kept me motivated.

I am thankful to all my former and present colleagues at the University of Liège, the University of Cambridge, and INRIA Paris-Rocquencourt. My heartfelt thanks go to Mathieu Claeys, Anne Collard, Julie Dethier, Guillaume Drion, Fulvio Forni, Alessio Franci, Rodolphe Jenatton, Michel Journée, Augustin Lefèvre, Raphaël Liégeois, Alexandre Mauroy, Pierre Sacré, Alain Sarlette, Laura Trotta, and Mattia Zorzi. It has been a real pleasure working with them. Morning coffee breaks and group lunches gave unique opportunities to learn, share, and discuss interesting maths and ideas.

I would also like to express my sincere gratitude to all the members of the Jury, Prof. Francis Bach (Ecole Normale Supérieure, Paris), Prof. Lieven De Lathauwer (Katholieke Universiteit Leuven), Prof. Quentin Louveaux (University of Liège), Prof. Philippe Toint (Université de Namur), Prof. Rodolphe Sepulchre (University of Liège), and Prof. Louis Wehenkel (University of Liège), for devoting time and interest to the manuscript.

I gratefully acknowledge the financial support from the Belgian National Fund for Scientific Research (FNRS) as well as from the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office.

Finally, I wish to thank my family and friends for their unconditional support and encouragement.

Contents

Chapter 1

Introduction

In most online shopping activities today, *recommendations* form an integral part of consumer-company interaction which in turn result from *predicting* the online buying preferences of a consumer. A well-know example is the prediction of movie ratings, popularized by the *MovieLens* recommendation website (MovieLens, 1997) and the famous *Netflix prize* problem (Netflix, 2006). The problem in such tasks amounts to estimating missing entries of a movie ratings matrix, a very sparse matrix with few ratings per user, from a limited number of its known entries, where an entry of the matrix corresponds to a user's rating for a specific movie. The aim then is to predict ratings that the user might have had for the unseen movies, had he or she seen them before. For example, the dataset MovieLens (1997) has a million known ratings that correspond to about 4% of the total ratings. Similarly, the dataset Netflix (2006) has a hundred million ratings that correspond to 1% of the total number of ratings. Such tasks fall in the arena of, what is called, *collaborative filtering* (Abernethy et al., 2009; Rennie and Srebro, 2005).

A prior assumption that is frequently used in such prediction tasks is that the underlying matrix is *low-rank*, i.e., the rank of the matrix is very small compared to the dimensions of the matrix. Prediction of unknown ratings with a low-rank prior has the interpretation that users' preferences of movies depend only on few movie (but unknown) *genres*, which is often the case in practice. The importance of the rank constraint in the Netflix prize problem has been highlighted in the blog by Amatriain and Basilico (2012), where they quote:

> "We looked at the two underlying algorithms with the best performance in the ensemble: Matrix Factorization (which the community generally called SVD, Singular Value Decomposition) and Restricted Boltzmann Machines (RBM). SVD by itself provided a 0.8914 Root Mean Square Error (RMSE), while RBM alone provided a competitive but slightly worse 0.8990 RMSE. A linear blend of these two reduced the error to 0.88. To put these algorithms to use, we had to work to overcome some limitations, for instance that they were built to handle 100 million ratings, instead of the more than 5 billion that we have, and that they were not built to adapt as members added more ratings. But once we overcame those challenges, we put the two algorithms into production, where they are still used as part of our recommendation engine. . . ",

which clearly stresses the importance of rank constraint in the Netflix recommendation task by emphasizing the role of *singular value decomposition*. It should be stated that the singular value decomposition is a fundamental tool in matrix analysis (Golub and Van Loan, 1996).

In an academic setup the above prediction problem is simplified to an optimization problem that minimizes a *least-squares* cost function with a *low-rank constraint*. Reconstruction or completion of a low-rank matrix under particular assumptions on the distribution of known entries and by exploiting the rank constraint parameterizations have been proposed by Candès and Recht (2009); Gross (2011); Keshavan et al. (2010). This includes both the *fixed-rank* approach and the *convex relaxation* approach with *trace norm* (also called *nuclear norm*) (Candès and Recht, 2009; Fazel, 2002; Recht et al., 2010). Simultaneously, this has also led to much research in developing computationally efficient algorithms (Boumal and Absil, 2011; Cai et al., 2010; Jain et al., 2010; Keshavan et al., 2010; Lee and Bresler, 2010; Mazumder et al., 2010; Meyer et al., 2011a; Ngo and Saad, 2012; Rennie and Srebro, 2005; Vandereycken, 2013). An iterative algorithm is understood to scale well when its cost per iteration scales *linearly* with data. In most cases, fixed-rank *matrix factorizations* (including the singular value decomposition) play a critical role in achieving computational efficiency (Absil et al., 2014; Bonnabel and Sepulchre, 2009; Boumal and Absil, 2011; Burer and Monteiro, 2003; Dai et al., 2012; Journée et al., 2010; Keshavan et al., 2010; Meyer et al., 2011a; Ngo and Saad, 2012; Vandereycken, 2013).

The importance of matrix factorizations and rank constraint is not confined to low-rank matrix completion alone and it plays a fundamental role in problems spanning classification (Amit et al., 2007), image clustering (Joulin et al., 2010), learning on pairs (Abernethy et al., 2009), learning low-rank distances (Kulis et al., 2009; Meyer et al., 2011b), structured low-rank approximation (Markovsky, 2008, 2014), neuroimaging (Vounou et al., 2010), control systems applications (Benner and Saak, 2013), tensor completion (Da Silva and Herrmann, 2014; Kressner et al., 2013) to name a few.

A different set of constraints that connect naturally to the rank constraint are *orthogonality* constraints which have also been a topic of much research over the years (Absil et al., 2004a; Edelman et al., 1998; Eldén and Park, 1999; Manton, 2002; Wen and Yin, 2013). A fundamental example concerning orthogonality constraints is the *generalized eigenvalue problem*, a least-squares problem with orthogonality constraints (Absil et al., 2002; Edelman et al., 1998; Golub and Van Loan, 1996). Some important applications include subspace identification (Balzano et al., 2010), analysis of gene expression data (Journée et al., 2010; Teschendorff et al., 2007), synchronization of rotations (Boumal et al., 2013), system identification (Usevich and Markovsky, 2014), and the orthogonal *Procustes* problem (Viklands, 2006).

Both orthogonality and rank constraints are *nonlinear* and *nonconvex* but nevertheless *very special*. In particular, they have the structure of a *quotient manifold*, a topological space arising from structured *symmetries* on a *matrix manifold* (Absil et al., 2008; Edelman et al., 1998; Meyer, 2011). The precise definition of a quotient manifold can be found in the book by Lee (2003, Chapter 9). A matrix manifold has the property that its elements have suitable matrix characterizations. Structured symmetries here refer to particular *equivalence relations* that may exist on matrix manifolds. Consequently, both rank-constrained matrix completion and generalized eigenvalue problems are optimization problems over quotient manifolds. A further motivation as to why acknowledging the quotient structure is critical in optimization is discussed in Section 2.3.

Although the study of matrix manifolds, per se, is a classical subject (see Lee (2003) and the references therein), the development of numerical optimization techniques over matrix manifolds, including quotient manifolds, have arose significant interest only recently, primarily driven by concrete applications. An important reference is the seminal work of Edelman et al. (1998) that bridges the gap between abstract differential geometric objects and their efficient matrix representations. Particularly, it shows the development of algorithms for problems with orthogonality constraints. A general treatment of deriving concretely a number of first-order and second-order algorithms for optimization on matrix manifolds (including quotient manifolds) is given in the recent monograph by Absil et al. (2008). The references from Absil et al. (2008); Edelman et al. (1998) form the foundation on which the present thesis rests.

Consider an optimization problem that minimizes a cost function over a set of equality constraints. (We assume here that the set of equality constraints is a differentiable manifold.) For this case, the advocated Riemannian framework of Absil et al. (2008); Edelman et al. (1998) proceeds by endowing the set of equality constraints with a *Riemannian metric structure*, i.e., a Riemannian manifold structure that is endowed with a Riemannian metric (a smooth inner product on the manifold). As a result, the constrained optimization problem (minimization of a smooth cost function over a set of equality constraints) is *conceptually* translated into an optimization problem on a Riemannian manifold (structure of the set of equality constraints). This has the interpretation of an *unconstrained optimization* problem over a Riemannian manifold which paves the way to extend a number of unconstrained methods to the manifold setup. The Riemannian algorithms have been shown to be competitive with various state-of-the-art on many different benchmarks across many different applications. See Absil (2003); Boumal (2014); Edelman et al. (1998); Journée (2009); Meyer (2011); Vandereycken (2010) for the application of Riemannian algorithms on various problems. The benefits of the Riemannian optimization framework are twofold. First, it takes the intrinsic geometry of the problem into account and provides a cluster of algorithms that come with rigorous convergence analysis. It also lays down a systematic procedure to handle symmetries, the theme of the present thesis. The second benefit of the framework is the recent and ongoing developments of competitive softwares such as Manopt (Boumal et al., 2014) to support the numerical implementation of Riemannian algorithms.

It should also be mentioned that most of the aforementioned optimization problems involve minimizing least-squares cost functions. Least-squares have been a popular class of optimization problems in various fields of engineering applications (Nocedal and Wright, 2006, Chapter 10). The primary reason for this popularity arises from the fact that the least-squares structure of the cost function leads to an efficient computation of first-order derivatives (*gradient* information) and also an efficient approximation of second-order derivatives. The least-squares structure becomes all the more relevant when dealing with large dimensional problems, where even computation of first-order derivatives are computationally demanding. A second source of least-squares problems arise naturally while solving systems of linear and nonlinear equations (Nocedal and Wright, 2006, Chapter 11). Quite often, the exercise to solve a system of equations (linear or nonlinear) translates to solving an optimization problem where the *critical points* of a (chosen) least-squares cost are identified with zeros of the system of equations that is sought to be solved.

While, individually, both least-squares and the geometry of rank and orthogonality constraints are well-studied in literature, this thesis exploits these two fundamental structures *simultaneously* in the Riemannian optimization setup. Our particular emphasis lies on exploiting the *Riemannian metric structure* on the sets of orthogonality and rank constraints in optimization problems. The thesis deals with a number of specific (and popular) least-squares optimization problems to showcase this contribution of the thesis.

1.1 Contributions of the thesis and the related publications

The thesis is primarily motivated by the low-rank matrix completion problem that is viewed as a least-squares problem on a matrix manifold with symmetries. The manifold here is the manifold of fixed-rank matrices. The main ideas in this thesis, however, extend beyond the low-rank matrix completion example to other least-squares problems involving rank and orthogonality constraints. We show this by considering the problems of generalized eigenvalue decomposition, large-scale matrix Lyapunov equations, and multivariate linear regression. All these problems are tackled in the framework of Riemannian manifold optimization with the main emphasis on proposing *computationally efficient* algorithms.

The main contribution of the thesis is to stress the benefit of a Riemannian structure that depends on both the constraints and the cost function. The conventional way is to disregard the role of cost function in deciding the Riemannian structure on the constraints.

The specific contributions of the thesis and the related publications are as follows.

- We review a number of fixed-rank matrix factorizations and study their underlying symmetries. It is emphasized that the symmetries arise from the interplay of few (but individually well-studied) matrix manifolds.

  This is discussed in Chapter 2 and the main theme is based on the publication (Mishra et al., 2014).

- In Chapter 3, we explore the connection between Riemannian gradient optimization and sequential quadratic programming (for equality constraints). The question of selecting a Riemannian metric on a Riemannian manifold is addressed. Traditionally, this question has been answered by exploiting the differential geometric framework of the search space but disregarding the role of the cost function in an optimization problem. Instead, we propose to take both the cost function and the constraints of an optimization problem into account to decide a metric structure. To this end, sequential quadratic programming is shown to provide a systematic guidance to choose a metric in Riemannian optimization. This choice of metric has a *preconditioning* effect on optimization problems, often an advantage in dealing with ill-conditioned data. The relevance of metric tuning is shown on least-squares problems with rank and orthogonality constraints. Specifically, on the *generalized eigenvalue problem* and the problem of computing low-rank solution to matrix *Lyapunov equations*. For the generalized eigenvalue problem, we show connection to *power*, inverse, and Rayleigh quotient iteration. For the matrix Lyapunov equations, we propose a simpler metric that has shown promising initial results.

  This work has been reported in the technical report (Mishra and Sepulchre, 2014b) and is currently under review.

- We apply the metric tuning idea to the low-rank matrix completion problem by proposing *two* Riemannian metric structures on the fixed-rank manifold (the set of fixed-rank matrices) in Chapter 4. Concrete implementation of the Riemannian conjugate-gradient algorithms are shown with a detailed numerical cost analysis. Our proposed algorithms generalize to a number of state-of-the-art algorithms in this field.

  The material of Chapter 4 is reported in the technical reports (Mishra and Sepulchre, 2014a; Mishra et al., 2012). The technical report (Mishra and Sepulchre, 2014a) has, subsequently, been accepted for publication in the *Proceedings of the 53rd IEEE Conference on Decision and Control, 2014*.

- The problem of large-scale optimization with low-rank regularization is addressed in Chapter 5. We propose a numerically efficient algorithm that alternates between fixed-rank optimization and rank-one updates. The fixed-rank optimization is characterized by an efficient matrix factorization. In many regularization problems, often a requirement is to compute a grid of solutions, also called a *regularization path*, corresponding to different values of the regularization parameter. The manifold setup is exploited to construct a regularization path efficiently. Finally, the performance of the proposed algorithm is illustrated on the least-squares problems of low-rank matrix completion and multivariate linear regression.

  This work has has been published in the *SIAM Journal on Optimization, 2013* (Mishra et al., 2013).

- On the algorithmic side, we deal with both first-order (steepest-descent and conjugate-gradients) and second-order (trust-regions) methods in the thesis. The matrix representations of all the optimization-related ingredients are tabulated for all the considered problems. It readily allows for an implementation of all the algorithms proposed in this thesis in Manopt, the Matlab toolbox for optimization on manifolds (Boumal et al., 2014).

  A contribution of the thesis is the development of the Manopt toolbox and is presented in a paper published in the *Journal of Machine Learning Research, 2014* (Boumal et al., 2014). Specific algorithmic implementations in Manopt that are related to the thesis are also available from `http://www.montefiore.ulg.ac.be/~mishra/`.

In addition, the following publications are related to the work done during the course of the PhD study, but are not discussed in the thesis.

- Mishra B, Vandereycken B (2014) A Riemannian approach to low-rank algebraic Riccati equations. In: Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems (MTNS), pp 965–968 (Mishra and Vandereycken, 2014).

- Mishra B, Meyer G, Sepulchre R (2011) Low-rank optimization for distance matrix completion. In: Proceedings of the 50th IEEE Conference on Decision and Control (CDC-ECC), pp 4455–4460 (Mishra et al., 2011).

1.2 Brief outline of the thesis

The outline of the thesis is as follows. In Chapter 2, we motivate the low-rank matrix completion and the generalized eigenvalue problems as least-squares problems on manifolds of symmetries. The need for taking symmetries in optimization problems into account is emphasized. Chapter 3 first reviews the Riemannian optimization framework and its connection with standard sequential quadratic programming. Building upon this connection, we address the question of selecting a Riemannian metric that exploits both the cost function nature, specifically least-squares, and the constraints. It is shown that for the particular case of quadratic optimization with rank and orthogonality constraints, the chosen metrics admit a simpler form. The metric tuning idea of Chapter 3 leads to novel conjugate-gradient algorithms for the low-rank matrix completion problem which is the subject of Chapter 4. Chapter 5 focuses on the general large-scale convex trace norm minimization problem, where a low-rank constraint is enforced "softly". Efficient computation of a regularization path of solutions is shown by exploiting the structure of the trace norm constraint. Finally, the main ideas of the thesis are summarized in Chapter 6 along with few future research perspectives.

1.3 Abbreviations and notations

Most of the abbreviations, notations, and acronyms are defined within the text where they are used. However for the sake of completeness, we list here the notations that are frequently used in this thesis.

We are concerned with the following sets of matrices with real entries.

$\mathbb{R}^n$	The set of all n-dimensional real column vectors.
$\mathbb{R}^{n\times m}$	The set of all $n \times m$ real matrices.
$\mathbb{R}_r^{n\times m}$	The set of rank-r matrices of size $n \times m$ and $r \leq \min(n, m)$.
$\mathrm{S}_+(r, n)$	The set of symmetric positive semidefinite matrices of size $n \times n$ of rank $r \leq \min(n, m)$.
$\mathbb{R}_*^{n\times r}$	The set of full column-rank matrices of size $n \times r$ with $r \leq n$.
$\mathrm{St}(r, n)$	The Stiefel manifold of full column-rank matrices of size $n \times r$ with orthonormal columns and $r \leq n$.
$\mathrm{Gr}(r, n)$	The Grassmann manifold of r-dimensional subspaces in $\mathbb{R}^n$.
$\mathrm{S}_{++}(r)$	The set of symmetric positive definite matrices of size $r \times r$.
$\mathrm{Diag}_{++}(r)$	The set of symmetric diagonal matrices of size $r \times r$ with positive entries.
$\mathcal{O}(r)$	The set orthogonal matrices of size $r \times r$ with positive entries. It is also called the orthogonal group.
$\mathrm{GL}(r)$	The general linear group of $r \times r$ matrices with non-zero determinant, i.e., full rank matrices.
$\cdot \times \cdot$	The cartesian product of two sets.

Given a matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ $(n \geq m)$, we define the following operations. Without loss generality, we also assume that $m \leq n$.

$\mathbf{X}^T$	The transpose of $\mathbf{X}$.
$\mathrm{Sym}(\mathbf{X})$	It computes $(\mathbf{X}+\mathbf{X}^T)/2$, provided $\mathbf{X}$ is a square matrix.
$\mathrm{Skew}(\mathbf{X})$	It computes $(\mathbf{X}-\mathbf{X}^T)/2$, provided $\mathbf{X}$ is a square matrix.
$\mathrm{Trace}(\mathbf{X})$	The sum of the diagonal entries of $\mathbf{X}$, provided $\mathbf{X}$ is a square matrix.
$\mathbf{X} \succ 0$	Positive definiteness of a symmetric matrix $\mathbf{X}$, i.e., all the eigenvalues of $\mathbf{X}$ are strictly positive.
$\mathbf{X} \succeq 0$	Positive semidefiniteness of a symmetric matrix $\mathbf{X}$, i.e., all the eigenvalues of $\mathbf{X}$ are nonnegative.
$\mathrm{expm}(\mathbf{X})$	The matrix exponential of $\mathbf{X}$, provided $\mathbf{X}$ is a square matrix.
$\mathrm{logm}(\mathbf{X})$	The matrix logarithm of $\mathbf{X}$, provided $\mathbf{X}$ is a square matrix.
$\mathbf{X}^{1/2}$	The matrix square root of $\mathbf{X}$, provided $\mathbf{X}$ is a square matrix. For a symmetric positive semidefinite matrix $\mathbf{X}$, we choose the unique symmetric positive definite matrix square root.
$\mathrm{uf}(\mathbf{X})$	The factor $\mathbf{U}$ of the polar decomposition of $\mathbf{X}$ such that $\mathbf{X} = \mathbf{US}$, where $\mathbf{U} \in \mathbb{R}^{n\times m}$ has orthonormal columns and $\mathbf{S} \in \mathbb{R}^{m\times m}$ is a symmetric positive semidefinite matrix. If $\mathbf{X}$ is full column-rank, then $\mathrm{uf}(\mathbf{X})$ is computed as $\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1/2}$.
$\mathrm{qf}(\mathbf{X})$	The factor $\mathbf{Q}$ of the QR decomposition of $\mathbf{X}$ such that $\mathbf{X} = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{R}^{n\times m}$ has orthonormal columns and $\mathbf{R} \in \mathbb{R}^{m\times m}$ is an upper triangular matrix.
$\langle \mathbf{X}, \mathbf{Y} \rangle$	It computes $\mathrm{Trace}(\mathbf{X}^T\mathbf{Y})$ for matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times m}$.
$\mathbf{X} \odot \mathbf{Y}$	The element-wise multiplication of matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times m}$.
$\mathrm{rank}(\mathbf{X})$	The number of non-zero singular values of $\mathbf{X}$.
$\lVert\mathbf{X}\rVert_F$	The Frobenius norm of $\mathbf{X}$, i.e., the square root of the sum of the squares of the entries of the matrix $\mathbf{X}$, $\lVert\mathbf{X}\rVert_F = \sqrt{\mathrm{Trace}(\mathbf{X}^T\mathbf{X})}$.
$\lVert\mathbf{X}\rVert_*$	The trace norm of $\mathbf{X}$, i.e., the sum of the singular values of $\mathbf{X}$.
$\lVert\mathbf{X}\rVert_{op}$	The operator norm of $\mathbf{X}$, i.e., the largest singular value of $\mathbf{X}$.

Given a differentiable manifold $\mathcal{M}$, we use the following notations.

x, y	The elements of $\mathcal{M}$.
$T_x\mathcal{M}$	The tangent space of the manifold $\mathcal{M}$ at x.
ξ_x, η_x	The tangent vectors in the tangent space $T_x\mathcal{M}$.
g_x	The Riemannian metric at $x \in \mathcal{M}$.
$g_x(\xi_x\eta_x)$	The Riemannian metric between tangent vectors ξ_x and η_x at $x \in \mathcal{M}$.
$\nabla_{\xi_x}\eta_x$	The Riemannian connection of η_x with respect to ξ_x, where $\xi_x, \eta_x \in T_x\mathcal{M}$.
$R_x(\xi_x)$	The retraction mapping at $x \in \mathcal{M}$ along $\xi_x \in T_x\mathcal{M}$.
$\mathcal{T}_{\eta_x}\xi_x$	The vector transport of the tangent vector ξ_x along the tangent vector η_x.
Ψ_x	The orthogonal projector on the tangent space $T_x\mathcal{M}$ from the ambient space.

Given a differentiable manifold $\mathcal{M}$ equipped with an equivalence relation $\sim$, we have the following additional notations.

$[x]$	The equivalence class of $x \in \mathcal{M}$, defined as $[x] := \{y \in \mathcal{M} : x \sim y\}$.

$\mathcal{M}/\sim$ The quotient manifold of $\mathcal{M}$ by $\sim$, i.e., the set of equivalence classes.

$\mathcal{V}_x$ The vertical space at $x \in \mathcal{M}$. It provides a computational representation to the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$, where $[x]$ is the equivalence class of x and $\mathcal{M}/\sim$ is the quotient manifold.

$\mathcal{H}_x$ The horizontal space at $x \in \mathcal{M}$.

Π_x The orthogonal projector onto the horizontal space $\mathcal{H}_x$ from the tangent space $T_x\mathcal{M}$.

Given a smooth function $f : \mathcal{M} \to \mathbb{R} : x \mapsto f(x)$ on the manifold $\mathcal{M}$, we use the following notations.

$\mathrm{grad}_x f$ The Riemannian gradient of the function f at $x \in \mathcal{M}$ with respect to the equipped Riemannian metric g_x.

$\mathrm{Hess}_x f[\xi_x]$ The Riemannian Hessian of the function f at $x \in \mathcal{M}$ along the the tangent vector $\xi_x \in T_x\mathcal{M}$, with respect to the equipped Riemannian metric g_x. Using the Riemannian connection notation ∇, $\mathrm{Hess}_x f[\xi_x]$ is equivalent to $\nabla_{\xi_x}\mathrm{grad}_x f$.

Given a smooth function $F : \mathbb{R}^{n\times m} \to \mathbb{R} : \mathbf{X} \mapsto F(\mathbf{X})$, we use the following notations.

$\mathrm{Grad}_{\mathbf{X}} F$ The Euclidean gradient of the function F at $\mathbf{X}$ in $\mathbb{R}^{n\times m}$.

$\mathrm{D}F(\mathbf{X})[\mathbf{Z}]$ The first-order Euclidean directional derivative of the function F along $\mathbf{Z} \in \mathbb{R}^{n\times m}$, i.e., $\lim\limits_{t\to 0}(F(\mathbf{X}+t\mathbf{Z}) - F(\mathbf{X}))/t$.

$\mathrm{D}^2F(\mathbf{X})[\mathbf{Z}]$ The second-order Euclidean directional derivative of the function F along $\mathbf{Z} \in \mathbb{R}^{n\times m}$, i.e., $\lim\limits_{t\to 0}(\mathrm{Grad}_{\mathbf{X}+t\mathbf{Z}}F - \mathrm{Grad}_{\mathbf{X}}F)/t$. It should be noted that $\mathrm{D}^2F(\mathbf{X})[\mathbf{Z}] = \mathrm{D}\mathrm{Grad}_{\mathbf{X}}F[\mathbf{Z}]$, where $\mathrm{Grad}_{\mathbf{X}}F$ is defined earlier.

We use the following acronyms in the thesis.

SVD Singular value decomposition.

SQP Sequential quadratic programming.

Chapter 2

The geometry of constraints with symmetries

The present chapter deals motivates optimization problems on quotient manifolds and stresses particular structure of rank and orthogonality constraints.

The organization of the chapter is as follows. Section 2.1 motivates the low-rank matrix completion problem and the generalized eigenvalue problem. Specific optimization formulations are listed. The quotient geometrical structure of rank and orthogonality constraints is presented in Section 2.3. In particular, we show that the quotient structure arises from the interplay of well-studied matrix manifolds. Finally, the need for an optimization framework to deal with optimization with symmetries is discussed in Section 2.3.

2.1 Motivation

In this section, we specifically motivate the low-rank matrix completion problem, a least-squares problem with rank constraint and the generalized eigenvalue problem, a least-squares problem with orthogonality constraints.

The choice of these two examples rests on the fact that these are fundamental problems in many applications (Absil et al., 2002; Edelman et al., 1998; Meyer, 2011; Vandereycken, 2010). Consequently, these problems offer a number of benchmark algorithms to compare and contrast. A study of these optimization problems also allows us to extend the basic ideas that we propose in the subsequent chapters to other related problems of least-squares.

2.1.1 The low-rank matrix completion problem: a least-squares problem with rank constraint

The problem of *low-rank matrix completion* amounts to estimating the missing entries of a matrix from a limited number of its entries. Let $\mathbf{X}^\star \in \mathbb{R}^{n\times m}$ be a matrix whose entries $\mathbf{X}^\star_{ij}$ are only given for some indices $(i,j) \in \Omega$, where Ω is a subset of the complete set of indices $\{(i,j) : i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,m\}\}$. The completion problem amounts to identifying the unknown entries of $\mathbf{X}^\star$ with the assumption that the matrix to recover is low-rank, i.e., the rank $r \ll (n,m)$. The low-rank constraint captures redundant patterns in $\mathbf{X}^\star$ and ties the known and unknown entries together. Usually in problems, the number of given entries $|\Omega|$ is of order $O(nr + mr - r^2)$ which corresponds to the dimension of rank-r matrices that is much smaller than nm, the total number of entries in $\mathbf{X}^\star$, where $r \ll \min(n,m)$.

There has been a large number of research contributions on this subject over the last few years, addressing the problem both from a theoretical (Candès and Recht, 2009; Gross, 2011; Keshavan et al., 2010) and from an algorithmic point of view (Boumal and Absil, 2011; Cai et al., 2010; Jain et al., 2010; Keshavan et al., 2010; Lee and Bresler, 2010; Mazumder et al., 2010; Meka et al., 2009; Ngo and Saad, 2012; Rennie and Srebro, 2005; Simonsson and Eldén, 2010; Vandereycken, 2013). An important and popular application of the low-rank matrix completion problem is in movie recommendation systems (mentioned earlier in Chapter 1). The matrix to complete is a matrix of movie ratings of different users; a very sparse matrix with few ratings per user. The predictions of unknown ratings with a low-rank prior would have the interpretation that users' preferences only depend on few latent *genres* (MovieLens, 1997; Netflix, 2006).

In the optimization setup, the problem of low-rank matrix completion translates to solving for $\mathbf{X} \in \mathbb{R}^{n\times m}$ by *minimizing the rank of* $\mathbf{X}$ *while best fitting with the known entries in* $\mathbf{X}^\star$. Equivalently,

$$\begin{array}{lll} \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times m}} & \mathrm{rank}(\mathbf{X}) & \\ \text{subject to} & \mathbf{X}_{ij} = \mathbf{X}^\star_{ij} & \text{for all } (i,j) \in \Omega, \end{array} \tag{2.1}$$

where $\mathrm{rank}(\mathbf{X})$ is the rank of the matrix $\mathbf{X}$.

The above formulation, although intuitive, is difficult to solve as noted by Cai et al. (2010); Candès and Recht (2009). As a result, several practically useful formulations, including numerical implementations, to the problem (2.1) have been considered in the literature. A list of algorithms is collected by Carron (2014).

Two popular formulations of the low-rank matrix completion problem that encompass a number of recent contributions are presented later in this section. A first formulation is obtained by fixing the rank *explicitly*. The second formulation is obtained when the rank constraint is replaced by its *convex surrogate*, the trace norm (Fazel, 2002, Chapter 5). Both these formulations have been well-studied and recent contributions provide conditions on the number of known entries under which an *exact reconstruction* is possible from the sampled entries. Notable are the papers by Candès and Recht (2009); Gross (2011); Keshavan et al. (2010). Both these formulations also *highlight* the role of fixed-rank matrix factorizations in the low-rank matrix completion problem. In Section 2.2.1 we present few popular matrix factorizations and give an overview of their geometry. In particular, we show that the factorizations have *non-uniqueness* that make the solutions of (2.2) non-isolated on the matrix factorization search space.

Formulation with the fixed-rank constraint

The low-rank matrix completion problem with the fixed-rank formulation is

$$\begin{array}{ll} \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times m}} & \sum\limits_{(i,j)\in\Omega} (\mathbf{X}_{ij} - \mathbf{X}^{\star}_{ij})^2 \\ \text{subject to} & \operatorname{rank}(\mathbf{X}) = r, \end{array} \tag{2.2}$$

where Ω is the set of indices of the known entries of $\mathbf{X}^{\star}$ and $|\Omega|$ is the cardinality of Ω, i.e., it is equal to the number of known entries.

It should be noted that the problem (2.2) is a nonconvex problem (the set of rank-r matrices is not a convex set). The problem (2.2) admits a closed-form solution *only* when the set Ω contains all the indices of $\mathbf{X}^{\star}$, that is, when all the entries of $\mathbf{X}^{\star}$ are known. This corresponds to the classical rank-r singular value decomposition of $\mathbf{X}^{\star}$ (Golub and Van Loan, 1996). In all other cases, the problem (2.2) calls for iterative algorithms. In principle, only *critical points* and not global minima of the problem (2.2) are to be expected with any iterative optimization algorithm. Despite the nonconvexity, the formulation (2.2) is often preferred in many applications. Primarily, it has the main advantage of drastically reducing the number of search variables from nm to $nr + mr - r^2$ (the dimension of the set of rank-r matrices of size $n \times m$) which is especially the case when $r \ll \min(n, m)$. The other observation is that the problem formulation (2.2) gives *good results* in many practical problems, e.g., the works that specifically exploit the fixed-rank formulation (2.2) include (Boumal and Absil, 2011, 2012; Keshavan et al., 2010; Meyer et al., 2011a; Ngo and Saad, 2012; Vandereycken, 2013; Wen et al., 2012).

Later in Section 2.2.1, we characterize the set of rank-r matrices of size $n \times m$ precisely by invoking *fixed-rank matrix factorizations*. In particular, it is characterized as a smooth quotient manifold (discussed later in Section 2.2.1). Chapter 4 exploits this smooth manifold structure (of the set of fixed-rank matrices) to propose numerically efficient algorithms for (2.2) that compete effectively with state-of-the-art algorithms on various benchmarks.

Formulation with the convex relaxation approach

A second formulation for the low-rank matrix completion is

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} \quad \sum_{(i,j)\in\Omega} (\mathbf{X}_{ij} - \mathbf{X}^{\star}_{ij})^2 + \lambda \|\mathbf{X}\|_*, \tag{2.3}$$

where $\|\mathbf{X}\|_*$ is the trace norm of $\mathbf{X}$, i.e., the sum of the singular values of $\mathbf{X}$ (Cai et al., 2010; Fazel, 2002; Recht et al., 2010), $\lambda > 0$ is the regularization parameter, and Ω is the set of indices of the known entries of $\mathbf{X}^{\star}$.

The trace norm is a convex alternative to the fixed-rank constraint that induces low-rank solutions *implicitly* instead of the explicit enforcement, as is the case in (2.2). Enforcement of low-rank solutions in (2.3) by having a sufficiently large value of λ. Tuning the value of λ also provides a *trade-off* between minimizing data-fitting error (fitting with the known entries) and minimizing the rank of solutions. A large value of λ would seek very low-rank solutions. On the other hand, a smaller value of λ would seek better data-fitting with the known entries.

Overall, the problem (2.3) is a convex optimization problem that has the structure of a smooth convex least-squares cost $\sum\limits_{(i,j)\in\Omega}(\mathbf{X}_{ij}-\mathbf{X}^{\star}_{ij})^2$ and nonsmooth-convex term $\|\mathbf{X}\|_*$. Numerically efficient convex optimization algorithms with strong *global* and *local* convergence guarantees exist for such problems (Ma et al., 2011; Nesterov, 2003; Toh and Yun, 2010). The works that specifically exploit the trace norm formulation (2.3) to propose efficient convex optimization algorithms are, among others, from Cai et al. (2010); Ma et al. (2011); Mazumder et al. (2010); Toh and Yun (2010).

A critical issue in most convex optimization algorithms that directly deal with (2.3) is that the ranks of intermediate iterates seem to be uncontrolled and only asymptotically, a low-rank solution is expected. This poses significant practical challenges when dealing with large-scale instances. To circumvent this issue, *low-rank projection* of iterates, that is, curtailing smaller singular values of iterates, is often recommend to accelerate convergence of the algorithms (Toh and Yun, 2010). This observation suggests that combining trace norm $\|\mathbf{X}\|_*$ (implicit enforcement of low-rank solutions) and fixed-rank constraint (explicit enforcement of low-rank solutions) is beneficial in large-scale algorithmic implementations. This is the topic of discussion in Chapter 5, where we exploit the manifold structure of the set of fixed-rank matrices (discussed later in Section 2.2.1) to our advantage in tackling the trace norm regularization problem in a very general setup. We also revisit the trace norm regularized matrix completion problem (2.3) in Section 5.6.2. A second advantage of combining trace norm and fixed-rank constraints in optimization is that it leads to an efficient construction of the *regularization path of solutions*, discussed in Section 5.5. The regularization path of solutions is the sequence of minimizers of (2.3) corresponding to different values of λ.

2.1.2 The generalized eigenvalue problem: a least-squares problem with orthogonality constraints

A second example of interest in this thesis is the *generalized eigenvalue problem* that seeks to compute the extreme eigenvalues and eigenvectors of a matrix. In an optimization setup, this amounts to solving the least-squares problem

$$\begin{array}{ll}\min\limits_{\mathbf{X}\in\mathbb{R}^{n\times r}} & \frac{1}{2}\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})\\ \text{subject to} & \mathbf{X}^T\mathbf{B}\mathbf{X}=\mathbf{I},\end{array}\tag{2.4}$$

where $\mathbf{A}$ is a symmetric $n\times n$ matrix and $\mathbf{B}$ is an $n\times n$ symmetric positive definite matrix. The constraint $\mathbf{X}^T\mathbf{B}\mathbf{X}=\mathbf{I}$ is called the *orthogonality constraint* that imposes orthogonality among the columns of $\mathbf{X}$. It is well-known that the global minimizer of the problem (2.4) is the smallest r-dimensional subspace of the matrix $\mathbf{B}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}^{-\frac{1}{2}}$ (Absil et al., 2008, Proposition 2.1.1).

The problem (2.4) has attracted much interest in the numerical optimization community and is fundamental in a wide range of applications that require *extremal* (dominant or smallest) eigenspace information (Absil, 2003). For example, when $\mathbf{B}$ is $\mathbf{I}$ and $\mathbf{A}\succeq 0$, the maximization of $\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})$ with $\mathbf{X}^T\mathbf{X}=\mathbf{I}$

leads to identification of the *principal components* of $\mathbf{A}$, a popular tool in statistical analysis. See (Journée, 2009) and the references therein for a recent survey on the topic. Similarly, the *nonsymmetric* extension of (2.4) leads to identifying singular vectors of a matrix, i.e., the formulation

$$\begin{array}{ll} \min\limits_{\substack{\mathbf{U}\in\mathbb{R}^{n\times r}\\ \mathbf{V}\in\mathbb{R}^{m\times r}}} & \frac{1}{2}\mathrm{Trace}(\mathbf{U}^T\mathbf{A}\mathbf{V}) \\ \text{subject to} & \mathbf{U}^T\mathbf{U}=\mathbf{I} \\ & \mathbf{V}^T\mathbf{V}=\mathbf{I}, \end{array} \tag{2.5}$$

where $\mathbf{A}\in\mathbb{R}^{n\times m}$. The solution to (2.5) computes the *dominant singular vectors* of $\mathbf{A}$.

It should be noted that the cost, $\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})$ in (2.4), remains invariant under the transformation $\mathbf{X}\mapsto\mathbf{X}\mathbf{O}$ for all matrices $\mathbf{O}\in\mathcal{O}(r)$ (the orthogonal group, that is, the set of $r\times r$ such that $\mathbf{O}^T\mathbf{O}=\mathbf{O}\mathbf{O}^T=\mathbf{I}$). As a consequence, the minimizers of the problem (2.4) are *not isolated*, i.e., if $\mathbf{X}^\star$ is a solution to (2.4), then all the matrices $\mathbf{X}\mathbf{O}$ for $\mathbf{O}\in\mathcal{O}(r)$ are also the solutions. This has profound implications in the convergence analysis of any optimization algorithm for (2.4) (Absil et al., 2008; Edelman et al., 1998; Nocedal and Wright, 2006). To take the symmetry into account, the search space is reduced in dimension. In Section 2.2.2 we characterize the search space of (2.4) concretely to deal with the invariance of the cost. We provide an algorithmic treatment to (2.4) in Section 3.3.

2.2 The characterization of rank and orthogonality constraints

Both fixed-rank matrices and orthogonality constraints have underlying symmetries that are structured. In particular, the search spaces spanned by the set of fixed-rank matrices and by the set of orthogonality constraints have the structure of a *quotient space*, that result from equivalence relations (Absil et al., 2002; Edelman et al., 1998; Meyer, 2011). The quotient spaces are abstract spaces. Working with a quotient space calls, by necessity, for a computational space where the elements have matrix representations. This space is called the *total space*.

We discuss the quotient nature of rank and orthogonality constraints below.

2.2.1 The quotient nature of rank constraint

We review three popular matrix factorizations for fixed-rank non-symmetric matrices that parameterize the rank constraint, e.g., $\mathrm{rank}(\mathbf{X})=r$, where $\mathbf{X}\in\mathbb{R}^{n\times m}$. The fixed-rank matrix factorizations result from the *thin* singular value decomposition (SVD) of a rank-r, i.e., rank deficient, matrix $\mathbf{X}\in\mathbb{R}^{n\times m}$ (Golub and Van Loan, 1996, Theorems 2.5.2 and 2.5.3). Figure 2.1 shows the matrix factorization schemes that are of relevance to this thesis. Specifically, the SVD of a rank-r $\mathbf{X}\in\mathbb{R}^{n\times m}$ decomposes it into the product of three matrices as

$$\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \tag{2.6}$$

where $\mathbf{U}$ is an $n\times r$ matrix with orthogonal columns, that is, an element of the *Stiefel manifold* $\mathrm{St}(r,n)=\{\mathbf{U}\in\mathbb{R}^{n\times r}:\ \mathbf{U}^T\mathbf{U}=\mathbf{I}\}$, $\mathbf{V}\in\mathrm{St}(r,m)$, and $\mathbf{\Sigma}\in\mathrm{Diag}_{++}(r)$ is a $r\times r$ diagonal matrix with positive entries called the *singular values* (which are ordered). $\mathbf{I}$ is the identity matrix (with appropriate dimensions).

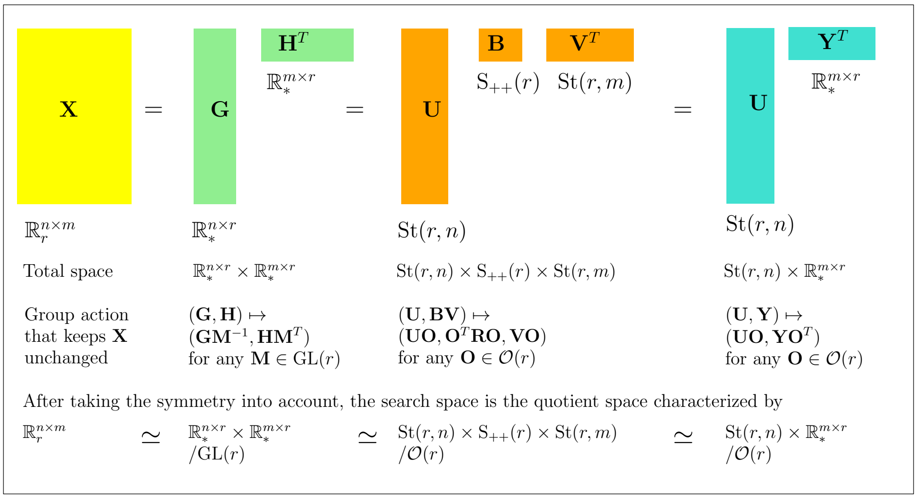

$\mathbf{X}$
$=$
$\mathbf{G}$
$\mathbf{H}^T$
$\mathbb{R}_*^{m\times r}$
$=$
$\mathbf{U}$
$\mathbf{B}$
$\mathbf{V}^T$
$\mathrm{S}_{++}(r)$
$\mathrm{St}(r,m)$
$=$
$\mathbf{U}$
$\mathbf{Y}^T$
$\mathbb{R}_*^{m\times r}$
$\mathbb{R}_r^{n\times m}$
$\mathbb{R}_*^{n\times r}$
$\mathrm{St}(r,n)$
$\mathrm{St}(r,n)$
Total space
$\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$
$\mathrm{St}(r,n)\times\mathrm{S}_{++}(r)\times\mathrm{St}(r,m)$
$\mathrm{St}(r,n)\times\mathbb{R}_*^{m\times r}$
Group action that keeps $\mathbf{X}$ unchanged
$(\mathbf{G},\mathbf{H})\mapsto(\mathbf{G}\mathbf{M}^{-1},\mathbf{H}\mathbf{M}^T)$ for any $\mathbf{M}\in\mathrm{GL}(r)$
$(\mathbf{U},\mathbf{B}\mathbf{V})\mapsto(\mathbf{U}\mathbf{O},\mathbf{O}^T\mathbf{R}\mathbf{O},\mathbf{V}\mathbf{O})$ for any $\mathbf{O}\in\mathcal{O}(r)$
$(\mathbf{U},\mathbf{Y})\mapsto(\mathbf{U}\mathbf{O},\mathbf{Y}\mathbf{O}^T)$ for any $\mathbf{O}\in\mathcal{O}(r)$
After taking the symmetry into account, the search space is the quotient space characterized by
$\mathbb{R}_r^{n\times m}$
$\simeq$
$\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}/\mathrm{GL}(r)$
$\simeq$
$\mathrm{St}(r,n)\times\mathrm{S}_{++}(r)\times\mathrm{St}(r,m)/\mathcal{O}(r)$
$\simeq$
$\mathrm{St}(r,n)\times\mathbb{R}_*^{m\times r}/\mathcal{O}(r)$

FIGURE 2.1: Fixed-rank matrix factorizations lead to quotient search spaces due to their intrinsic *symmetries*. The pictures emphasize the situation of interest, i.e., the rank r is small compared to the matrix dimensions. Matrix factorizations admit product structures of well-studied differentiable manifolds $\mathrm{St}(r,n)$, $\mathbb{R}_*^{n\times r}$, $\mathrm{GL}(r)$, and $\mathrm{S}_{++}(r)$ (Bhatia, 2007; Edelman et al., 1998). Similarly, the group actions are by $\mathrm{GL}(r)$ and $\mathcal{O}(r)$ with well-known characterizations (Lee, 2003).

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

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

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

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

$$(\mathbf{G},\mathbf{H}) \mapsto (\mathbf{G}\mathbf{M}^{-1},\mathbf{H}\mathbf{M}^T), \tag{2.7}$$

where $\mathbf{M} \in \mathrm{GL}(r) := \{\mathbf{M}\in\mathbb{R}^{r\times r} : \mathrm{determinant}(\mathbf{M}) \neq 0\}$, leaves the original matrix $\mathbf{X}$ unchanged (Piziak and Odell, 1999). This symmetry comes from the fact that the row and column spaces are *invariant to change of coordinates.* The classical remedy to remove this indeterminacy in the case of symmetric positive semidefinite matrices is the *Cholesky factorization*, which imposes further (triangular-like) structure in the factors (Golub and Van Loan, 1996, Section 4.2). The *LU decomposition* plays a similar role for non-symmetric matrices (Jeffrey, 2010). In a manifold setting, we instead encode the invariance map (2.7) in an abstract search space by optimizing over a set of equivalence classes defined as

$$[(\mathbf{G},\mathbf{H})] = \{(\mathbf{G}\mathbf{M}^{-1},\mathbf{H}\mathbf{M}^T) : \mathbf{M}\in\mathrm{GL}(r)\}, \tag{2.8}$$

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

$$\mathbb{R}_r^{n\times m} \simeq \mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}/\mathrm{GL}(r). \tag{2.9}$$

The product space $\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$ is called the *total space* or the computational space. The set $\mathrm{GL}(r)$ is called the *fiber space.* The set of equivalence classes $[(\mathbf{G},\mathbf{H})]$ is called the quotient space.

2.2.1.2 Polar factorization (beyond SVD)

The second quotient structure for the set $\mathbb{R}_r^{n\times m}$ is obtained by considering the following group action on the SVD (Bonnabel and Sepulchre, 2009; Meyer, 2011),

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

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

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

This results in the *polar factorization*

$$\mathbf{X} = \mathbf{UBV}^T, \tag{2.10}$$

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

$$\mathrm{S}_{++}(r) := \{\mathbf{B}\in\mathbb{R}^{r\times r} : \mathbf{B}^T = \mathbf{B}\succ 0\}.$$

The polar factorization reflects the original geometric purpose of singular value decomposition as representing an arbitrary linear transformation as the composition of two *isometries* and a *scaling* (Golub and Van Loan, 1996, Section 2.5.3). Allowing the scaling $\mathbf{B}$ to be positive definite rather than diagonal gives more flexibility to optimization algorithms and removes the discrete symmetries induced by interchanging the order on the singular values. Empirical evidence to support the choice of $\mathrm{S}_{++}(r)$ over $\mathrm{Diag}_{++}(r)$ (the set of diagonal matrices with positive entries) for the middle factor $\mathbf{B}$ is shown in Section 5.6.1. The resulting search space is again the set of equivalence classes defined by

$$[(\mathbf{U},\mathbf{B},\mathbf{V})] = \{(\mathbf{UO},\mathbf{O}^T\mathbf{BO},\mathbf{VO}) :\ \mathbf{O}\in\mathcal{O}(r)\}. \tag{2.11}$$

The total space is now $\mathrm{St}(r,n)\times\mathrm{S}_{++}(r)\times\mathrm{St}(r,m)$. The fiber space is $\mathcal{O}(r)$ and the resulting quotient space is the set of equivalence classes

$$\mathbb{R}_r^{n\times m} \simeq \mathrm{St}(r,n)\times\mathrm{S}_{++}(r)\times\mathrm{St}(r,m)/\mathcal{O}(r). \tag{2.12}$$

Yet another factorization is obtained by defining the group action

$$(\mathbf{U},\mathbf{\Sigma},\mathbf{V}) \mapsto (\mathbf{UO}_1,\mathbf{O}_1^T\mathbf{\Sigma}\mathbf{O}_2,\mathbf{VO}_2)$$

on the SVD, where $\mathbf{O}_1,\mathbf{O}_2\in\mathcal{O}(r)$. This results in the factorization

$$\mathbf{X} = \mathbf{URV}^T, \tag{2.13}$$

where $\mathbf{U}\in\mathrm{St}(r,n)$, $\mathbf{V}\in\mathrm{St}(r,m)$, and $\mathbf{R}\in\mathrm{GL}(r)$. This further relaxes the symmetric positive constraint on the factor $\mathbf{B}$ of (2.10) while retaining separation between isometries and scaling. The total space is now $\mathrm{St}(r,n)\times\mathrm{GL}(r)\times\mathrm{St}(r,m)$. The fiber space is $\mathcal{O}(r)\times\mathcal{O}(r)$ and the resulting quotient space is the set of equivalence classes

$$\mathbb{R}_r^{n\times m} \simeq \mathrm{St}(r,n) \times \mathrm{GL}(r) \times \mathrm{St}(r,m)/(\mathcal{O}(r) \times \mathcal{O}(r)). \tag{2.14}$$

2.2.1.3 Subspace-projection factorization (beyond QR decomposition)

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

$$\mathbf{X} = \mathbf{U}(\mathbf{\Sigma}\mathbf{V}^T) = \mathbf{U}\mathbf{Y}^T,$$

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

$$[(\mathbf{U},\mathbf{Y})] = \{(\mathbf{UO},\mathbf{YO}) :\ \mathbf{O} \in \mathcal{O}(r)\}. \tag{2.15}$$

The search space is the quotient space

$$\mathbb{R}_r^{n\times m} \simeq \mathrm{St}(r,n) \times \mathbb{R}_*^{m\times r}/\mathcal{O}(r), \tag{2.16}$$

where the total space is $\mathrm{St}(r,n) \times \mathbb{R}_*^{m\times r}$ and the fiber space is $\mathcal{O}(r)$. Recent contributions exploiting this factorization include the works of Boumal and Absil (2011); Dai et al. (2011); Simonsson and Eldén (2010).

2.2.2 The quotient nature of orthogonality constraints

The basis of symmetry in the set of orthogonality constraints in (2.4) is the invariance of the cost function, $\mathrm{Trace}(\mathbf{X}^T\mathbf{AX})$, with the transformation $\mathbf{X} \mapsto \mathbf{XO}$ for all matrices $\mathbf{O} \in \mathcal{O}(r)$ (the orthogonal group), where $\mathbf{X} \in \mathrm{St}(r,n)$. This implies that the search space in (2.4) is not set spanned by $\mathbf{X}^T\mathbf{BX} = \mathbf{I}$ but rather the set of equivalence classes

$$[\mathbf{X}] := \{\mathbf{XO} : \mathbf{O} \in \mathcal{O}(r)\}.$$

Consequently, the search space of (2.4) is characterized as the quotient space $\mathrm{St}_{\mathbf{B}}(r,n)/\mathcal{O}(r)$, where $\mathrm{St}_{\mathbf{B}}(r,n) := \{\mathbf{X} \in \mathbb{R}^{n\times r} : \mathbf{X}^T\mathbf{BX} = \mathbf{I}\}$. The total space, that is, the computational space is $\mathrm{St}_{\mathbf{B}}(r,n)$ and the fiber space is the set $\mathcal{O}(r)$.

For the specific case when $\mathbf{B} = \mathbf{I}$, the quotient space $\mathrm{St}_{\mathbf{B}}(r,n)/\mathcal{O}(r)$ is the *Grassmann manifold* $\mathrm{Gr}(r,n)$, the set of r-dimensional subspaces in $\mathbb{R}^n$ (Edelman et al., 1998). It should be mentioned that there exists other characterizations of the Grassmann manifold, similar to different characterizations of the rank constraint in Section 2.2.1. We point to one particular characterization of the Grassmann manifold presented by Absil et al. (2004a), where the Grassmann manifold $\mathrm{Gr}(r,n)$ is identified with the quotient

Rank constraint	**Orthogonality constraints**
$\mathrm{St}(r,n)$	$\mathrm{St}(r,n)$
$\mathbb{R}_*^{n\times r}$	$\mathbb{R}_*^{n\times r}$
$\mathcal{O}(r)$	$\mathcal{O}(r)$
$\mathrm{GL}(r)$	$\mathrm{GL}(r)$
$\mathrm{S}_{++}(r)$	

Figure 2.2: A list of matrix manifolds that appear in rank and orthogonality constraints. Individually, each of them is well-studied. The notations can be followed from Section 1.3.

space $\mathbb{R}_*^{n\times r}/\mathrm{GL}(r)$. Here $\mathbb{R}_*^{n\times r}$ is the set of full column-rank matrices of size $n \times r$ and $\mathrm{GL}(r)$ is the set of non-singular $r \times r$ matrices.

2.3 Optimization on matrix manifolds with symmetries

Following the previous section it is clear that the problems of low-rank matrix completion with fixed-rank (Section 2.1.1) and generalized eigenvalue computation (Section 2.1.2) should be treated as *optimization problems on quotient spaces*. Additionally, these quotient spaces result from the *interplay* of individually well-studied matrix manifolds, shown in Figure 2.2. More appropriately, the resulting quotient spaces of rank and orthogonality constraints have the structure of *quotient manifolds* (Lee, 2003, Theorem 9.16; Absil et al., 2008, Section 3.4.1; Meyer, 2011). To understand why taking the quotient structure into account in optimization problems is critical, we consider the *Rayleigh quotient minimization* problem (Golub and Van Loan, 1996, Chapter 8)

$$\min_{x\in\mathbb{R}_*^n} \quad \frac{x^T\mathbf{A}x}{x^Tx} \tag{2.17}$$

which computes the *smallest* (algebraic) eigenvalue-eigenvector pair of the symmetric $n \times n$ matrix $\mathbf{A}$ (simplicity of the smallest eigenvalue is assumed), where $\mathbb{R}_*^n := \mathbb{R}^n - \{0\}$. It should be stated the problem (2.17) is a particular case of the generalized eigenvalue problem (2.4). The cost function in (2.17) is invariant to multiplication of x by a non-zero scalar, i.e., $x \to \alpha x$ for $\alpha \in \mathbb{R} - \{0\}$ keeps the cost function unchanged. This symmetry is reflected in the Newton iteration for (2.17) which yields the iteration $x \to 2x$ implying that the failure of the Newton method when symmetry is not taken into account (Absil et al., 2008, Proposition 2.1.2). In fact, this result is not particular to the Rayleigh quotient problem (2.1) but rather holds for any *homogenous function of degree zero* (Absil et al., 2008, Section 2.1.1). Indeed, the search space in (2.17) is the set of equivalence classes $[x] = \{\alpha x : \alpha \in \mathbb{R} - \{0\}\}$ for all $x \in \mathbb{R}_*^n$, rather than the set $\mathbb{R}_*^n$. This set of equivalence classes is the *real projective space*, $\mathbb{R}_*^n/\mathbb{R}_*$.

Any iterative optimization algorithm on a quotient manifold involves computing a search direction (e.g., the gradient direction) and then "moving in that direction". Both these optimization-related operations admit *simple matrix representations* in the Riemannian optimization framework (Absil et al., 2008;

Edelman et al., 1998). The first step is to endow the search space, that is a quotient space in our case, with a Riemannian manifold structure and a *metric* (a smooth inner product on the manifold). Once the structure is in place, the Riemannian framework conceptually transforms a constrained optimization problem like (2.2) into an *unconstrained* optimization problem on a Riemannian quotient manifold. Consequently, all unconstrained optimization algorithms can be extended to the Riemannian setup. The monograph from Absil et al. (2008) provides a systematic introduction to the Riemannian optimization framework with a list of algorithms that come with rigorous convergences guarantees.

The subsequent chapters of this thesis exploit the Riemannian optimization framework for the problems (2.1.1) and (2.1.2). In Chapter 3, along with the quotient nature of the search space, we specifically exploit the least-squares nature of the cost function to define a number of useful metrics on the search space.

2.4 Chapter summary

In this chapter, we have presented the optimization problems of low-rank matrix completion and generalized eigenvalue computation. These are motivated as optimization problems on quotient manifolds that arise from structured symmetries in the search space. The quotient nature of rank and orthogonality constraints are discussed in Section 2.3. We have emphasized that the quotient structure is captured by the interplay of few well-studied matrix manifolds shown in Figure 2.2. Finally, the motivation for the Riemannian optimization framework is presented.

The main theme of the chapter is based on the publication (Mishra et al., 2014).

Chapter 3

Metric tuning in Riemannian optimization and its application to least-squares problems

In this chapter, we exploit a basic connection between sequential quadratic programming and Riemannian gradient optimization to address the general question of selecting a *metric* in Riemannian optimization. The proposed method is shown to be particularly insightful and efficient in quadratic optimization with orthogonality and/or rank constraints, which covers most current applications of Riemannian optimization in matrix manifolds. We view this approach of selecting a metric from SQP as a form of *Riemannian preconditioning*. Similar to the notion of preconditioning in the unconstrained case (Nocedal and Wright, 2006, Chapter 5), the chosen Riemannian metrics have a preconditioning effect on optimization algorithms. We do not aim at a comprehensive treatment on the topic but rather focus on connections between several classical branches of matrix calculus: matrix factorizations and shifts in numerical linear algebra, Riemannian submersions in differential geometry, and sequential quadratic programming in constrained optimization.

The organization of the chapter is as follows. A brief motivation of metric tuning is presented in Section 3.1. The general idea of using sequential quadratic programming (SQP) to select a metric in Riemannian optimization is presented in Section 3.2. SQP and the Riemannian optimization framework are specifically introduced in sections 3.2.1 and 3.2.2, respectively. We show that under quite general assumptions, the SQP approach defines a Riemannian metric and that sequential quadratic programming is equivalent to Riemannian steepest-descent optimization for this metric. We further discuss the choice of the metric depending on the curvature properties of both the cost and the constraint and the interpretation of the Lagrange parameter as a shift. Section 3.3 develops the specific situation of quadratic cost and orthogonality constraints, revisiting the classical generalized eigenvalue problem. Section 3.4 further develops the specific situation of quadratic cost and rank constraints, with applications to solving large-scale matrix Lyapunov equations. All numerical illustrations use the Matlab toolbox Manopt (Boumal et al., 2014).

3.1 Motivation

Gradient algorithms are a method of choice for large-scale optimization, but their convergence properties critically depend on the choice of a suitable *metric*. Good adaptive metrics can lead to *superlinear* convergence whereas bad metrics can lead to very slow convergence (Nocedal and Wright, 2006, Chapter 3). The goodness of the metric depends on its ability to encode second-order information about the optimization problem. For general optimization problems with equality constraints, sequential quadratic programming (SQP) methods provide an efficient selection procedure based on (approximating) the Hessian of a local quadratic approximation of the problem (Nocedal and Wright, 2006, Chapter 18). This approach is *Lagrangian*, that is, it lifts the constraint into the cost function. An alternative is to embed the constraint into the search space, leading to unconstrained optimization on a nonlinear search space. Selecting the metric then amounts to equipping the search space with a Riemannian structure (Absil et al., 2008; Edelman et al., 1998). A current limitation of Riemannian optimization is however in the choice of the metric. Previous work has mostly focused on the search space, exploiting the differential geometry of the constraint but disregarding the role of the cost function. This limitation was pointed out early (Manton, 2002) and has been addressed in a number of recent contributions that emphasized the importance of preconditioning (Ngo and Saad, 2012; Vandereycken and Vandewalle, 2010) but with no general procedure. The simple observation of the present chapter is that sequential quadratic programming provides a systematic framework to choose a metric in Riemannian optimization in a way that takes into consideration both the cost function and the constrained search space. This connection seems novel and insightful.

The use of sequential quadratic programming to select a metric in Riemannian optimization is general and connects two rather independent areas of constrained optimization. We focus in particular on the special case of quadratic cost functions with orthogonality and/or rank constraints. This particular situation encompasses a great deal of current successful applications of Riemannian optimization, including the popular generalized eigenvalue problem (Absil et al., 2002; Edelman et al., 1998) and linear matrix equation problems (Benner and Saak, 2013; Vandereycken and Vandewalle, 2010). Even in these highly researched problems, we show that SQP methods unify a number of recent disparate results and provide novel metrics. In the eigenvalue problem, where both the cost and constraints are quadratic, the SQP method suggests a parameterized family of Riemannian metrics that provides novel insights on the role of shifts in the power, inverse, and Rayleigh quotient iteration methods. In the problem of solving linear matrix equations, low-rank matrix factorizations make the cost function quadratic in each of the factors, leading to Riemannian metrics rooted in block diagonal approximations of the Hessian. In all of the mentioned applications, we stress the complementary but not equivalent role of sequential quadratic programming and Riemannian programming: the SQP method provides a systematic procedure to select the metric while the Riemannian framework provides the necessary generalization of unconstrained optimization to quotient manifolds, allowing for rigorous design and convergence analysis of a broad class of quasi-Newton algorithms in optimization problems over *classes of equivalences* of matrices.

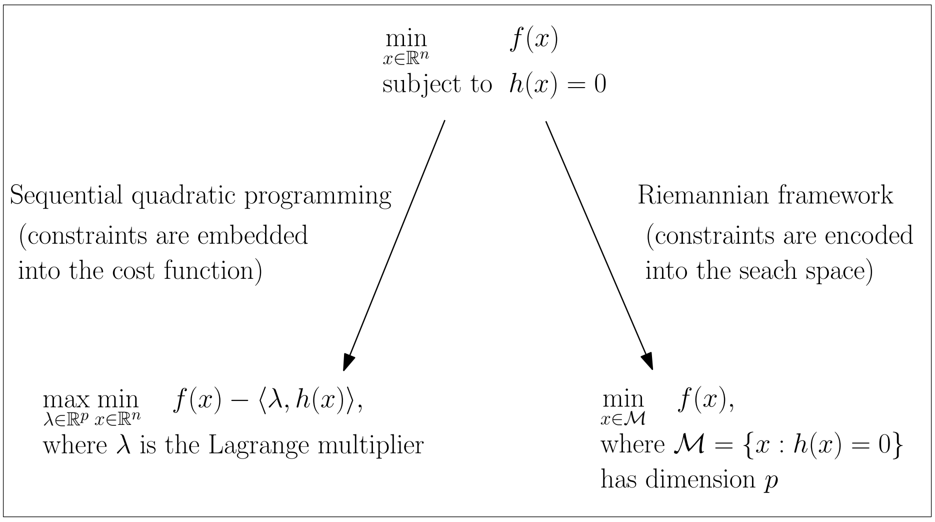

$\min_{x\in\mathbb{R}^n} \quad f(x)$
subject to $h(x) = 0$
Sequential quadratic programming
(constraints are embedded
into the cost function)
Riemannian framework
(constraints are encoded
into the seach space)
$\max_{\lambda\in\mathbb{R}^p}\min_{x\in\mathbb{R}^n} \quad f(x) - \langle\lambda, h(x)\rangle,$
where λ is the Lagrange multiplier
$\min_{x\in\mathcal{M}} \quad f(x),$
where $\mathcal{M} = \{x : h(x) = 0\}$
has dimension p

FIGURE 3.1: Two complementary viewpoints on optimization with equality constraints.

3.2 Locally selecting the metric of a gradient scheme

Consider the optimization problem

$$\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n} & f(x) \\ \text{subject to} & h(x) = 0, \end{array} \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^p$ are smooth functions. We assume that the set $\mathcal{M} = \{x : h(x) = 0\}$ has the structure of an *embedded submanifold* of $\mathbb{R}^n$ (Absil et al., 2008, Section 3.3).

In this section we discuss two complementary approaches for the problem (3.1), namely the sequential quadratic approach and the Riemannian approach. The schematic view is shown in Figure 3.1. We aim at connecting these two approaches in order to tune the metric on the search space in such a way that it incorporates *second-order* information of the problem.

3.2.1 The constrained optimization viewpoint (SQP)

Sequential quadratic programming (SQP) is a particularly well-known approach for equality constrained nonlinear optimization (Nocedal and Wright, 2006, Chapter 18). The core idea is to optimize the unconstrained *Lagrangian function* $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^p \to \mathbb{R} : (x, \lambda) \mapsto \mathcal{L}(x, \lambda)$, defined as

$$\mathcal{L}(x, \lambda) = f(x) - \langle\lambda, h(x)\rangle, \tag{3.2}$$

over the two sets of parameters, $x \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}^p$, where $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product and λ is the *Lagrange multiplier*. This leads to a *primal-dual* iterative algorithm in $(x, \lambda) \in \mathbb{R}^n \times \mathbb{R}^p$. Linearity of λ in (3.2) is further exploited to reduce the number of variables. For example, locally in the neighborhood of the minimum, the best *least-squares estimate* of the Lagrangian multiplier is

$$\lambda_x = (h_x(x)(h_x(x))^T)^{-1}h_x(x)f_x(x), \tag{3.3}$$

The SQP algorithm for

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ \text{subject to} & h(x) = 0 \end{array}$$

1. Compute the search direction ζ_x^* that is the solution of (3.4).
2. The next iterate x_+ is obtained by projecting $x + s\zeta_x^*$ onto the constrained space, where the step-size s is obtained by a backtracking line search.
3. Repeat until convergence.

Table 3.1: The SQP algorithm.

where $h_x(x)$ and $f_x(x)$ are the first-order derivatives of the functions h and f at x, respectively (Nocedal and Wright, 2006, Equation (18.20)). It should be noted that other estimates of λ near the minimum are also equally in valid in SQP (Absil et al., 2009). Substituting λ with λ_x transforms the primal-dual iteration in (x, λ) to a purely primal iteration in the variable x alone (Nocedal and Wright, 2006, Page 539). Once the Lagrangian multiplier is estimated, the SQP optimization approach proceeds by minimizing the quadratic programming problem

$$\begin{array}{ll} \operatorname*{arg\,min}\limits_{\zeta_x \in \mathbb{R}^n} & f(x) + \langle f_x(x), \zeta_x \rangle + \frac{1}{2} \langle \zeta_x, \mathrm{D}^2 \mathcal{L}(x, \lambda_x)[\zeta_x] \rangle \\ \text{subject to} & \mathrm{D}h(x)[\zeta_x] = 0 \end{array} \tag{3.4}$$

at each iteration, where $f_x(x)$ is the derivative of the cost function f, $\mathrm{D}^2\mathcal{L}_x(x, \lambda_x)[\zeta_x]$ is the second-order partial derivative of $\mathcal{L}(x, \lambda_x)$ with respect to x (keeping λ_x fixed) that is applied in the direction $\zeta_x \in \mathbb{R}^n$, $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product, and $\mathrm{D}h(x)[\zeta_x]$ is the standard Euclidean directional derivative of $h(x)$ in the direction $\zeta_x \in \mathbb{R}^n$, i.e., $\mathrm{D}h(x)[\zeta_x] = \lim_{t \to 0}(h(x + t\zeta_x) - h(x))/t$. If the quantity $\langle \zeta_x, \mathrm{D}^2\mathcal{L}(x, \lambda_x)[\zeta_x] \rangle$ is strictly positive in the tangent space of constraints, then the problem (3.4) is convex and has a unique solution (Nocedal and Wright, 2006, Section 18.1).

The next iterate x_+ in the SQP algorithm is obtained by projecting $x + \zeta_x^*$ onto the constrained space to maintain strict feasibility of the iterates, where ζ_x^* is the solution to (3.4). The resulting iterative algorithm is shown in Table 3.1 and has the properties of a quasi-Newton algorithm with favorable convergence properties (Nocedal and Wright, 2006, Section 18.3).

The SQP approach is appealing for the simplicity of its formulation. But its convergence properties asymptotically rely on the regularity of the Hessian of the Lagrangian $\mathcal{L}(x, \lambda)$ (Nocedal and Wright, 2006, Assumptions 18.2). In many applications, underlying symmetries make the Hessian of the Lagrangian singular. It is here that the quotient manifold optimization comes into play.

3.2.2 The Riemannian optimization viewpoint

The general philosophy of optimization on manifolds is to recast a constrained optimization problem in the Euclidean space into an unconstrained optimization on a nonlinear search space that encodes the constraint. For special constraints that are sufficiently structured, the framework leads to an efficient computational framework (Absil et al., 2008). This is particularly so when the constraint set is an

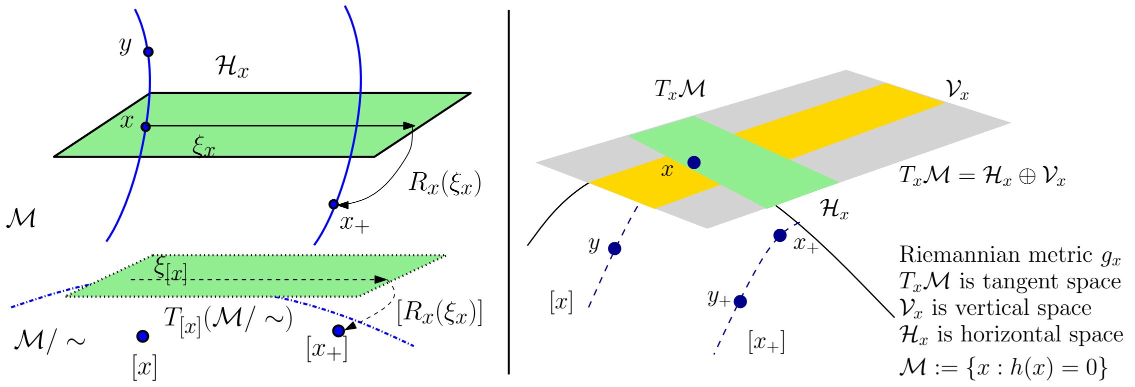

y
$\mathcal{H}_x$
x
ξ_x
$R_x(\xi_x)$
$\mathcal{M}$
x_+
$\xi_{[x]}$
$T_{[x]}(\mathcal{M}/\sim)$
$[R_x(\xi_x)]$
$\mathcal{M}/\sim$
$[x]$
$[x_+]$
$T_x\mathcal{M}$
$\mathcal{V}_x$
x
$T_x\mathcal{M} = \mathcal{H}_x \oplus \mathcal{V}_x$
$\mathcal{H}_x$
y
x_+
$[x]$
y_+
$[x_+]$
Riemannian metric g_x
$T_x\mathcal{M}$ is tangent space
$\mathcal{V}_x$ is vertical space
$\mathcal{H}_x$ is horizontal space
$\mathcal{M} := \{x : h(x) = 0\}$

FIGURE 3.2: Optimization on a Riemannian quotient manifold. The points y and x in the total (computational) space $\mathcal{M}$ belong to the same equivalence class and they represent a single point $[x] := \{y \in \mathcal{M} : y \sim x\}$ in the quotient space $\mathcal{M}/\sim$. An algorithm by necessity is implemented in the computation space but conceptually, the search is on the quotient manifold. With the Riemannian metric g (3.5), the quotient manifold $\mathcal{M}/\sim$ is *submersed* into $\mathcal{M}$. The horizontal space $\mathcal{H}_x$ provides a matrix representation to the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ of the Riemannian quotient manifold. Consequently, tangent vectors on the quotient space are *lifted* to the Horizontal space. The mapping R_x maps a horizontal vector ξ_x onto an element on the total space. Here ξ_x is the horizontal lift, i.e., matrix representation of the tangent vector $\xi_{[x]}$ on the abstract space.

embedded manifold $\mathcal{M}$ up to an equivalence relationship $\sim$. The search space is then a set of equivalence classes. Optima are not isolated in the computational *total* space $\mathcal{M}$, but they become isolated on the *quotient* space $\mathcal{M}/\sim$. If the total space can be equipped with a metric that turns both the total space $\mathcal{M}$ and its quotient space $\mathcal{M}/\sim$ into a differentiable Riemannian manifold, then any unconstrained optimization algorithm can be *realized* in the total space but *analyzed* in the quotient space (Absil et al., 2008; Edelman et al., 1998; Smith, 1994). The exposition for quotient manifolds here follows from Absil et al. (2008, Chapters 3, 5,and 8).

Consider an equivalence relation $\sim$ in the *total* (computational) space $\mathcal{M}$. The quotient manifold $\mathcal{M}/\sim$ generated by this equivalence property consists of elements that are *equivalence classes* of the form $[x] = \{y \in \mathcal{M} : y \sim x\}$. In other words, if $[x]$ is an element in $\mathcal{M}/\sim$, then its matrix representation in $\mathcal{M}$ is x. For example, the Grassmann manifold $\mathrm{Gr}(r, n)$, which is the set of r-dimensional subspaces in $\mathbb{R}^n$, is obtained by the equivalence relationship $\mathrm{Gr}(r, n) = \mathrm{St}(r, n)/\mathcal{O}(r)$. $\mathrm{St}(r, n)$ is the set of matrices of size $n \times r$ with orthogonal columns and $\mathcal{O}(r)$ is the set of square $r \times r$ *orthogonal matrices*. Each element in the total space $\mathcal{M} := \mathrm{St}(r, n) = \{\mathbf{X} \in \mathbb{R}^{n\times r} : \mathbf{X}^T\mathbf{X} = \mathbf{I}\}$ is characterized by a matrix $\mathbf{X} \in \mathbb{R}^{n\times r}$ such that $\mathbf{X}^T\mathbf{X} = \mathbf{I}$. And an abstract element on the Grassmann manifold $\mathrm{Gr}(r, n)$ is characterized by the equivalence class $[\mathbf{X}] = \{\mathbf{XO} : \mathbf{O} \in \mathcal{O}(r)\}$ at $\mathbf{X} \in \mathrm{St}(r, n)$.

Since the manifold $\mathcal{M}/\sim$ is an abstract space, the elements of its tangent space $T_{[x]}(\mathcal{M}/\sim)$ at $[x]$ also call for a matrix representation in the total space $\mathcal{M}$ that respects the equivalence relationship $\sim$. Equivalently, matrix representation of $T_{[x]}(\mathcal{M}/\sim)$ should be restricted to the directions in the tangent space $T_x\mathcal{M}$ on the total space $\mathcal{M}$ at x that do not induce a displacement along the equivalence class $[x]$. This is realized by decomposing $T_x\mathcal{M}$ into complementary subspaces, the *vertical* and *horizontal* subspaces such that $\mathcal{V}_x \oplus \mathcal{H}_x = T_x\mathcal{M}$. The vertical space $\mathcal{V}_x$ is the tangent space of the equivalence class $[x]$. On the other hand, the horizontal space $\mathcal{H}_x$, which is the complementary space of $\mathcal{V}_x$, provides a valid matrix representation of the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ (Absil et al., 2008, Section 3.5.8).

The Riemannian steepest-descent algorithm for

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

1. Search direction: compute the negative Riemannian gradient $\xi_x = -\mathrm{grad}_x f$ with respect to the Riemannian metric g_x (3.5). Equivalently, by solving the problem (3.8).
2. Retract with backtracking line search: the next iterate is computed using the retraction (3.9) such that $x_+ = R_x(s\xi_x)$, where the step-size s is obtained by a backtracking line search.
3. Repeat until convergence.

Table 3.2: The Riemannian steepest-descent algorithm.

An abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ at $[x]$ has a unique element in the horizontal space $\xi_x \in \mathcal{H}_x$ that is called its *horizontal lift*.

The other critical ingredient is the Riemannian metric. A metric g_x at $x \in \mathcal{M}$ in the total space defines a valid Riemannian metric $g_{[x]}$ on the quotient manifold $\mathcal{M}/\sim$ if

$$g_{[x]}(\xi_{[x]}, \eta_{[x]}) := g_x(\xi_x, \eta_x), \tag{3.5}$$

where $\xi_{[x]}$ and $\eta_{[x]}$ are any tangent vectors in $T_{[x]}(\mathcal{M}/\sim)$, and ξ_x, η_x are their horizontal lifts in $\mathcal{H}_x$. In other words, the metric g_x (3.5) in $\mathcal{M}$ induces a Riemannian metric $g_{[x]}$ on the quotient manifold $\mathcal{M}/\sim$, provided $g_x(\xi_x, \zeta_x) = g_y(\xi_y, \zeta_y)$ for all $x, y \in [x]$; where $\xi_x, \zeta_x \in \mathcal{H}_x$ and $\xi_y, \zeta_y \in \mathcal{H}_y$ are the horizontal lifts of $\xi_{[x]}, \zeta_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ along the same equivalence class $[x]$ (Absil et al., 2008, Section 3.6.2). In words, the metric g_x is *invariant* along the equivalence class $[x]$. Endowed with this Riemannian metric, the quotient manifold $\mathcal{M}/\sim$ is called a Riemannian quotient manifold of $\mathcal{M}$.

The choice of the metric (3.5), which is invariant along the equivalence class $[x]$, and of the horizontal space $\mathcal{H}_x$ as the orthogonal complement of $\mathcal{V}_x$, in the sense of the Riemannian metric (3.5), makes the space $\mathcal{M}/\sim$ a *Riemannian submersion* and allows for a convenient matrix representation of the gradient of a cost function. Figure 3.2 presents a schematic view of the search space. Consequently, the steepest-descent algorithm on the manifold $\mathcal{M}$ that respects the equivalence property $\sim$ on the space acquires the form shown in Table 3.2. Convergence of the steepest-descent algorithm in the neighborhood of the minimum is shown by Absil et al. (2008, Theorems 4.3.1 and 4.5.1). The main ingredients of Table 3.2 are the gradient direction and the *retraction* mapping.

Riemannian gradient

The horizontal lift of the Riemannian gradient $\mathrm{grad}_{[x]}$ of a cost function, say $f : \mathcal{M} \to \mathbb{R}$, on the quotient manifold $\mathcal{M}/\sim$ is uniquely represented by the matrix representation

$$\text{horizontal lift of } \mathrm{grad}_{[x]} f = \mathrm{grad}_x f, \tag{3.6}$$

where $\mathrm{grad}_x f$ is the gradient on the computational space $\mathcal{M}$. The equality in (3.6) is possible due to invariance of the cost function along the equivalence class $[x]$ (Absil et al., 2008, Section 3.6.2).

The gradient on the computational space $\mathrm{grad}_x f$ is computed from its definition: it is the unique element of $T_x\mathcal{M}$ that satisfies (Absil et al., 2008, Equation 3.31)

$$\mathrm{D}f(x)[\eta_x] = g_x(\mathrm{grad}_x f, \eta_x) \quad \text{for all } \eta_x \in T_x\mathcal{M}, \tag{3.7}$$

where g_x is the Riemannian metric (3.5) and $\mathrm{D}f(x)[\eta_x]$ is the standard Euclidean directional derivative of f in the direction η_x, i.e., $\mathrm{D}f(x)[\eta_x] = \lim_{t\to 0}(f(x + t\eta_x) - f(x))/t$ (Absil et al., 2008, Section 3.6). An equivalent way of computing $\mathrm{grad}_x f$ is by solving the *convex quadratic* problem

$$\mathrm{grad}_x f = \underset{\zeta_x \in T_x\mathcal{M}}{\arg\min}\, f(x) - \langle f_x(x), \zeta_x\rangle + \tfrac{1}{2} g_x(\zeta_x, \zeta_x), \tag{3.8}$$

where $f_x(x)$ is the derivative of the cost function f, $\langle\cdot,\cdot\rangle$ is the standard inner product, and g_x is the Riemannian metric (3.5) at $x \in \mathcal{M}$. It should be noted that $\langle f_x(x), \zeta_x\rangle = \mathrm{D}f(x)[\zeta_x]$ which is the standard Euclidean directional derivative of f in the direction $\zeta_x \in T_x\mathcal{M}$. The equivalence between solution to (3.7) and (3.8) is established by observing that the condition (3.7) is, in fact, equivalent to the optimality condition of the convex quadratic problem (3.8).

Retraction

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

$$x_+ = R_x(\xi_x), \tag{3.9}$$

where $\xi_x \in \mathcal{H}_x$ is a search direction and $x_+ \in \mathcal{M}$. The retraction R_x defines a valid retraction on the Riemannian quotient manifold $\mathcal{M}/\sim$ such that $R_{[x]}(\xi_{[x]}) := [R_x(\xi_x)]$, where ξ_x is the horizontal lift of an abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ in $\mathcal{H}_x$ and $[\cdot]$ is the equivalence class defined earlier in the section.

3.2.3 Connecting SQP to the Riemannian framework

The practical performance of the Riemannian steepest-descent algorithm in Table 3.2 greatly depends on the choice of the metric (3.5). The dominant trend in Riemannian optimization has been to infer the metric from the geometry of the search space. Symmetry properties of the search space suggest to choose invariant metrics, that is, metrics that are not affected by a symmetry transformation of variables. In many situations, invariance properties single out the choice of a unique metric (Absil et al., 2008; Edelman et al., 1998). However, the metrics that are motivated solely by the search space may not perform well in an optimization setup as they do not take into consideration the cost function (Manton, 2002).

To address the above issue, we connect the SQP approach in Table 3.1 to the Riemannian steepest-descent algorithm in Table 3.2 with a specific metric that is induced by the Lagrangian (3.2) on the horizontal space $\mathcal{H}_x$. The connection has a twofold objective. First, it provides a guidance in choosing metrics on a manifold. Second, it provides a framework to analyze the performance of SQP for constraints with equivalence relations.

Theorem 3.1. *Assume that both $\mathcal{M}$ and $\mathcal{M}/\sim$ have the structure of a differentiable manifold and that the function $f:\mathcal{M}\to\mathbb{R}$ is a smooth function with isolated minima on the quotient manifold $\mathcal{M}/\sim$. If $x^*\in\mathcal{M}$ is a local minimum of $f:\mathcal{M}\to\mathbb{R}$ on $\mathcal{M}$ that is endowed with an equivalence relationship $\sim$, then*

(i) $\langle \eta_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\eta_{x^*}]\rangle = 0$ for all $\eta_{x^*}\in\mathcal{V}_{x^*}$,
$\langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\xi_{x^*}]\rangle > 0$ for all $\xi_{x^*}\in\mathcal{H}_{x^*}$, *and*

(ii) The quantity $\langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\xi_{x^*}]\rangle$ at optimality is equal to the second-order term of the Taylor expansion f with any Riemannian metric structure imposed on $\mathcal{M}/\sim$,

where $\mathcal{V}_{x^*}$ is the vertical space, $\mathcal{H}_{x^*}$ is horizontal space (the subspace of $T_{x^*}\mathcal{M}$ that is complementary to $\mathcal{V}_{x^*}$), $\langle\cdot,\cdot\rangle$ is the standard inner product, and $\mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\xi_{x^*}]$ is the second-order partial derivative of $\mathcal{L}(x,\lambda_x)$ with respect to x at $x^*\in\mathcal{M}$ applied in the direction $\xi_{x^*}\in\mathcal{H}_{x^*}$, keeping λ_{x^*} fixed to its least-squares estimate (3.3).

Proof. The Lagrangian $\mathcal{L}(x,\lambda_x)$, because both the cost and constraint terms remain invariant under the equivalence relationship $\sim$, is *constant* along the equivalence class $[x] := \{y\in\mathcal{M} : y\sim x\}$. Consequently from the Taylor expansion of $\mathcal{L}(x,\lambda_x)$ along the linearization of the equivalence class $[x]$, that is the vertical space $\mathcal{V}_x$, all the Taylor terms equate to zero. Hence, the first equality in (i) follows immediately.

The second inequality in (ii), $\langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\xi_{x^*}]\rangle > 0$ for all $\xi_{x^*}\in\mathcal{H}_{x^*}$, follows from the second-order optimality conditions of the problem on the quotient manifold. Consider the first-order optimality condition for the problem that states $\mathcal{L}_x(x^*,\lambda_x^*) = 0$ at the minimum, where $\mathcal{L}_x(x,\lambda_x)$ is the first-order derivative of $\mathcal{L}(x,\lambda_x) = f(x) - \langle\lambda_x, h(x)\rangle$ with respect to x (Edelman et al., 1998, Section 4.9). Additionally from calculus, the second-order *necessary* condition for $x^*\in\mathcal{M}$ to be a local minimum of a smooth function $f:\mathcal{M}\to\mathbb{R}$ on the constraint set $\mathcal{M}$ is

$$\frac{d^2}{dt^2} f(x(t))\Big|_{t=0} \geq 0,$$

where $t\in\mathbb{R}$ and $x(t)\in\mathcal{M}$ is a curve originating from x^* such that $x(0)=x^*$. By twice-differentiating $\mathcal{L}(x(t),\lambda_{x^*}) = f(x(t)) - \langle\lambda_{x^*}, h(x(t))\rangle$ and $h(x(t))=0$ at $t=0$, we have $\langle \zeta_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\zeta_{x^*}]\rangle \geq 0$ for all tangent vectors $\zeta_{x^*}\in T_{x^*}\mathcal{M}$. With the decomposition $T_{x^*}\mathcal{M} = \mathcal{V}_{x^*}\oplus\mathcal{H}_{x^*}$ (the decomposition is complementary) and the equality $\langle \eta_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\eta_{x^*}]\rangle = 0$ for all $\eta_{x^*}\in\mathcal{V}_{x^*}$, the necessary condition for a local minimum on the quotient manifold boils down to $\langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*,\lambda_{x^*})[\xi_{x^*}]\rangle \geq 0$ for all $\xi_{x^*}\in\mathcal{H}_{x^*}$.

However, as x^* (by definition) represents an isolated (non-degenerate) local minimum on the quotient manifold $\mathcal{M}/\sim$, i.e., the function $f(x(t)) > f(x^*)$ in the neighborhood of $x(0) = x^*$ on the quotient manifold, implying that $\langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*, \lambda_{x^*})[\xi_{x^*}]\rangle > 0$ for all $\xi_{x^*} \in \mathcal{H}_{x^*}$.

To prove the statement (ii) of the theorem, consider a Riemannian metric $\bar{g}$ on the manifold $\mathcal{M}$ that *submerses* $\mathcal{M}/\sim$ in $\mathcal{M}$ (Absil et al., 2008, Chapter 3). The theory of Riemannian submersion states that the horizontal space $\mathcal{H}_x$ is the *orthogonal* to the vertical space $\mathcal{V}_x$ with the metric $\bar{g}_x$ and allows us to compute the Riemannian gradient and Hessian of f on $\mathcal{M}/\sim$ using the *orthogonal projection* of their counterparts in the total space $\mathcal{M}$. From the computation of the Riemannian gradient $\mathrm{grad}_x f$ and the first-derivative of the Lagrangian we have (Absil et al., 2008, Equation 3.31)

$$\mathrm{D}f(x)[\xi_x] = \langle \mathcal{L}_x(x, \lambda_x), \xi_x\rangle = \bar{g}_x(\mathrm{grad}_x f, \xi_x)$$

for all $\xi_x \in T_x\mathcal{M}$. Taking the directional derivative of the above equation x^* along $\xi_{x^*} \in \mathcal{H}_{x^*}$ with the additional information that $\mathrm{grad}_{x^*} f = 0$ (x^* is a local minimum),

$$\langle \xi_{x^*}, \mathrm{D}^2 f(x^*)[\xi_{x^*}]\rangle = \langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*, \lambda_{x^*})[\xi_{x^*}]\rangle = \bar{g}_{x^*}(\xi_{x^*}, \mathrm{Dgrad}_{x^*} f[\xi_{x^*}]). \tag{3.10}$$

Defining $\Pi_x : \mathbb{R}^n \to \mathcal{H}_x$ be the orthogonal projection operator in the metric $\bar{g}_x$, and $\mathrm{Hess}_x f[\xi_x]$ be the Riemannian Hessian in the total space $\mathcal{M}$ applied along the direction $\xi_x \in \mathcal{H}_x$, the second-order term of the Taylor expansion of f along $\xi_{x^*} \in \mathcal{H}_{x^*}$ is

$$\begin{aligned}
\bar{g}_{x^*}(\xi_{x^*}, \underbrace{\Pi_{x^*}(\mathrm{Hess}_{x^*} f[\xi_{x^*}])}_{\text{Hessian on } \mathcal{M}/\sim}) &= \bar{g}_{x^*}(\xi_{x^*}, \underbrace{\mathrm{Hess}_{x^*} f[\xi_{x^*}]}_{\text{Hessian on } \mathcal{M}}) \quad \text{for } \Pi_{x^*}(\xi_{x^*}) = \xi_{x^*} \\
&= \bar{g}_{x^*}(\xi_{x^*}, \mathrm{Dgrad}_{x^*} f[\xi_{x^*}]) \quad \text{for } \mathrm{grad}_{x^*} f = 0 \\
&= \langle \xi_{x^*}, \mathrm{D}^2\mathcal{L}(x^*, \lambda_{x^*})[\xi_{x^*}]\rangle \quad \text{from (3.10).}
\end{aligned}$$

This proves the statement (ii).

□

Theorem 3.1 states that even though the underlying symmetries make the Hessian of the Lagrangian *singular* in the tangent space $T_x\mathcal{M}$ of the total space $\mathcal{M}$, the Hessian of the Lagrangian is *non-singular* on the horizontal space $\mathcal{H}_x$ and its singularity is *only* along the vertical space $\mathcal{V}_x$. The other important observation is that the quantity $\langle \xi_x, \mathrm{D}^2\mathcal{L}(x, \lambda_x)[\xi_x]\rangle$ captures the full second-order information at the local minimum along the horizontal space, where the horizontal space is any subspace of $T_x\mathcal{M}$ that is complementary to the vertical space $\mathcal{V}_x$.

Constructing a metric from SQP

As an immediate consequence of Theorem 3.1, we observe that in the neighborhood of the minimum, a valid selection of the search direction is given by solving

$$\operatorname*{arg\,min}_{\zeta_x \in \mathcal{H}_x} \quad f(x) + \langle f_x(x), \zeta_x\rangle + \tfrac{1}{2}\langle \zeta_x, \mathrm{D}^2\mathcal{L}(x, \lambda_x)[\zeta_x]\rangle. \tag{3.11}$$

Followed by a retraction operation (Section 3.2.2), (3.11) defines locally a steepest-descent algorithm on the quotient manifold $\mathcal{M}/\sim$, that is, an algorithm that iterates on the classes of equivalences. This holds for *any* characterization of the horizontal space $\mathcal{H}_x$. Here $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product and $f_x(x)$ is the first-order derivative of the function f. The scheme with (3.11) has the interpretation of a steepest descent algorithm on the quotient manifold $\mathcal{M}/\sim$. This result is of significance because the computational problem (3.11) is considerably simpler than the computational machinery needed for a steepest-descent algorithm on a general quotient manifold. However, it is no obvious how to extend (3.11) away from the local minimum, nor how to use the same metric in more general optimization algorithms.

A remedy to those limitations is to use the insight from (3.11) to build a Riemannian metric $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ *induced* from the Hessian of the Lagrangian $\mathcal{L}(x, \lambda_x)$ that apart from satisfying standard metric properties (Absil et al., 2008, Section 3.6), should also

- satisfy the inequality $g_x(\xi_x, \xi_x) > 0$ for all $\xi_x \in T_x\mathcal{M}$ for all $x \in \mathcal{M}$,
- exploit the full Hessian of the Lagrangian information *only* at the minimum, that is, it converges to
$$g_x(\xi_x, \eta_x) \quad = \langle \xi_x, \mathrm{D}^2\mathcal{L}(x, \lambda_x)[\eta_x]\rangle \qquad \text{for all } \xi_x, \eta_x \in T_x\mathcal{M}, \tag{3.12}$$
where λ_x is the least-squares estimate (3.3), and
- is invariant along the equivalence class $[x] = \{y \in \mathcal{M} : y \sim x\}$ at $x \in \mathcal{M}$.

To construct metrics with the above properties the Lagrangian structure $\mathcal{L}(x, \lambda_x) = f(x) - \langle \lambda_x, h(x)\rangle$ (3.2), that has terms arising from cost and constraints, plays a critical role. Construction of a family of Riemannian metrics by exploiting this structure for specific scenarios is discussed later in Section 3.2.4.

In addition to the properties listed in (3.11), if the horizontal space $\mathcal{H}_x$ is also *chosen* as the orthogonal subspace to the vertical space $\mathcal{V}_x$ with respect to the constructed Riemannian metric g (3.12), then the manifold $\mathcal{M}/\sim$ has the structure of a Riemannian submersion (Absil et al., 2008; Edelman et al., 1998). Consequently, computation of the search direction has a simpler characterization, that is, it is equivalent to solving the problem
$$\operatorname*{arg\,min}_{\zeta_x \in T_x\mathcal{M}} \quad f(x) + \langle f_x(x), \zeta_x\rangle + \tfrac{1}{2} g_x(\zeta_x, \zeta_x), \tag{3.13}$$
where $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product and $f_x(x)$ is the first-order derivative of the function f, and $g(\cdot,\cdot)$ is the Riemannian metric that is induced from Hessian of the Lagrangian (3.12). It should be stated that even though the minimization (3.13) is on the tangent space $T_x\mathcal{M}$, the solution ζ_x^* to (3.13), by construction, also belongs to the chosen horizontal space $\mathcal{H}_x$ (orthogonal to $\mathcal{V}_x$).

The resulting algorithms arising both from (3.11) and (3.13), and followed with a retraction operation (Section 3.2.2), by construction, define a steepest-descent algorithm on the quotient manifold. They are summarized in Figure 3.3 for completeness. The performance characterization of these algorithms on $\mathcal{M}$, equipped with the equivalence relationship $\sim$, follows from the (local) analysis of SQP by Absil et al. (2008, Section 6.3.1), Absil et al. (2009, Proposition 4.1), Nocedal and Wright (2006, Theorem 18.4). Theorem 3.1 emphasizes the fact that SQP provides a systematic guidance to identify Riemannian metrics that locally in the neighborhood of the minimum exploit second-order information of the function.

Search direction computation	Comments
$\arg\min_{\zeta_x\in\mathcal{H}_x} f(x)+\langle f_x(x),\zeta_x\rangle$ $+\frac{1}{2}\langle\zeta_x,\mathrm{D}^2\mathcal{L}(x,\lambda_x)[\zeta_x]\rangle$	• Any horizontal space characterization • Valid locally in the neighborhood of a local minimum
$\arg\min_{\zeta_x\in T_x\mathcal{M}} f(x)+\langle f_x(x),\zeta_x\rangle$ $+\frac{1}{2}g_x(\zeta_x,\zeta_x)$	• Horizontal space is orthogonal to the vertical space • Riemannian submersion • Possibly valid globally

Figure 3.3: Two ways computing a search direction on the quotient manifold.

3.2.4 Riemannian optimization and local convexity

As mentioned in Section 3.2.3, the Riemannian metrics for the Riemannian steepest-descent algorithm in Table 3.2 are identified from the second-order partial derivative of $\mathcal{L}(x,\lambda_x)$ with respect to x keeping λ_x fixed to its least-squares estimate (3.3), where $h_x(x)$ and $f_x(x)$ are first-order derivatives of h and f, respectively. Because the Lagrangian $\mathcal{L}(x,\lambda_x)$ consists of contributions from the cost function as well as the constraints, the metric g (3.12) admits the simple decomposition

$$\begin{aligned} g_x(\xi_x,\eta_x) &= \langle\xi_x,\mathrm{D}^2\mathcal{L}(x,\lambda_x)[\eta_x]\rangle \\ &= \underbrace{\langle\xi_x,\mathrm{D}^2 f(x)[\eta_x]\rangle}_{\text{cost related}}+\underbrace{\langle\xi_x,\mathrm{D}^2 c(x,\lambda_x)[\eta_x]\rangle}_{\text{constraint related}}, \end{aligned} \tag{3.14}$$

where ξ_x,η_x are any tangent vectors in $T_x\mathcal{M}$, $c(x,\lambda_x)=-\langle\lambda_x,h(x)\rangle$, and $\mathrm{D}^2c(x,\lambda_x)[\eta_x]$ is the second-order partial derivative of $c(x,\lambda_x)$ with respect to x keeping λ_x fixed, applied in the direction η_x.

The decomposition (3.14) between cost and constraint terms can be weighted in a way that turns (3.14) into a proper metric, i.e., $g_x(\xi_x,\xi_x)>0$ for all $\xi_x\in T_x\mathcal{M}$. The discussion is problem dependent but illustrated in Figure 3.4. Additionally, updating the weighing parameter ω, that weighs different terms of (3.14), is also discussed in the context of *globalizing* the metrics, i.e., extending the proposed Riemannian metrics away from the neighborhood. We further discuss two scenarios that suggest how to exploit the available structure to construct novel Riemannian metrics. The problem structure can be exploited in more general situations along the same lines.

Case I: minimizing a strictly convex function

Consider the case when f is a *strictly* convex function. In this case the second-order derivative $f_{xx}(x)\succ 0$ (due to strict convexity assumption) is a good metric candidate. In addition, locally in the neighborhood of the minimum, the family of Riemannian metrics is identified as

$$g_x(\xi_x,\eta_x)=\underbrace{\langle\xi_x,\mathrm{D}^2 f(x)[\eta_x]\rangle}_{f_{xx}\succ 0\text{ and dominating}}+\omega\langle\xi_x,\mathrm{D}^2c(x,\lambda_x)[\eta_x]\rangle, \tag{3.15}$$

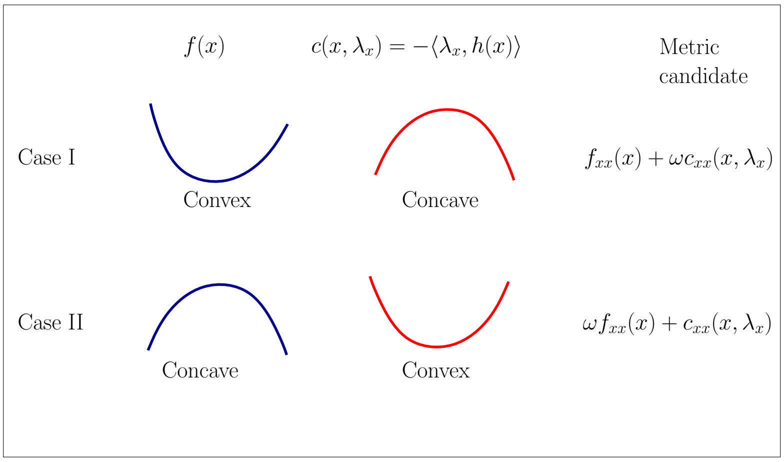

$f(x)$
$c(x,\lambda_x) = -\langle \lambda_x, h(x)\rangle$
Metric candidate
Case I
Convex
Concave
$f_{xx}(x) + \omega c_{xx}(x,\lambda_x)$
Case II
Concave
Convex
$\omega f_{xx}(x) + c_{xx}(x,\lambda_x)$

FIGURE 3.4: Choosing metrics for the Riemannian steepest-descent algorithm in Table 3.2. Shown are two extreme situations in which the Lagrangian (3.2) provides a clear metric candidate locally in the neighborhood of the minimum. $f_{xx}(x)$ is the second-order derivative of $f(x)$ and $c_{xx}(x,\lambda_x)$ is the second-order partial derivative of $c(x,\lambda_x)$ with respect to x keeping λ_x fixed. Because of the local convexity of the Lagrangian (on the tangent space) at the minimum, convex and concave structures of the function f lead to well-defined family of metrics parameterized by the weight $\omega \in [0,1)$. It locally captures the second-order information of the problem. To extend the metrics away from the minimum, the weight ω is updated at every iteration with a barrier function that tends to 1 as iterations tend to infinity.

where the weight $\omega \in [0,1)$, ξ_x, η_x are any tangent vectors in $T_x\mathcal{M}$, $c(x) = -\langle \lambda_x, h(x)\rangle$, and $\mathrm{D}^2 c(x,\lambda_x)[\eta_x]$ is the second-order partial derivative of $c(x,\lambda_x)$ with respect to x keeping λ_x fixed, applied in the direction η_x. It should be noted that for $\omega \in [0,1)$, $g_x(\xi_x,\xi_x) > 0$ locally in the neighborhood of the minimum for all $\xi_x \in T_x\mathcal{M}$.

Case II: maximizing a strictly convex function

Consider the problem of maximizing a convex cost function, that is equivalent to minimizing a concave cost function, on a manifold. In this case, $f_{xx}(x,\lambda_x) \prec 0$, and locally in the neighborhood of the minimum, the second-order information of $c(x) = -\langle \lambda_x, h(x)\rangle$ is the proper source of convexity. This fact follows from the second-order optimality condition of the optimization problem (Nocedal and Wright, 2006, Chapter 18). Here the problem structure suggests the family of Riemannian metrics

$$g_x(\xi_x,\eta_x) = \omega\langle \xi_x, \mathrm{D}^2 f(x)[\eta_x]\rangle + \underbrace{\langle \xi_x, \mathrm{D}^2 c(x,\lambda_x)[\eta_x]\rangle}_{c_{xx} \text{ is locally positive definite}} , \qquad (3.16)$$

where the weight $\omega \in [0,1)$, ξ_x, η_x are any tangent vectors in $T_x\mathcal{M}$, $c(x) = -\langle \lambda_x, h(x)\rangle$, $c_{xx}(x,\lambda_x)$ is the second-order partial derivative of $c(x,\lambda_x)$ with respect to x keeping λ_x fixed. Once again for $\omega \in [0,1)$, $g_x(\xi_x,\xi_x) > 0$ locally in the neighborhood of the minimum for all $\xi_x \in T_x\mathcal{M}$.

Globalizing the local metrics

The weight $\omega \in [0,1)$ in the metrics (3.15) and (3.16), apart from providing a family of Riemannian metrics, also plays a critical role in the numerical performance of the Riemannian steepest-descent algorithm in Table 3.2. With $\omega = 0$, the Riemannian metric captures only part of the second-order information and therefore, locally in the neighborhood of the minimum, the Riemannian steepest-descent algorithm may converge poorly, e.g., linearly. On the other hand with $\omega = 1$, the Riemannian metric captures the full second-order information and the Riemannian steepest-descent algorithm is expected to show better convergence. A numerical technique to interpolate between these two extreme scenarios is to vary $\omega = [0, 1)$ at every iteration with a increasing barrier function that tends to 1 as the number of iterations increase. A simple updating technique is $\omega(k) = 1 - 2^{k-1}$, where k is the iteration number. A strategy to safeguard against a non-descent search direction (by solving the quadratic programming problem (3.4)) is to ignore the updated ω that resulted in a non-descent direction (checking this is straightforward) and restart the procedure of updating ω again.

A different technique is to modify ω as and when required. For example, defining $\delta = 1 - \omega$, we have the strategy where at the k^{th} iteration

$$\delta_k = \begin{cases} \frac{1}{2}\delta_{k-1}, & \text{when a descent direction is obtained} \\ 4\delta_{k-1}, & \text{when a non} - \text{descent direction is obtained} \end{cases} \tag{3.17}$$

with $\delta_0 = 1$. Care is taken to ensure that $\omega \in [0, 1)$ for all iterations.

Safeguards similar to the trust-regions, i.e., by constraining the norm of the search direction, can also be implemented to ensure that the search direction computed with the Riemannian metric remains a locally descent direction (Nocedal and Wright, 2006, Section 18.5).

3.3 Quadratic optimization with orthogonality constraints: revisiting the generalized eigenvalue problem

Constrained quadratic optimization problems arise naturally in a number of applications, especially while solving linear systems of matrix equations (Absil et al., 2008, Section 2.2). Also popular are the *orthogonality* constraints in large-scale problems that are imposed to identify relevant smaller dimensional subspaces (Edelman et al., 1998). Specific optimization problems include the generalized eigenvalue problem (Absil and Van Dooren, 2010; Absil et al., 2002; Edelman et al., 1998), the generalized orthogonal Procrustes problem (Eldén and Park, 1999), and the joint diagonalization problem in signal processing (Theis et al., 2009), to name a few.

For the sake of illustration, we specifically focus on the well-studied generalized eigenvalue problem that computes the smallest eigenvalues and eigenvectors of the matrix $\mathbf{B}^{-1}\mathbf{A}$, where $\mathbf{A}$ is a symmetric matrix of size $n \times n$ and $\mathbf{B}$ is a symmetric positive definite matrix of size $n \times n$ (Edelman et al., 1998, Section 4.5; Golub and Van Loan, 1996, Chapter 8). This is realized by solving the optimization problem below iteratively, an extensively researched question in literature (Golub and Van Loan, 1996, Chapter 8; Absil et al., 2004b). In this section we exploit the quadratic nature of the cost function and the constraints to show that the family of Riemannian metrics has a simple characterization. It is also shown that the algorithms that result from the proposed metrics connect to a number of established algorithms. Each of which is interpreted as a steepest-descent algorithm with a specific Riemannian metric.

The minimal r-eigenspace of $\mathbf{B}^{-1}\mathbf{A}$ is computed iteratively by solving the constrained quadratic optimization problem

$$\begin{array}{ll} \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times r}} & \frac{1}{2}\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X}) \\ \text{subject to} & \mathbf{X}^T\mathbf{B}\mathbf{X} = \mathbf{I}, \end{array} \tag{3.18}$$

where the constraint set of $n \times r$ matrices that satisfy $\mathbf{X}^T\mathbf{B}\mathbf{X} = \mathbf{I}$ is known as the *generalized Stiefel manifold* $\mathrm{St}_{\mathbf{B}}(r, n)$. The constraint enforces orthogonality among vectors in coordinates spanned by $\mathbf{B}^{1/2}$. Specifically when $\mathbf{B} = \mathbf{I}$, the generalized Stiefel manifold is the popular Stiefel manifold $\mathrm{St}(r, n) := \{\mathbf{X} \in \mathbb{R}^{n\times r} : \mathbf{X}^T\mathbf{X} = \mathbf{I}\}$ (Edelman et al., 1998). The symmetry in the cost function $\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})/2$ comes from its invariance under the transformation $\mathbf{X} \mapsto \mathbf{X}\mathbf{O}$ for all $\mathbf{O} \in \mathcal{O}(r)$. $\mathcal{O}(r)$ is the set of $r \times r$ orthogonal matrices.

The symmetry in the cost function translates the property that an orthogonal set of vectors characterize a subspace *modulo* rotations in the subspace, i.e., the eigenspace is *invariant* to rotations of vectors in the eigenspace. As a consequence, the problem (3.18) is an optimization problem on the abstract quotient space $\mathrm{St}_{\mathbf{B}}(r, n)/\mathcal{O}(r)$, also known as the generalized Grassmann manifold. For the case $\mathbf{B} = \mathbf{I}$, this again boils down to the well known Grassmann manifold $\mathrm{Gr}(r, n)$, the set of r-dimensional subspaces in $\mathbb{R}^n$ (Edelman et al., 1998). The optimization problem (3.18) is, therefore, reformulated on the generalized Grassmann quotient manifold, i.e.,

$$\begin{array}{ll} \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times r}} & \frac{1}{2}\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X}) \\ \text{subject to} & [\mathbf{X}] \in \mathrm{St}_{\mathbf{B}}(r, n)/\mathcal{O}(r), \end{array} \tag{3.19}$$

where the optimization is on the set of equivalence classes $[\mathbf{X}] = \{\mathbf{X}\mathbf{O} : \mathbf{O} \in \mathcal{O}(r)\}$ at $\mathbf{X} \in \mathrm{St}_{\mathbf{B}}(r, n)$.

The *conventional* metric of choice in the Riemannian framework is

$$g_x(\eta_x, \xi_x) = \mathrm{Trace}(\eta_x^T\xi_x), \tag{3.20}$$

where $x = \mathbf{X} \in \mathrm{St}_{\mathbf{B}}(r, n)$ and ξ_x, η_x are tangent vectors in the tangent space of the constraints (the matrix characterization the tangent space is shown in Table 3.3). It is the unique metric that is invariant to the group action of $\mathcal{O}(r)$. Because of its simplicity and its geometric consideration, this metric is advocated by Absil et al. (2008); Edelman et al. (1998).

In contrast, the developments in Section 3.2.3 suggest a family of Riemannian metrics that take the complete problem structure into account by computing the Lagrangian and its derivatives. We have the matrix representations in (3.21). It should be noted that we introduce an additional factor of 1/2 in the constraint penalization term of the Lagrangian to make resulting expressions simpler.

$$\begin{array}{rrl} & \mathcal{L}(x, \lambda_x) = & \mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})/2 - \langle\lambda_x, \mathbf{X}^T\mathbf{B}\mathbf{X} - \mathbf{I}\rangle/2 \\ \Rightarrow & \mathcal{L}_x(x, \lambda_x) = & \mathbf{A}\mathbf{X} - \mathbf{B}\mathbf{X}\lambda_x \\ \Rightarrow & \mathrm{D}^2\mathcal{L}(x, \lambda_x)[\xi_x] = & \mathbf{A}\xi_x - \mathbf{B}\xi_x\lambda_x, \end{array} \tag{3.21}$$

where x has the matrix representation $\mathbf{X} \in \mathrm{St}_{\mathbf{B}}(r, n)$, $\langle\cdot, \cdot\rangle$ is the standard inner product, and the least-squares Lagrange multiplier is $\lambda_x = \mathrm{Sym}((\mathbf{X}^T\mathbf{B}\mathbf{B}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{B}\mathbf{A}\mathbf{X}))$ from (3.3) with the additional symmetry condition from the constraint, where $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e.,

$\mathrm{Sym}(\mathbf{D}) = (\mathbf{D}+\mathbf{D}^T)/2$. Here $\mathcal{L}_x(x, \lambda_x)$ is the first-order partial derivative of $\mathcal{L}(x, \lambda_x)$ and $\mathrm{D}^2\mathcal{L}(x, \lambda_x)[\xi_x]$ is the second-order partial derivative of $\mathcal{L}(x, \lambda_x)$ applied in the direction ξ_x, both computed while keeping λ_x fixed.

It should be noted that $\lambda_x = \mathrm{Sym}((\mathbf{X}^T\mathbf{BBX})^{-1}(\mathbf{X}^T\mathbf{BAX}))$ is the solution to the problem $\arg\min_{\lambda\in\mathbb{R}^{r\times r}} \|\mathbf{AX} - \mathbf{BX}\lambda\|_Q^2$ such that λ is symmetric, where $\|\mathbf{AX} - \mathbf{BX}\lambda\|_Q^2 = \mathrm{Trace}((\mathbf{AX} - \mathbf{BX}\lambda)^T\mathbf{Q}(\mathbf{AX} - \mathbf{BX}\lambda))$ and $\mathbf{Q} = \mathbf{BX}(\mathbf{X}^T\mathbf{BBX})^{-2}\mathbf{X}^T\mathbf{B}$. A different estimate of λ_x is obtained by solving the problem $\arg\min_{\lambda\in\mathbb{R}^{r\times r}} \|\mathbf{AX} - \mathbf{BX}\lambda\|_F^2$ such that λ is symmetric.

It is readily checked that the Lagrangian $\mathcal{L}(x, \lambda_x)$ in (3.21) remains unchanged under the action $\mathbf{X} \mapsto \mathbf{XO}$ for any $\mathbf{O} \in \mathcal{O}(r)$. Finally, based on the matrix characterizations (3.21), we have the following proposition for constructing a family of Riemannian metrics for the quadratic optimization problem (3.19).

Proposition 3.2. *The family of Riemannian metrics, locally in the neighborhood of the minimum, for the quadratic optimization problem (3.19) has the form*

$$g_x(\xi_x, \eta_x) = \underbrace{\langle \xi_x, \mathbf{A}\eta_x\rangle}_{\text{cost related}} - \underbrace{\langle \xi_x, \mathbf{B}\eta_x\lambda_x\rangle}_{\text{constraints related}}, \tag{3.22}$$

where ξ_x, η_x are any tangent vectors in tangent space of the constraints at $x = \mathbf{X}$ such that $\mathbf{X}^T\mathbf{BX} = \mathbf{I}$ and $\lambda_x = \mathrm{Sym}((\mathbf{X}^T\mathbf{BBX})^{-1}(\mathbf{X}^T\mathbf{BAX}))$, where $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\mathbf{D}) = (\mathbf{D} + \mathbf{D}^T)/2$.

Proof. This follows directly from the second-order partial derivative of the Lagrangian in (3.21) with respect to x. □

Matrix characterizations of various optimization related ingredients are summarized in Table 3.3. The retraction operator is the standard generalization of the retraction operator on the Stiefel manifold $\mathrm{St}(r, n)$ (Absil et al., 2008, Example 4.1.3).

3.3.1 Metric tuning and shift policies

Due to the quadratic nature of both cost and constraints, the metric (3.22) has the appealing feature of being parameterized by the Lagrangian parameter λ_x. This object is low-dimensional when $r \ll n$. It provides an interesting interpretation of various "shift" policies developed in numerical linear algebra for eigenspace computations (Golub and Van Loan, 1996, Chapter 8). We further specialize the selection of (3.22) when $\mathbf{A} \succ 0$ and when $\mathbf{A} \not\succ 0$. In both these cases, we propose metrics with that connect to a number of classical algorithms for the generalized eigenvalue problem (Absil and Van Dooren, 2010; Absil et al., 2002, 2004b).

When $\mathbf{A} \succ 0$

This instance falls under Case I in Figure 3.4 and therefore, the family of proposed Riemannian metrics has the structure

$$g_x(\xi_x, \zeta_x) = \mathrm{Trace}(\xi_x^T\mathbf{A}\zeta_x) - \omega\mathrm{Trace}(\xi_x^T\mathbf{B}\zeta_x\lambda_x), \tag{3.23}$$

	$\min_{\mathbf{X}\in\mathbb{R}^{n\times r}}$ $\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})/2$ subject to $\mathbf{X}^T\mathbf{B}\mathbf{X}=\mathbf{I}$
Matrix representation of an element in $\mathcal{M}$	$x=\mathbf{X}$
Computational space $\mathcal{M}$	$\mathrm{St}_{\mathbf{B}}(r,n)=\{\mathbf{X}\in\mathbb{R}^{n\times r}:\mathbf{X}^T\mathbf{B}\mathbf{X}=\mathbf{I}\}$
Group action	$\mathbf{X}\mathbf{O},\quad\forall\mathbf{O}\in\mathcal{O}(r)$ such that $\mathbf{O}^T\mathbf{O}=\mathbf{O}\mathbf{O}^T=\mathbf{I}$
Quotient space	$\mathrm{St}_{\mathbf{B}}(r,n)/\mathcal{O}(r)$
Tangent vectors in $T_x\mathcal{M}$	$\{\xi_x\in\mathbb{R}^{n\times r}:\xi_x^T\mathbf{B}\mathbf{X}+\mathbf{X}^T\mathbf{B}\xi_x=0\}$
Metric $g_x(\xi_x,\zeta_x)$ for any $\xi_x,\zeta_x\in T_x\mathcal{M}$	$g_x(\xi_x,\zeta_x)=\mathrm{Trace}(\zeta_x^T\mathbf{A}\xi_x)$ $-\mathrm{Trace}(\zeta_x^T\mathbf{B}\xi_x\lambda_x)$
	or the metrics proposed in Section 3.3.1, where $\lambda_x=\mathrm{Sym}((\mathbf{X}^T\mathbf{B}\mathbf{B}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{B}\mathbf{A}\mathbf{X}))$
Cost function	$f(x)=\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X})/2$
First $-$ order derivative of $f(x)$	$f_x(x)=\mathbf{A}\mathbf{X}$
Search direction	$\operatorname*{arg\,min}_{\zeta_x\in T_x\mathcal{M}}\quad f(x)+\langle f_x(x),\zeta_x\rangle+\frac{1}{2}g_x(\zeta_x,\zeta_x)$
Retraction $R_x(\xi_x)$ that maps a search direction ξ_x onto $\mathcal{M}$	$\mathbf{U}(\mathbf{U}^T\mathbf{B}\mathbf{U})^{-1/2}$, where $\mathbf{U}$ is the $\mathrm{Q}-$factor of the QR decomposition of $\mathbf{X}+\xi_x$

TABLE 3.3: Optimization-related ingredients for computing the extreme eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$. Three choices of metrics with their shifts connect to and generalize the popular power iteration, inverse iteration, and Rayleigh quotient iteration algorithms. The numeric complexity per iteration depends on solving the quadratic programming problem. In many instances exploiting sparsity in matrices $\mathbf{A}$ and $\mathbf{B}$ leads to numerically efficient schemes. Here $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\mathbf{D})=(\mathbf{D}+\mathbf{D}^T)/2$

where ξ_x and ζ_x are any tangent vectors in the tangent space of constraints, the least-squares Lagrange multiplier $\lambda_x=\mathrm{Sym}((\mathbf{X}^T\mathbf{B}\mathbf{B}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{B}\mathbf{A}\mathbf{X}))$, and $\omega=[0,1)$.

The metric (3.23) provides two insightful connections to the literature. First, the proposed metric (3.23) with $\omega=0$ generalizes the well-known inverse iteration algorithm for computing the smallest eigenvalues of a symmetric matrix (Golub and Van Loan, 1996, Section 8.2.2). For the case when $\mathbf{B}=\mathbf{I}$, the *negative* Riemannian gradient with the metric with (3.23) $\omega=0$ is computed as in Table 3.3 as

$$\left.\begin{array}{ll}\operatorname*{arg\,min}_{\substack{\zeta_x\in\mathbb{R}^{n\times r}\\ \zeta_x^T\mathbf{X}+\mathbf{X}^T\zeta_x=0}} & \langle\mathbf{A}\mathbf{X},\zeta_x\rangle+\frac{1}{2}\mathrm{Trace}(\zeta_x^T\mathbf{A}\zeta_x)\end{array}\right\}=\mathbf{A}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})^{-1}-\mathbf{X}.$$

The Riemannian steepest-descent update with *unit step-size*, thus, is $x_+=R_x(\mathbf{A}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})^{-1}$

$-\mathbf{X}) = \mathrm{qf}(\mathbf{A}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})^{-1}) = \mathrm{qf}(\mathbf{A}^{-1}\mathbf{X})$, where $R_x(\cdot)$ is the retraction operator defined later in Table 3.3, and $\mathrm{qf}(\mathbf{A}^{-1}\mathbf{X})$ is the Q-factor of the QR decomposition of $\mathbf{A}^{-1}\mathbf{X}$. This is precisely the classical inverse iteration update (Golub and Van Loan, 1996, Section 8.2.2). This shows that the inverse iteration has the interpretation of a Riemannian steepest-descent algorithm with the metric (3.27) for $\omega = 0$.

A second insight is obtained for the case when ω is updated with iterations, the Riemannian steepest descent algorithm with the metric (3.23) generalizes the popular Rayleigh quotient iteration algorithm (Golub and Van Loan, 1996, Section 8.2.3; Absil et al., 2002; Absil et al., 2004b). Consider again the case when $\mathbf{B} = \mathbf{I}$. At each iteration of the Riemannian steepest-descent algorithm with the metric (3.23), we are required to solve the system of linear equations (by looking at the optimality conditions of the quadratic program for computing the search direction) for $\zeta_x \in \mathbb{R}^{n\times r}$ and $\mu_x \in \mathbb{R}^{r\times r}$ of the form

$$\begin{aligned} &\mathbf{A}\zeta_x - \omega\zeta_x\lambda_x = \mathbf{X}\mu_x - \mathbf{A}\mathbf{X} \\ &\mathbf{X}^T\zeta_x + \zeta_x^T\mathbf{X} = 0, \end{aligned} \tag{3.24}$$

where the weight $\omega \in [0, 1)$, ζ_x is the search direction, and μ_x is the matrix scaling that guarantees that the search direction ζ_x belongs to the tangent space of the constraints. It should be noted that the linear system of equations (3.24) can be solved efficiently by exploiting additional sparsity structure in $\mathbf{A}$ (Absil et al., 2002). The Riemannian steepest-descent update x_+ with unit step-size is

$$\begin{aligned} &\left.\begin{array}{lcl} \mathbf{A}\zeta_x^* - \omega\zeta_x^*\lambda_x & = & \mathbf{X}\mu_x^* - \mathbf{A}\mathbf{X} \\ x_+ & = & R_x(\zeta_x^*) = \mathrm{qf}(\mathbf{X} + \zeta_x^*) \end{array}\right\} \\ \Rightarrow &\left\{\begin{array}{lcl} \mathbf{A}\mathbf{Z} - \omega\mathbf{Z}\lambda_x & = & \mathbf{X}(\mu_x^* - \omega\lambda_x) \\ x_+ & = & \mathrm{qf}(\mathbf{Z}), \end{array}\right. \end{aligned} \tag{3.25}$$

where ζ_x^* and μ_x^* are solutions to (3.24), $\mathbf{Z} = \mathbf{X} + \zeta_x^*$, $R_x(\cdot)$ is the retraction operation defined in Table 3.3, and $\mathrm{qf}(\mathbf{Z})$ is the Q-factor of a the QR decomposition of $\mathbf{Z}$. It should be emphasized that the update (3.25) is equivalent, in the neighborhood of the minimum, to the update proposed by Absil et al. (2002). In other words, the algorithm proposed by Absil et al. (2002) has the interpretation of a Riemannian steepest-descent algorithm with the metric (3.27).

When $\mathbf{A} \not\succ 0$

Consider first the case when $\mathbf{A} \prec 0$ that falls under Case II in Figure 3.4, suggesting (locally) the family of Riemannian metrics has the form

$$g_x(\xi_x, \zeta_x) = \omega\mathrm{Trace}(\xi_x^T\mathbf{A}\zeta_x) - \mathrm{Trace}(\xi_x^T\mathbf{B}\zeta_x\lambda_x), \tag{3.26}$$

where ξ_x and ζ_x are any tangent vectors in the tangent space of constraints and $\omega \in [0, 1)$. The expression for the least-squares Lagrange multiplier from (3.3) is $\lambda_x = \mathrm{Sym}((\mathbf{X}^T\mathbf{B}\mathbf{B}\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{B}\mathbf{A}\mathbf{X}))$, where $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\mathbf{D}) = (\mathbf{D}+\mathbf{D}^T)/2$. It should be noted that $-\lambda_x$ is only guaranteed to be positive definite locally in the neighborhood of the minimum. To circumvent the issue, we modify the metric (3.26) by replacing $-\lambda_x$ with $(\lambda_x^T\lambda_x)^{1/2}$ resulting in the metric

$$g_x(\xi_x, \zeta_x) = \omega \mathrm{Trace}(\xi_x^T \mathbf{A}\zeta_x) + \mathrm{Trace}(\xi_x^T \mathbf{B}\zeta_x(\lambda_x^T\lambda_x)^{1/2}), \tag{3.27}$$

where $(\lambda_x^T\lambda_x)^{1/2}$ is the matrix square root of $\lambda_x^T\lambda_x$ that is well defined as long as λ_x is full rank, and therefore the metric (3.27) is a smooth inner product. The modified metric (3.27) is also a good candidate for the case when $\mathbf{A}$ is symmetric indefinite since $(\lambda_x^T\lambda_x)^{1/2}$ is also positive definite in this case.

The proposed metric (3.27) with $\omega = 0$ generalizes the well-known power iteration algorithm for computing the dominant eigenvalues of a matrix (Golub and Van Loan, 1996, Section 8.2.1). Consider the case $\mathbf{B} = \mathbf{I}$, where the update of the Riemannian steepest-descent algorithm unit step-size has the characterization, after few computations, $x_+ = R_x(\mathbf{X}(\mathbf{I} + \lambda_x(\lambda_x^T\lambda_x)^{-1/2}) - \mathbf{AX}(\lambda_x^T\lambda_x)^{-1/2})$, where $R_x(\cdot)$ is the retraction operator defined in Table 3.3. Locally, in the neighborhood of the minimum, $\mathbf{I} + \lambda_x(\lambda_x^T\lambda_x)^{-1/2} \approx 0$, and therefore, the equivalent update is $x_+ = R_x(\mathbf{AX})$ which is the standard power iteration update (Golub and Van Loan, 1996, Section 8.2.1). In other words, the power algorithm has the interpretation of a Riemannian steepest-descent algorithm with the metric (3.27) with $\omega = 0$. Similarly, the steepest-descent algorithm with shifted version of the metric (3.27), i.e., for ω updated with iterations, generalizes the algorithm proposed of Absil et al. (2002).

It should be noted that a similar insight still holds when the quadratic cost is generalized to a strictly concave function, i.e., minimizing a concave cost (or maximizing a convex cost) with orthogonality constraints. For the metric with $\omega = 0$, i.e., taking only the constraint-related term, this is the essence of the *generalized power method* proposed by Journée et al. (2010).

3.3.2 A numerical illustration

As a numerical comparison, we consider the example proposed by Manton (2002, Section 8). $\mathbf{A}$ is a diagonal matrix of size 500×500 with entries evenly placed on the interval $[10, 11]$. $\mathbf{B}$ is chosen as the identity matrix of size 500×500. In Figure 3.5, we seek to compute the $r = 5$ smallest eigenvalues of $\mathbf{B}^{-1}\mathbf{A}$. The algorithms compared are the Riemannian steepest-descent algorithms with the standard metric (3.20) and the preconditioned Riemannian metric in (3.23) with the ω-updating procedure (3.17). Both the algorithms are stopped when either the norm of the gradient is below 10^{-8} or when they complete 500 iterations. Distances of the iterates to the solution ares plotted for the algorithms. The distance of an iterate $\mathbf{X}$ to the solution $\mathbf{X}_{\mathrm{opt}}$ is defined as the square root of the the sum of canonical angles between $\mathbf{X}$ and $\mathbf{X}_{\mathrm{opt}}$. In Matlab it is computed using the command `norm(acos(svd(orth(X)'*orth(Xopt))))`. Figure 3.5 shows that tuning the metric to the problem structure leads to improved performance.

3.4 Quadratic optimization with rank constraints

This class of problems have met with considerable interest in recent years. Applications include collaborative filtering (Rennie and Srebro, 2005), multivariate linear regression (Amit et al., 2007), dimensionality reduction (Cai et al., 2007), learning of low-rank distances (Kulis et al., 2009; Meyer et al., 2011b), filter

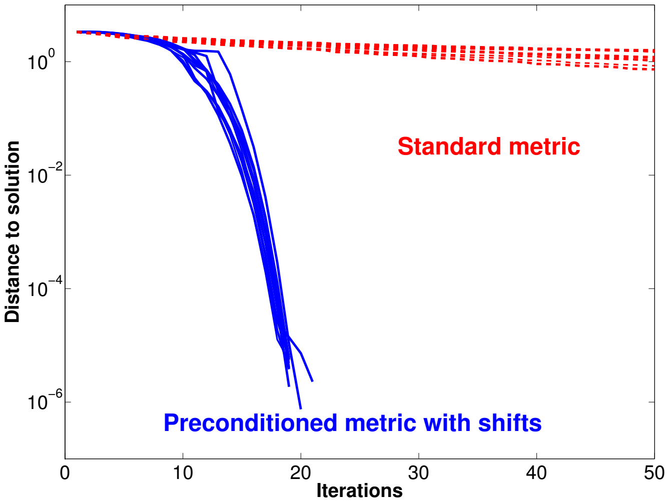


FIGURE 3.5: Benefits of the proposed metric (3.23) for the generalized eigenvalue problem to compute the extreme 5-dimensional subspace (corresponding to the smallest 5 eigenvalues) of the matrix pencil $(\mathbf{A}, \mathbf{B})$ of size 500×500. The problem instance is described in Section 3.3.2. Shown are 10 runs of the Riemannian steepest-descent algorithms with random initializations for the problem instance. The distance to the solution is defined as the square root of the the sum of canonical angles between the current subspace and the dominant 5-dimensional subspace of $\mathbf{B}^{-1}\mathbf{A}$.

design problems (Manton, 2002), model reduction in dynamical systems (Benner and Saak, 2013; Li and White, 2004; Vandereycken and Vandewalle, 2010), sparse principal components analysis (Burer and Monteiro, 2003; Journée et al., 2010), computing maximal cut of a graph (Burer and Monteiro, 2003; Journée et al., 2010), and low-rank matrix completion (Boumal and Absil, 2011; Keshavan et al., 2010; Ngo and Saad, 2012), to name a few.

In all those applications, SQP provides preconditioned Riemannian metrics.

A popular way to characterize the set of fixed-rank matrices is through fixed-rank matrix factorizations as mentioned in Chapter 2. Most matrix factorizations have symmetry properties that make them non-unique. $\mathbb{R}_r^{n\times m}$, the set of rank r of $n \times m$ matrices, is identified with structured (smooth and differentiable) quotient spaces. Figure 2.1 shows three different fixed-rank matrix factorizations and the quotient manifold structure of the set $\mathbb{R}_r^{n\times m}$.

To identify proper Riemannian metrics on the low-rank manifold, we consider minimization of a quadratic cost function on the low-rank manifold $\mathbb{R}_r^{n\times m}$. Specifically, we focus on the low-rank manifold parameterization $\mathbf{X} = \mathbf{G}\mathbf{H}^T$, where $\mathbf{X} \in \mathbb{R}_r^{n\times m}$, $\mathbf{G} \in \mathbb{R}_*^{n\times r}$ (the set of full column-rank matrices), and $\mathbf{H} \in \mathbb{R}_*^{m\times r}$. Other fixed-rank matrix factorizations are dealt with similarly. Consider the optimization problem

$$\min_{\mathbf{X}\in\mathbb{R}_r^{n\times m}} \quad \tfrac{1}{2}\mathrm{Trace}(\mathbf{X}^T\mathbf{A}\mathbf{X}\mathbf{B}) + \mathrm{Trace}(\mathbf{X}^T\mathbf{C}), \tag{3.28}$$

where $\mathbf{A} \succ 0$ of size $n \times n$, $\mathbf{B} \succ 0$ of size $m \times m$, and $\mathbf{C} \in \mathbb{R}^{n\times m}$. Positive definiteness of $\mathbf{A}$ and $\mathbf{B}$ implies that the cost function is bounded from below and is *convex* in $\mathbf{X}$. Invoking the low-rank parameterization

$\mathbf{X} = \mathbf{G}\mathbf{H}^T$, shown in Figure 2.1, the problem (3.28) translates to

$$\begin{array}{ll} \min\limits_{(\mathbf{G},\mathbf{H})\in\mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}} & \frac{1}{2}\mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{A}\mathbf{G}\mathbf{H}^T\mathbf{B}) + \mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{C}) \\ \text{subject to} & [(\mathbf{G},\mathbf{H})] \in \mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}/\mathrm{GL}(r), \end{array}$$

where the equivalences class $[(\mathbf{G},\mathbf{H})] = \{(\mathbf{G}\mathbf{M}^{-1}, \mathbf{H}\mathbf{M}^T) : \mathbf{M} \in \mathrm{GL}(r)\}$ and $\mathrm{GL}(r)$ is the set of $r \times r$ square matrices of non-zero determinant.

A standard way to handle this symmetry in the Riemannian framework is with the *natural metric* on the set $\mathbb{R}_*^{n\times r}$ (Absil et al., 2008, Section 3.6.4). Since the computational space $\mathcal{M}$ is the product space $\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$, the metric is

$$g_x(\eta_x,\xi_x) = \mathrm{Trace}((\mathbf{G}^T\mathbf{G})^{-1}\eta_{\mathbf{G}}^T\xi_{\mathbf{G}}) + \mathrm{Trace}((\mathbf{H}^T\mathbf{H})^{-1}\eta_{\mathbf{H}}^T\xi_{\mathbf{H}}),$$

where x has the matrix representation $(\mathbf{G},\mathbf{H}) \in \mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$ and ξ_x, η_x are vectors belonging to the tangent space $\mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}$, i.e., ξ_x has the matrix representation $(\xi_{\mathbf{G}},\xi_{\mathbf{H}}) \in \mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}$.

In contrast, we follow the developments in Section 3.2.3 to propose a family of metrics that takes the problem structure into account by exploiting the structure of the Lagrangian. Since the set $\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$ is an open subset of the space $\mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}$, the Lagrangian $\mathcal{L}(x,\lambda_x)$ is only characterized by the cost function, i.e.,

$$\begin{array}{rrl} & \mathcal{L}(x) = & \mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{A}\mathbf{G}\mathbf{H}^T\mathbf{B})/2 + \mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{C}) \\ \Rightarrow & \mathcal{L}_x(x) = & (\mathbf{A}\mathbf{G}\mathbf{H}^T\mathbf{B}\mathbf{H} + \mathbf{C}\mathbf{H}, \mathbf{B}\mathbf{H}\mathbf{G}^T\mathbf{A}\mathbf{G} + \mathbf{C}^T\mathbf{G}) \\ \Rightarrow & \mathrm{D}^2\mathcal{L}(x)[\xi_x] = & (\mathbf{A}\xi_{\mathbf{G}}\mathbf{H}^T\mathbf{B}\mathbf{H} + 2\mathbf{A}\mathbf{G}\mathrm{Sym}(\mathbf{H}^T\mathbf{B}\xi_{\mathbf{H}}) + \mathbf{C}\xi_{\mathbf{H}}, \\ & & \mathbf{B}\xi_{\mathbf{H}}\mathbf{G}^T\mathbf{A}\mathbf{G} + 2\mathbf{B}\mathbf{H}\mathrm{Sym}(\mathbf{G}^T\mathbf{A}\xi_{\mathbf{G}}) + \mathbf{C}^T\xi_{\mathbf{G}}), \end{array} \tag{3.29}$$

where x has the matrix representation $(\mathbf{G},\mathbf{H}) \in \mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$, ξ_x has the matrix representation $(\xi_{\mathbf{G}},\xi_{\mathbf{H}}) \in \mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}$, $\mathcal{L}_x(x)$ is the first-order derivative of $\mathcal{L}(x)$, $\mathrm{D}^2\mathcal{L}(x)[\xi_x]$ is the second-order derivative of $\mathcal{L}(x)$ applied in the direction ξ_x, and $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\mathbf{D}) = (\mathbf{D}^T + \mathbf{D})/2$. It is readily checked that the Lagrangian $\mathcal{L}(x)$ remains unchanged under the transformation $(\mathbf{G},\mathbf{H}) \mapsto (\mathbf{G}\mathbf{M}^{-1}, \mathbf{H}\mathbf{M}^T)$ for all $\mathbf{M} \in \mathrm{GL}(r)$. Subsequently, we have the following proposition for constructing the Riemannian metric for (3.28).

Proposition 3.3. *For the problem (3.28), the expression*

$$\begin{array}{rl} g_x(\xi_x,\eta_x) = & \langle\eta_{\mathbf{G}}, \mathbf{A}\xi_{\mathbf{G}}\mathbf{H}^T\mathbf{B}\mathbf{H} + 2\mathbf{A}\mathbf{G}\mathrm{Sym}(\mathbf{H}^T\mathbf{B}\xi_{\mathbf{H}}) + \mathbf{C}\xi_{\mathbf{H}}\rangle \\ & +\langle\eta_{\mathbf{H}}, \mathbf{B}\xi_{\mathbf{H}}\mathbf{G}^T\mathbf{A}\mathbf{G} + 2\mathbf{B}\mathbf{H}\mathrm{Sym}(\mathbf{G}^T\mathbf{A}\xi_{\mathbf{G}}) + \mathbf{C}^T\xi_{\mathbf{G}}\rangle, \end{array} \tag{3.30}$$

defines a family of Riemannian metrics, locally in the neighborhood of the minimum, where $x = (\mathbf{G},\mathbf{H}) \in \mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$, ξ_x, η_x any tangent vectors in the tangent space.

Proof. The proof follows directly by computing the second-order derivative of the Lagrangian (3.29). □

Matrix characterizations of various optimization related ingredients are summarized in Table 3.4. The retraction operator is the standard generalization of the retraction operator on the manifold $\mathbb{R}_*^{n\times r}$ defined by Absil et al. (2008, Example 3.6.4).

	$\min_{\substack{\mathbf{G}\in\mathbb{R}^{n\times r}\\ \mathbf{H}\in\mathbb{R}^{m\times r}}} \quad \mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{A}\mathbf{G}\mathbf{H}^T\mathbf{B})/2+\mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{C})$
Matrix representation	$x=(\mathbf{G},\mathbf{H})$
Computational space $\mathcal{M}$	$\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}$
Group action	$(\mathbf{G}\mathbf{M}^{-1},\mathbf{H}\mathbf{M}^T)\ ,\ \forall\mathbf{M}\in\mathrm{GL}(r)$
Quotient space	$\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}/\mathrm{GL}(r)$
Tangent vectors in $T_x\mathcal{M}$	$\xi_x=(\xi_{\mathbf{G}},\xi_{\mathbf{H}})\in\mathbb{R}^{n\times r}\times\mathbb{R}^{m\times r}$
Metric $g_x(\xi_x,\zeta_x)$ for any $\xi_x,\zeta_x\in T_x\mathcal{M}$	$g_x(\xi_x,\eta_x)=\langle\eta_{\mathbf{G}},\mathbf{A}\xi_{\mathbf{G}}\mathbf{H}^T\mathbf{B}\mathbf{H}+2\mathbf{A}\mathbf{G}\mathrm{Sym}(\mathbf{H}^T\mathbf{B}\xi_{\mathbf{H}})+\mathbf{C}\xi_{\mathbf{H}}\rangle$ $+\langle\eta_{\mathbf{H}},\mathbf{B}\xi_{\mathbf{H}}\mathbf{G}^T\mathbf{A}\mathbf{G}+2\mathbf{B}\mathbf{H}\mathrm{Sym}(\mathbf{G}^T\mathbf{A}\xi_{\mathbf{G}})+\mathbf{C}^T\xi_{\mathbf{G}}\rangle,$ or the metrics proposed in Section 3.4.1
Cost function	$f(x)=\mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{A}\mathbf{G}\mathbf{H}^T\mathbf{B})/2+\mathrm{Trace}(\mathbf{H}\mathbf{G}^T\mathbf{C})$
First − order derivative of $f(x)$	$f_x(x)=(\mathbf{S}\mathbf{H},\mathbf{S}^T\mathbf{G}),$ where $\mathbf{S}=\mathbf{A}\mathbf{G}\mathbf{H}^T\mathbf{B}+\mathbf{C}$
Search direction	$\arg\min_{\zeta_x\in T_x\mathcal{M}} \quad f(x)+\langle f_x(x),\zeta_x\rangle+\frac{1}{2}g_x(\zeta_x,\zeta_x)$
Retraction $R_x(\xi_x)$ that maps a search direction ξ_x onto $\mathcal{M}$	$(\mathbf{G}+\xi_{\mathbf{G}},\mathbf{H}+\xi_{\mathbf{H}})$

TABLE 3.4: Optimization-related ingredients for the problem (3.28). The numerical complexity per iteration of the Riemannian steepest-descent algorithm depends on solving for ζ_x, where sparsity in matrices $\mathbf{A}$ and $\mathbf{B}$ considerably reduces the computation cost. The retraction mapping is the cartesian product of the standard retraction mapping on the manifold $\mathbb{R}_*^{n\times r}$ (Absil et al., 2008, Example 3.6.4).

It should be noted that numerical performance of algorithms depend on computing the Riemannian gradient efficiently with the metric (3.30). This may become a numerically cumbersome task due to a number of coupled terms that are involved in the metric (3.30). However, below we show that the problem structure can be further exploited to decompose the metric (3.30) into a locally dominating part with a simpler metric structure and a *weighted* remainder. The dominant approximation may be preferred in a number of situations.

3.4.1 Metric tuning and shift policies

It should be emphasized that the cost function in (3.28) is *convex and quadratic* in $\mathbf{X}$. Consequently, the cost function is also convex and quadratic in the arguments $(\mathbf{G},\mathbf{H})$ *individually*. As a consequence, the block diagonal elements of the second-order derivative $\mathcal{L}_{xx}(x)$ of the Lagrangian (3.29) is strictly positive definite. This enables us to construct a family of Riemannian metrics with *shifts* of the form

$$\begin{aligned} g_x(\xi_x,\eta_x) = \ & \omega\langle \eta_{\mathbf{G}}, 2\mathbf{A}\mathbf{G}\mathrm{Sym}(\mathbf{H}^T\mathbf{B}\xi_{\mathbf{H}}) + \mathbf{C}\xi_{\mathbf{H}}\rangle \\ & +\omega\langle \eta_{\mathbf{H}}, 2\mathbf{B}\mathbf{H}\mathrm{Sym}(\mathbf{G}^T\mathbf{A}\xi_{\mathbf{G}}) + \mathbf{C}^T\xi_{\mathbf{G}}\rangle \\ & + \underbrace{\langle \eta_{\mathbf{G}}, \mathbf{A}\xi_{\mathbf{G}}\mathbf{H}^T\mathbf{B}\mathbf{H}\rangle + \langle \eta_{\mathbf{H}}, \mathbf{B}\xi_{\mathbf{H}}\mathbf{G}^T\mathbf{A}\mathbf{G}\rangle,}_{\text{Block diagonal approximation of } \mathcal{L}_{xx}(x)} \end{aligned} \tag{3.31}$$

where $x = (\mathbf{G},\mathbf{H}) \in \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}$, ξ_x, η_x are tangent vectors in $\mathbb{R}^{n\times r} \times \mathbb{R}^{m\times r}$, and $\omega \in [0,1)$ is updated with iterations (Section 3.2.4). This implies that we exploit the full second-order information of the problem only in the neighborhood of the minimum. Away from the neighborhood, the metric (3.31) with $\omega = 0$ becomes a good metric candidate as $\mathbf{H}^T\mathbf{B}\mathbf{H}$ and $\mathbf{G}^T\mathbf{A}\mathbf{G}$ are positive definite for all $(\mathbf{G},\mathbf{H}) \in \mathbb{R}_*^{n\times r} \times \mathbb{R}_*^{m\times r}$. The other benefit of ω being 0 is that the resulting metric has a simpler matrix characterization, and hence, may be preferred in numerically demanding instances.

3.4.2 Symmetric positive definite matrices

A popular subset of fixed-rank matrices is the set of symmetric positive semidefinite matrices (Burer and Monteiro, 2003; Journée et al., 2010; Meyer et al., 2011b). The set $\mathrm{S}_+(r,n)$, the set of rank-r symmetric positive semidefinite matrices of size $n \times n$, is equivalent to the set $\mathbb{R}_r^{n\times m}$ with symmetry imposed on the rows and columns, and therefore, it admits a number of factorizations similar to those in Figure 2.1. Consequently, the low-rank parameterization discussed earlier, in the context of the general case, has the counterpart $\mathbf{X} = \mathbf{Y}\mathbf{Y}^T$, where $\mathbf{X} \in \mathrm{S}_+(r,n)$ and $\mathbf{Y} \in \mathbb{R}_*^{n\times r}$ (full column rank matrices of size $n \times r$). This parameterization is not unique as $\mathbf{X} \in \mathrm{S}_+(r,n) = \mathbf{Y}\mathbf{Y}^T$ remains unchanged under the transformation $\mathbf{Y} \mapsto \mathbf{Y}\mathbf{O}$ for any $O \in \mathcal{O}(r)$, where $\mathcal{O}(r)$ is set of orthogonal matrices of size $r \times r$ such that $\mathbf{O}\mathbf{O}^T = \mathbf{O}^T\mathbf{O} = \mathbf{I}$. The resulting search space is, thus, the set of equivalence classes $[\mathbf{Y}] = \{\mathbf{Y}\mathbf{O} : \mathbf{O} \in \mathcal{O}(r)\}$ and is the quotient manifold $\mathbb{R}_*^{n\times r}/\mathcal{O}(r)$ (Journée et al., 2010). Finally, we have the following proposition that summarizes the discussion for the case of symmetric positive semidefinite matrices.

Proposition 3.4. *Consider the optimization problem*

$$\begin{array}{ll} \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times n}} & \frac{1}{2}\mathrm{Trace}(\mathbf{X}\mathbf{A}\mathbf{X}\mathbf{B}) + \mathrm{Trace}(\mathbf{X}\mathbf{C}) \\ \text{subject to} & \mathbf{X} \in \mathrm{S}_+(r,n), \end{array} \tag{3.32}$$

where $\mathbf{A},\mathbf{B} \succ 0$ of size $n \times n$ and $\mathbf{C} \in \mathbb{R}^{n\times m}$ is a symmetric matrix. Consider also the factorization $\mathbf{X} = \mathbf{Y}\mathbf{Y}^T$ of rank-r symmetric positive semidefinite matrices to encode the rank constraint, where $\mathbf{Y} \in \mathbb{R}_*^{n\times r}$ (full column-rank matrices).

The family of Riemannian metrics, locally in the neighborhood of the minimum, for the problem (3.32) has the form

$$\begin{aligned} g_x(\xi_x,\eta_x) = \ & \omega\langle \eta_x, 2\mathbf{A}\mathbf{Y}\mathrm{Sym}(\mathbf{Y}^T\mathbf{B}\xi_x) + 2\mathbf{B}\mathbf{Y}\mathrm{Sym}(\mathbf{Y}^T\mathbf{A}\xi_x) + 2\mathbf{C}\xi_x\rangle \\ + \ & \underbrace{\langle \eta_x, \mathbf{A}\xi_x\mathbf{Y}^T\mathbf{B}\mathbf{Y} + \mathbf{B}\xi_x\mathbf{Y}^T\mathbf{A}\mathbf{Y}\rangle,}_{\text{Dominant positive definite approximation of } \mathcal{L}_{xx}(x)} \end{aligned} \tag{3.33}$$

where $x = \mathbf{Y} \in \mathbb{R}_*^{n\times r}$, ξ_x, η_x any tangent vectors in the tangent space $\mathbb{R}^{n\times r}$, $\mathcal{L}_{xx}(x)$ is the second-order derivative of the Lagrangian, $\omega \in [0, 1)$ is a positive weight that is updated (increased) with increasing iteration number (Section 3.2.4). Beyond the neighborhood, the metric (3.33) with $\omega = 0$ becomes a good metric candidate as $\mathbf{Y}^T\mathbf{B}\mathbf{Y}$ and $\mathbf{Y}^T\mathbf{A}\mathbf{Y}$ are positive definite for all $\mathbf{Y} \in \mathbb{R}_*^{n\times r}$.

Proof. The proof follows directly from the earlier discussion in Section 3.4.1. □

3.4.3 A numerical illustration

We showcase the Riemannian preconditioning approach for computing low-rank solutions to the generalized Lyapunov equation of the form

$$\mathbf{AXB} + \mathbf{BXA} = \mathbf{C}, \tag{3.34}$$

where $\mathbf{A}, \mathbf{B} \succ 0$, and $\mathbf{C}$ is a low-rank symmetric positive semidefinite matrix. Matrices have appropriate dimensions. $\mathbf{A}$ is referred to as the *system matrix* and $\mathbf{B}$ is referred to as the *mass matrix*. As a result, the solution of (3.34) is also expected to be low-rank symmetric positive semidefinite (Benner and Saak, 2013; Li and White, 2004; Vandereycken and Vandewalle, 2010).

To compute low-rank solutions to (3.34), we minimize a suitable cost function to over the set of rank-r symmetric and positive semidefinite matrices $\mathrm{S}_+(r, n)$. For the present case of interest, one could either minimize the *energy norm* $\mathrm{Trace}(\mathbf{XAXB}) - \mathrm{Trace}(\mathbf{XC})$ or the *residual norm* $\|\mathbf{AXB} + \mathbf{BXA} = \mathbf{C}\|_F^2$ (Vandereycken and Vandewalle, 2010). Here we only show minimization of the energy norm over $\mathrm{S}_+(r, n)$. Note that this is similar to the optimization problem (3.32) and we have the characterization of the family of metrics in Proposition 3.4.

In contrast to the proposed preconditioned metric (3.33), an alternative is to consider the standard Euclidean metric, i.e.,

$$g_x(\xi_x, \zeta_x) = \mathrm{Trace}(\zeta_x^T \xi_x), \tag{3.35}$$

where $x = \mathbf{Y}$ and ξ_x and ζ_x are tangent vectors. This is, for example, the Riemannian metric proposed by Journée et al. (2010). It is invariant to the group action $\mathbf{Y} \mapsto \mathbf{YO}$ for any $\mathbf{O} \in \mathcal{O}(r)$. Although the alternative choice (3.35) is appealing for its numerical simplicity, the following test case clearly illustrates the benefits of the Riemannian preconditioning approach.

We consider the standard benchmark problem from Penzl (1999, Example 2.1) that corresponds to discretization of a one-dimensional heat equation from heat flow in a thin rod. For this example, $\mathbf{A}$ is a tridiagonal matrix of size 500×500. The main diagonal of $\mathbf{A}$ has all the elements equal to 2. In addition, the first diagonals below and above the main diagonal of $\mathbf{A}$ have all the entries equal to -1. $\mathbf{A}$ is an ill-conditioned matrix with condition number 10^5. The mass matrix $\mathbf{B}$ is an identity matrix of size 500×500. The matrix $\mathbf{C}$ is a rank one matrix of the form ee^T, where e^T is a row vector of length 500 of the form $[0\ 0\ \ldots\ 0\ 1]$. We seek to find a rank-5 that best solves the generalized Lyapunov equation (3.34). Both the algorithms are stopped when either the norm of the gradient is below 10^{-8} or when they complete 500 iterations. The plots in Figure 3.6 show the progress of relative residual $\|\mathbf{AXB}+\mathbf{BXA}-\mathbf{C}\|_F/\|\mathbf{C}\|_F$ with iterations over 10 random initializations, where $\|\cdot\|_F$ is the Frobenius norm of a matrix and $\mathbf{X} = \mathbf{YY}^T$.

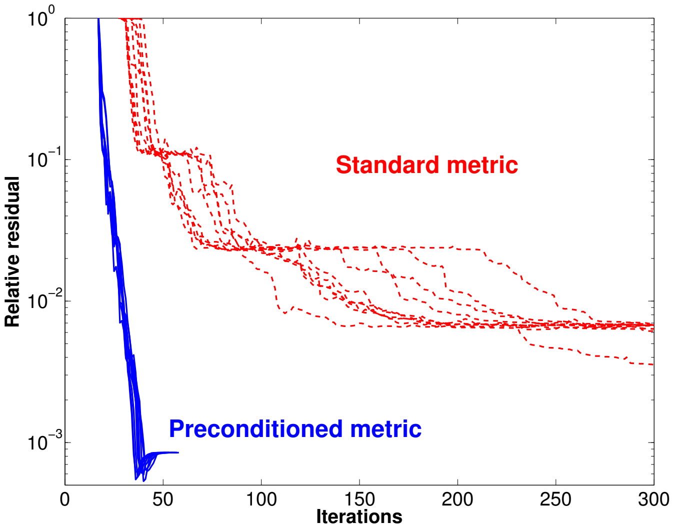


FIGURE 3.6: The generalized low-rank Lyapunov equation problem (3.34). The test case is the benchmark problem from Penzl (1999, Example 2.1) with $n = 500$. The proposed Riemannian preconditioning approach with the metric (3.33) and $\omega = 0$ drastically improves the performance over the algorithm based on the standard metric (3.35). $\omega = 0$ leads to a simpler metric structure that can be exploited in a large-scale setup. Here we show the convergence of the relative residual $\|\mathbf{AXB}+\mathbf{BXA}-\mathbf{C}\|_F/\|\mathbf{C}\|_F$ (not the cost function $\mathrm{Trace}(\mathbf{XAXB}) - \mathrm{Trace}(\mathbf{XC})$) that is often used as a measure of recovery.

The Riemannian algorithm with the metric (3.33) and $\omega = 0$ convincingly outperforms the algorithm based on the standard metric (3.35) in Figure 3.6 for a number of runs.

3.5 Chapter summary

The chapter addresses the important issue of selecting a metric in the Riemannian optimization framework. We have shown that sequential quadratic programming provides an insight into selecting a family of Riemannian metrics that takes the second-order information of the problem. Quadratic optimization with orthogonality or rank constraints provides a class of nonconvex problems for which the method is particularly insightful, thanks to the local convexity of the cost and constraint when taken separately. In those instances, the notion of metric tuning connects to a number of existing algorithms and provides a geometric interpretation of a number of "shift" policies in numerical linear algebra.

The results of this chapter have been reported in the technical report (Mishra and Sepulchre, 2014b).

Chapter 4

Riemannian conjugate-gradients for low-rank matrix completion

In this chapter, we propose efficient conjugate-gradient algorithms for the low-rank matrix completion problem (2.2). Following the notion of metric tuning introduced in Chapter 3, we select the metric for the problem (2.2) and discuss two different geometries studied in sections 2.2.1.1 and 2.2.1.2. The problem (2.2) is rewritten below for the sake of completeness.

$$\begin{array}{ll} \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times m}} & \frac{1}{|\Omega|}\|\mathcal{P}_\Omega(\mathbf{X})-\mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 \\ \text{subject to} & \operatorname{rank}(\mathbf{X})=r, \end{array} \tag{4.1}$$

where Ω is the subset of the complete set of indices $\{(i,j) : i \in \{1,\ldots,n\} \text{ and } j \in \{1,\ldots,m\}\}$ for which the entries are known, $\|\cdot\|_F$ is the Frobenius norm, and the operator $\mathcal{P}_\Omega$, called the orthogonal sampling operator, is introduced for notational convenience and is defined as $\mathcal{P}_\Omega(\mathbf{X})_{ij} = \mathbf{X}_{ij}$ for $(i,j)\in\Omega$ and $\mathcal{P}_\Omega(\mathbf{X})_{ij}=0$ otherwise. It should be noted that $\|\mathcal{P}_\Omega(\mathbf{X})-\mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 = \sum\limits_{(i,j)\in\Omega}(\mathbf{X}_{ij}-\mathbf{X}^\star_{ij})^2$.

The organization of the chapter is as follows. A brief motivation of employing metric tuning in low-rank matrix completion is presented in Section 4.1. Section 4.2 discusses the quotient nature of two fixed-rank matrix factorizations and paves way to propose novel Riemannian metrics (4.2), that are specifically tailored to the cost function of the matrix completion problem (4.1). A template of an off-the-shelf Riemannian conjugate-gradient on a manifold method is shown in Algorithm 1. Concrete matrix formulas for the implementation of the Riemannian nonlinear conjugate-gradient algorithm are listed in Section 4.4. Section 4.6.1 shows the efficacy of the proposed metrics as against conventional choices. In Section 4.6.2 we make connections with state-of-the-art algorithms. The numerical comparisons in Section 4.6 show the good performance of the resulting algorithms across different problem instances, focusing in particular, on scarcely sampled and ill-conditioned problems. A Matlab implementation of our proposed algorithms is available from `http://www.montefiore.ulg.ac.be/~mishra/pubs.html`.

4.1 Motivation

Out of a lot of different works on matrix completion and low-rank optimization, we are primarily motivated by the recent algorithms from Keshavan et al. (2010); Ngo and Saad (2012); Wen et al. (2012) that have shown better performance in a number of challenging scenarios. Keshavan et al. (2010) use the fixed-rank factorization $\mathbf{X} = \mathbf{USV}^T$ to embed the rank constraint, where $\mathbf{X}$ is a rank r matrix of size $n \times m$, $\mathbf{U}$ and $\mathbf{V}$ are column-orthonormal full rank matrices of size $n \times r$ and $m \times r$, and $\mathbf{S} \in \mathbb{R}^{r \times r}$. At each iteration, Keshavan et al. (2010) first update $\mathbf{U}$ and $\mathbf{V}$ on the *bi-Grassmann* manifold $\mathrm{Gr}(r, n) \times \mathrm{Gr}(r, m)$, where $\mathrm{Gr}(r, n)$ is the set of r-dimensional subspaces in $\mathbb{R}^n$. Subsequently, a least-squares problem is solved to update $\mathbf{S}$. Building upon the work of Keshavan et al. (2010), Ngo and Saad (2012) propose a *matrix scaling* on the bi-Grassmann manifold to *accelerate* the algorithm of Keshavan et al. (2010). In particular, Ngo and Saad (2012) motivate the matrix scaling as an *adaptive preconditioner* for the optimization problem (4.1) and implement a conjugate-gradient algorithm. The same matrix scaling also appears the algorithm proposed by Wen et al. (2012) where the authors motivate their *Gauss-Seidel* algorithm on the fixed-rank matrix factorization $\mathbf{X} = \mathbf{GH}^T$, where $\mathbf{G}$ and $\mathbf{H}$ are full column-rank matrices of size $n \times r$ and $m \times r$, respectively. $\mathbf{G}$ and $\mathbf{H}$ are updated alternatively and update of each of the arguments, while fixing the other, admits a closed form expression. A potential limitation of these algorithms is that they are alternating minimization and first-order algorithms, and extending them to other classes of optimization methods is not trivial.

In order to get the best of these methods, we reinterpret the matrix-scaling of Wen et al. (2012) and Ngo and Saad (2012) as an instance of *tuned* metrics in the Riemannian framework for matrix completion. We then exploit the metric tuning concept, proposed in Chapter 3, to propose novel Riemannian geometries for fixed-rank matrix factorizations, *trading-off* second-order information of the cost function with symmetries in the search space. As a result, the proposed Riemannian metrics confer a geometric foundation to the algorithms of Ngo and Saad (2012); Wen et al. (2012). Once the Riemannian metrics are proposed, we follow Absil et al. (2008, Chapters 3, 5 and, 8) to list various optimization-related ingredients and their concrete matrix formulas that are required to implement an off-the-shelf Riemannian conjugate-gradient algorithm, Algorithm 1. Although the new Riemannian geometries enable us to propose second-order methods like the Riemannian trust-region method, we specifically focus on *conjugate-gradients* as they offer an appropriate balance between convergence and computational cost. They have shown superior performance in our examples.

4.2 Metric tuning for low-rank matrix completion

We parameterize any $n \times m$ rank-r matrix $\mathbf{X} \in \mathbb{R}_r^{n \times m}$ as

$$\begin{array}{ll} \mathbf{X} = \mathbf{GH}^T & \text{(two} - \text{factor factorization)} \\ \mathbf{X} = \mathbf{URV}^T & \text{(three} - \text{factor factorization),} \end{array} \tag{4.2}$$

where $(\mathbf{G}, \mathbf{H}) \in \mathbb{R}_*^{n \times r} \times \mathbb{R}_*^{m \times r}$ and $(\mathbf{U}, \mathbf{R}, \mathbf{V}) \in \mathrm{St}(r, n) \times \mathrm{GL}(r) \times \mathrm{St}(r, m)$. The factorizations have been discussed in Section 2.2.1. Here $\mathbb{R}_*^{n \times r}$ is the set of full column rank matrices, $\mathrm{St}(r, n)$ is the set of

	Two − factor decomposition $\mathbf{X} = \mathbf{GH}^T$	Three − factor decomposition $\mathbf{X} = \mathbf{URV}^T$
Matrix representation	$x = (\mathbf{G}, \mathbf{H})$	$x = (\mathbf{U}, \mathbf{R}, \mathbf{V})$
Total space $\mathcal{M}$	$\mathbb{R}_*^{n \times r} \times \mathbb{R}_*^{m \times r}$	$\mathrm{St}(r, n) \times \mathrm{GL}(r) \times \mathrm{St}(r, m)$
Group action	$(\mathbf{GM}^{-1}, \mathbf{HM}^T)$ $\mathbf{M} \in \mathrm{GL}(r)$	$(\mathbf{UO}_1, \mathbf{O}_1^T \mathbf{RO}_2, \mathbf{VO}_2)$ $\mathbf{O}_1, \mathbf{O}_2 \in \mathcal{O}(r)$
Equivalence class $[x]$	$[(\mathbf{G}, \mathbf{H})] = \{(\mathbf{GM}^{-1}, \mathbf{HM}^T) :$ $\mathbf{M} \in \mathrm{GL}(r)\}$	$[(\mathbf{U}, \mathbf{R}, \mathbf{V})] = \{(\mathbf{UO}_1, \mathbf{O}_1^T \mathbf{RO}_2, \mathbf{VO}_2) :$ $(\mathbf{O}_1, \mathbf{O}_2) \in \mathcal{O}(r) \times \mathcal{O}(r)\}$
Quotient space $\mathcal{M}/\sim$	$\mathbb{R}_*^{n \times r} \times \mathbb{R}_*^{m \times r}$ $/\mathrm{GL}(r)$	$\mathrm{St}(r, n) \times \mathrm{GL}(r) \times \mathrm{St}(r, m)$ $/(\mathcal{O}(r) \times \mathcal{O}(r))$

Table 4.1: Fixed-rank matrix factorizations and their quotient manifold representations. The action of Lie groups $\mathrm{GL}(r)$ and $(\mathcal{O}(r) \times \mathcal{O}(r))$ make the quotient spaces smooth quotient manifolds (Lee, 2003, Theorem 9.16). Here $\mathbb{R}_*^{n \times r}$ is the set of full column rank matrices, $\mathrm{St}(r, n)$ is the set of matrices of size $n \times r$ with orthonormal columns, and $\mathrm{GL}(r)$ is the set of $r \times r$ non-singular matrices.

matrices of size $n \times r$ with orthonormal columns, and $\mathrm{GL}(r)$ is the set of $r \times r$ non-singular matrices. For each of the factorizations in (4.2), the matrix characterizations of the total space (computational space) $\mathcal{M}$ equipped with the equivalence relation $\sim$ and the resulting quotient space $\mathcal{M}/\sim$ are shown in Table 4.1. The equivalence class of a given point $x \in \mathcal{M}$ is the represented by the set $[x] = \{y \in \mathcal{M} : y \sim x\}$. The set $\mathcal{M}/\sim$ contains all such equivalence classes.

It should be stated that the total space of each of the two considered fixed-rank matrix factorizations (4.2) admits a product structure of well-known matrix manifolds, e.g., $\mathbb{R}_*^{n \times r}$, $\mathrm{St}(r, n)$, and $\mathrm{GL}(r)$. Each of these manifolds is smooth and, therefore, their product structures also preserve the smoothness property (Absil et al., 2008, Section 3.1.6). Similarly, the quotient spaces generated by smooth group actions (Table 4.1) have the structure of a smooth quotient manifold (Lee, 2003, Theorem 9.16).

The tangent space $T_x\mathcal{M}$ at $x \in \mathcal{M}$ admits a product structure following the product structure of the total space $\mathcal{M}$. Because the total space is a product space of matrix manifolds, its tangent space $T_x\mathcal{M}$ at x embodies the product space of the tangent spaces of individual manifolds, e.g, $\mathbb{R}_*^{n \times r}$, $\mathrm{St}(r, n)$, and $\mathrm{GL}(r)$, the characterizations of which are well-known. Refer Edelman et al. (1998, Section 2.2) or Absil et al. (2008, Example 3.5.2) for the characterization of the tangent space of $\mathrm{St}(r, n)$. The tangent spaces of $\mathbb{R}_*^{n \times r}$ and $\mathrm{GL}(r)$ are $\mathbb{R}^{n \times r}$ and $\mathbb{R}^{r \times r}$, respectively.

4.2.1 A simpler cost function

The abstract quotient search space $\mathcal{M}/\sim$ is given the structure of a Riemannian quotient manifold by choosing a Riemannian metric, that respects the symmetry (shown in Table 4.1) on manifold $\mathcal{M}$ (Absil et al., 2008). The metric defines an inner product between tangent vectors on the tangent space $T_x\mathcal{M}$. Building upon the product structure of the total space $\mathcal{M}$, a valid metric on $T_x\mathcal{M}$ is derived from choosing *natural* metrics of the individual manifolds, the characterization of which are also well-known (this is discussed in Section 4.6.1). However, this is not the only choice. Here we derive different metrics that better exploit the structure of the cost function at hand (4.1). It should be noted that the second-order derivative of the cost function $\|\mathcal{P}_\Omega(\mathbf{X})-\mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ in (4.1) with respect to $\mathbf{X}\in\mathbb{R}^{n\times m}$ is computationally costly to deal with and is rank deficient (Buchanan and Fitzgibbon, 2005). To circumvent the issue, we consider a simplified (but related) version of the cost function in (4.1). Specifically, consider the least-squares cost function $\|\mathbf{X}-\mathbf{X}^\star\|_F^2/2$ that is a simplification of (4.1) by *assuming* that Ω contains all the indices. The cost function $\|\mathbf{X}-\mathbf{X}^\star\|_F^2/2$ now acts as a surrogate for $\|\mathcal{P}_\Omega(\mathbf{X})-\mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ which we exploit to understand the underlying structure. It should be emphasized that the cost function $\|\mathbf{X}-\mathbf{X}^\star\|_F^2/2$ is *strictly convex* and *quadratic* in $\mathbf{X}$ and, therefore, also strictly convex and quadratic in each of the *individual* arguments of different matrix factorizations. For example, $\|\mathbf{G}\mathbf{H}^T-\mathbf{X}^\star\|_F^2$ is strictly convex in $\mathbf{G}\in\mathbb{R}_*^{n\times r}$ and $\mathbf{H}\in\mathbb{R}_*^{m\times r}$ individually for $\mathbf{X}=\mathbf{G}\mathbf{H}^T$ (4.2). Minimizing $\|\mathbf{G}\mathbf{H}^T-\mathbf{X}^\star\|_F^2$ with respect to $\mathbf{G}$ and $\mathbf{H}$ amounts to computing the *dominant* rank-r subspace of $\mathbf{X}^\star$.

4.2.2 A novel Riemannian metric

Observing that $\|\mathbf{X}-\mathbf{X}^\star\|_F^2/2=\|\mathbf{X}^\star\|_F^2/2+\mathrm{Trace}(\mathbf{X}^T\mathbf{X})/2-\mathrm{Trace}(\mathbf{X}^T\mathbf{X}^\star)$, we apply propositions 3.3 and 3.2, that exploit quadratic cost functions with orthogonality and rank constraints, to the simplified optimization problem

$$\begin{array}{ll}\displaystyle\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} & \frac{1}{2}\mathrm{Trace}(\mathbf{X}^T\mathbf{X})-\mathrm{Trace}(\mathbf{X}^T\mathbf{X}^\star)\\ \text{subject to} & \mathrm{rank}(\mathbf{X})=r\end{array}\tag{4.3}$$

in order to propose metrics for the original problem (4.1). Forming Lagrangians and computing their second-order derivatives as in Section 3.2.3 leads to the following matrix characterizations of the Riemannian metrics for the two particular factorizations of interest in this chapter:

$$\begin{array}{rl}\text{Two}-\text{factor}: & \\ g_x(\xi_x,\eta_x)= & \omega\langle\eta_{\mathbf{G}},2\mathbf{G}\mathrm{Sym}(\mathbf{H}^T\xi_{\mathbf{H}})-\mathbf{X}^\star\xi_{\mathbf{H}}\rangle\\ & +\omega\langle\eta_{\mathbf{H}},2\mathbf{H}\mathrm{Sym}(\mathbf{G}^T\xi_{\mathbf{G}})-\mathbf{X}^{\star T}\xi_{\mathbf{G}}\rangle\\ & +\quad\underbrace{\langle\eta_{\mathbf{G}},\xi_{\mathbf{G}}\mathbf{H}^T\mathbf{H}\rangle+\langle\eta_{\mathbf{H}},\xi_{\mathbf{H}}\mathbf{G}^T\mathbf{G}\rangle,}_{\text{Block diagonal approximation of the simplified cost function}}\end{array}\tag{4.4a}$$

$$\begin{array}{rl}\text{Three}-\text{factor}: & \\ g_x(\xi_x,\eta_x)= & \omega\langle\eta_{\mathbf{U}},2\mathbf{U}\mathbf{R}\mathrm{Sym}(\mathbf{V}^T\xi_{\mathbf{V}})\mathbf{R}^T+2\mathbf{U}\mathrm{Sym}(\mathbf{R}\xi_{\mathbf{R}}^T)-\mathbf{X}^\star(\xi_{\mathbf{V}}\mathbf{R}^T+\mathbf{V}{\xi_{\mathbf{R}}}^T)\rangle\\ & +\omega\langle\eta_{\mathbf{R}},2\mathrm{Sym}(\mathbf{U}^T\xi_{\mathbf{U}})\mathbf{R}+2\mathbf{R}\mathrm{Sym}(\mathbf{V}^T\xi_{\mathbf{V}})-\mathbf{U}^T\mathbf{X}^\star\xi_{\mathbf{V}}-\xi_{\mathbf{U}}^T\mathbf{X}^\star\mathbf{V}\rangle\\ & +\omega\langle\eta_{\mathbf{V}},2\mathbf{V}\mathbf{R}^T\mathrm{Sym}(\mathbf{U}^T\xi_{\mathbf{U}})\mathbf{R}+2\mathbf{V}\mathrm{Sym}(\mathbf{R}^T\xi_{\mathbf{R}})-\mathbf{X}^{\star T}(\xi_{\mathbf{U}}\mathbf{R}+\mathbf{U}\xi_{\mathbf{R}})\rangle\\ & -\omega\langle\eta_{\mathbf{U}},\xi_{\mathbf{U}}\mathrm{Sym}(\mathbf{R}\mathbf{R}^T-\mathbf{U}^T\mathbf{X}^\star\mathbf{V}\mathbf{R}^T)\rangle\\ & -\omega\langle\eta_{\mathbf{V}},\xi_{\mathbf{V}}\mathrm{Sym}(\mathbf{R}^T\mathbf{R}-\mathbf{V}^T\mathbf{X}^{\star T}\mathbf{U}\mathbf{R})\rangle\\ & +\quad\underbrace{\langle\eta_{\mathbf{U}},\xi_{\mathbf{U}}\mathbf{R}\mathbf{R}^T\rangle+\langle\eta_{\mathbf{R}},\xi_{\mathbf{U}}\rangle+\langle\eta_{\mathbf{V}},\xi_{\mathbf{V}}\mathbf{R}^T\mathbf{R}\rangle,}_{\text{Block diagonal approximation of the simplified cost function}}\end{array}\tag{4.4b}$$

where $g_x:T_x\mathcal{M}\times T_x\mathcal{M}\to\mathbb{R}:(\xi_x,\eta_x)\mapsto g_x(\xi_x,\eta_x)$ is the proposed Riemannian metric on $\mathcal{M}$, $\omega\in[0,1)$, $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product, and $\mathrm{Sym}(\cdot)$ extract the symmetric part of a square matrix, e.g., $\mathrm{Sym}(\mathbf{A})=(\mathbf{A}+\mathbf{A}^T)/2$ for a square matrix $\mathbf{A}$. An element $x\in\mathcal{M}$ has the matrix characterization

$(\mathbf{G}, \mathbf{H})$ and $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ for the two-factor and three-factor fixed-rank factorizations, respectively. Similarly, $(\xi_{\mathbf{G}}, \xi_{\mathbf{H}})$ and $(\xi_{\mathbf{U}}, \xi_{\mathbf{R}}, \xi_{\mathbf{V}})$ are the matrix representations of the tangent vector $\xi_x \in T_x\mathcal{M}$ (shown in Table 4.2).

In the decomposition of (4.4), the terms not multiplied by the parameter ω are the block diagonal terms of the Hessian, which are positive definite because the quadratic cost function is convex in each of the variables. Mimicking the developments in Section 3.4, all the remaining terms are multiplied by the weighting parameter as they do not guarantee positive definiteness of the metric away from the minimum, i.e., these contain off diagonal terms of the cost Hessian (4.4a and 4.4b) and terms from the constraint Hessian (4.4b).

Because the motivation in this chapter is to tackle large-scale problems and because the metric derivation does not capture the exact cost function anyway, we adopt the default choice $\omega = 0$ in (4.4), which means its online adaptation as in Chapter 3 (Section 3.2.4 in particular) is not considered worth the extra computation effort that it involves. Retaining only the positive definite block diagonal term from the full second-order information of the problems (4.3) thus leads to choice

$$\text{Two}-\text{factor}: \quad g_x(\xi_x, \eta_x) = \langle \eta_{\mathbf{G}}, \xi_{\mathbf{G}}\mathbf{H}^T\mathbf{H}\rangle + \langle \eta_{\mathbf{H}}, \xi_{\mathbf{H}}\mathbf{G}^T\mathbf{G}\rangle \tag{4.5a}$$

$$\text{Three}-\text{factor}: \quad g_x(\xi_x, \eta_x) = \langle \eta_{\mathbf{U}}, \xi_{\mathbf{U}}\mathbf{R}\mathbf{R}^T\rangle + \langle \eta_{\mathbf{R}}, \xi_{\mathbf{U}}\rangle + \langle \eta_{\mathbf{V}}, \xi_{\mathbf{V}}\mathbf{R}^T\mathbf{R}\rangle \tag{4.5b}$$

for the problem (4.1), where $g_x : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ is the metric imposed on $T_x\mathcal{M}$, $\xi_x, \eta_x \in T_x\mathcal{M}$, and x has the matrix characterizations $(\mathbf{G}, \mathbf{H})$ and $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ for the two-factor and three-factor fixed-rank factorizations (4.2), respectively.

4.3 Relevant matrix characterizations

The metric $g_{[x]} : T_{[x]}(\mathcal{M}/\sim) \times T_{[x]}(\mathcal{M}/\sim) \to \mathbb{R} : (\xi_{[x]}, \eta_{[x]}) \mapsto g_{[x]}(\xi_{[x]}, \eta_{[x]})$ on the abstract tangent space $T_{[x]}(\mathcal{M}/\sim)$ of the quotient manifold $\mathcal{M}/\sim$ is the restriction of the proposed metric g_x (4.5) to the horizontal space $\mathcal{H}_x$, i.e., to a subspace of $T_x\mathcal{M}$ that characterizes $T_{[x]}(\mathcal{M}/\sim)$. Equivalently, $g_{[x]}(\xi_{[x]}, \eta_{[x]}) := g_x(\xi_x, \eta_x)$, where

- $[x] \in \mathcal{M}/\sim$,
- $\xi_{[x]}$ and $\eta_{[x]}$ are abstract tangent vectors in $T_{[x]}(\mathcal{M}/\sim)$, and
- ξ_x, η_x are the matrix characterizations of $\xi_{[x]}$ and $\eta_{[x]}$, respectively, in $\mathcal{H}_x$.

Consequently, the abstract quotient manifold $\mathcal{M}/\sim$ has the structure of a Riemannian submersion of $\mathcal{M}$ with the metric g (4.5) (Absil et al., 2008, Chapter 3). This particular structure enables us to compute the following relevant matrix representations relating to the abstract quotient manifold $T_{[x]}(\mathcal{M}/\sim)$ that are necessary for implementing any iterative optimization algorithm on a quotient manifold, including the conjugate-gradient method.

- The matrix representations of the tangent space $T_x\mathcal{M}$ and the horizontal space $\mathcal{H}_x$ at $x \in \mathcal{M}$, including *projection operators* on these spaces which (Absil et al., 2008, Chapter 3)

	Two − factor decomposition $\mathbf{X} = \mathbf{G}\mathbf{H}^T$	Three − factor decomposition $\mathbf{X} = \mathbf{U}\mathbf{R}\mathbf{V}^T$
Tangent vectors in $T_x\mathcal{M}$	$\{(\xi_{\mathbf{G}}, \xi_{\mathbf{H}}) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{m\times r}\}$	$\{(\xi_{\mathbf{U}}, \xi_{\mathbf{R}}, \xi_{\mathbf{V}}) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{m\times r}:$ $\mathbf{U}^T\xi_{\mathbf{U}} + \xi_{\mathbf{U}}^T\mathbf{U} = 0,$ $\mathbf{V}^T\xi_{\mathbf{V}} + \xi_{\mathbf{V}}^T\mathbf{V} = 0\}$
Metric $g_x(\xi_x, \eta_x)$ for any $\xi_x, \eta_x \in T_x\mathcal{M}$	$\langle\eta_{\mathbf{G}}, \xi_{\mathbf{G}}(\mathbf{H}^T\mathbf{H})\rangle$ $+\langle\eta_{\mathbf{H}}, \xi_{\mathbf{H}}(\mathbf{G}^T\mathbf{G})\rangle$	$\langle\eta_{\mathbf{U}}, \xi_{\mathbf{U}}(\mathbf{R}\mathbf{R}^T)\rangle$ $+\langle\eta_{\mathbf{R}}, \xi_{\mathbf{R}}\rangle$ $+\langle\eta_{\mathbf{V}}, \xi_{\mathbf{V}}(\mathbf{R}^T\mathbf{R})\rangle$
Vertical tangent vectors in $\mathcal{V}_x$	$\{(-\mathbf{G}\mathbf{\Lambda}, \mathbf{H}\mathbf{\Lambda}^T):$ $\mathbf{\Lambda} \in \mathbb{R}^{r\times r}\}$	$\{(\mathbf{U}\mathbf{\Omega}_1, \mathbf{R}\mathbf{\Omega}_2 - \mathbf{\Omega}_1\mathbf{R}, \mathbf{V}\mathbf{\Omega}_2):$ $\mathbf{\Omega}_1, \mathbf{\Omega}_2 \in \mathbb{R}^{r\times r},$ $\mathbf{\Omega}_1^T = -\mathbf{\Omega}_1, \mathbf{\Omega}_2^T = -\mathbf{\Omega}_2\}$
Horizontal tangent vectors in $\mathcal{H}_x$	$\{(\zeta_{\mathbf{G}}, \zeta_{\mathbf{H}}) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{m\times r}:$ $\mathbf{G}^T\zeta_{\mathbf{G}}\mathbf{H}^T\mathbf{H} = \mathbf{G}^T\mathbf{G}\zeta_{\mathbf{H}}^T\mathbf{H}\}$	$\{(\zeta_{\mathbf{U}}, \zeta_{\mathbf{R}}, \zeta_{\mathbf{V}}) \in T_x\mathcal{M}:$ $\mathbf{R}\mathbf{R}^T\eta_{\mathbf{U}}^T\mathbf{U} + \eta_{\mathbf{R}}\mathbf{R}^T$ is symmetric, $\mathbf{R}^T\mathbf{R}\eta_{\mathbf{V}}^T\mathbf{V} - \mathbf{R}^T\eta_{\mathbf{R}}$ is symmetric$\}$

Table 4.2: With the proposed Riemannian metric (4.5) the quotient manifold has the structure of a Riemannian manifold. The Riemannian metric g_x makes the matrix representation of the abstract tangent space $T_x\mathcal{M}$ unique in the horizontal space $\mathcal{H}_x$.

- A way to "move" on the quotient manifold given a search direction $\xi_x \in T_x\mathcal{M}$. This is accomplished with a *retraction* mapping on $\mathcal{M}$ (Absil et al., 2008, Chapter 4). Care is taken so that we move from equivalence classes to equivalence classes and not only from point to point on $\mathcal{M}$. This makes sure that the mapping is valid on the quotient manifold $\mathcal{M}/\sim$.
- Specifically in the case of conjugate-gradients algorithm, a notion of comparing tangent vectors at different points, e.g., *gradients* of a function, on the manifold is also needed. This basic comparison is captured by a *vector transport* on a manifold (Absil et al., 2008, Section 8.1).

4.3.1 Tangent vector representation as horizontal lifts

The matrix representation of the tangent space $T_{[x]}(\mathcal{M}/\sim)$ of the abstract quotient manifold $\mathcal{M}/\sim$ is identified with a subspace of the tangent space of the total space $T_x\mathcal{M}$ that does not produce a displacement along the equivalence classes $[x] = \{y \in \mathcal{M} : y \sim x\}$. This is identified by decomposing the tangent space into two complementary subspaces as $T_x\mathcal{M} = \mathcal{H}_x \oplus \mathcal{V}_x$. This decomposition is with respect to the metric proposed in (4.5). Tangent vectors in the quotient manifold $\mathcal{M}/\sim$ are *horizontally lifted* to the horizontal space $\mathcal{H}_x$. The final matrix characterizations are shown in Table 4.2.

We work with the horizontal lifts to optimize any smooth cost function on the quotient space. Table 4.3 summarizes the concrete matrix operations involved in computing horizontal vectors. Starting from an arbitrary matrix (with appropriate dimensions), two linear projections are needed: the first projection Ψ_x is onto the tangent space of the total space $T_x\mathcal{M}$, while the second projection Π_x is onto the horizontal subspace $\mathcal{H}_x$.

The projection onto the tangent space $T_x\mathcal{M}$ is accomplished by extracting the component in the ambient space that is *normal* to the tangent space, i.e, the tangent space and the normal space together span the entire ambient space. Removing further the vertical component (the characterization of which is shown in Table 4.2) gives a horizontal vector. The Lyapunov equations involved in these projection operations, shown in Table 4.3, are solved efficiently and in closed form by diagonalizing $\mathbf{R}$ and performing similarity transforms on the variables. See Appendix A for the solution approach. Solving the Lyapunov equation costs $O(r^3)$. The computational cost of forming other matrix-matrix products is $O(nr^2 + mr^2 + r^3)$. Overall, the cost of using the projection operators is linear in the matrix dimensions n and m. This is critical for the computational efficiency of any iterative algorithm.

4.3.2 Retractions from the tangent space to the manifold

A retraction is a mapping that maps vectors in the horizontal space to points on $\mathcal{M}$ (Absil et al., 2008, Chapter 4)). It provides a natural way to iterate on the manifold along a search direction. Due to the product structure of the total space $\mathcal{M}$, a retraction $\mathcal{M}$is obtained by combining the retraction updates on $\mathbb{R}_*^{n\times r}$ (Absil et al., 2008, Example 4.1.5), $\mathrm{St}(r,n)$ (Absil et al., 2008, Example 4.1.3), and $\mathrm{GL}(r)$ (Absil et al., 2008, Chapter 4). The cartesian product of the retractions also defines a valid retraction on the quotient manifold $\mathcal{M}/\sim$ (Absil et al., 2008, Proposition 4.1.3). The retractions for the fixed-rank matrix factorizations are

$$\begin{array}{rl} \mathrm{Two-factor}: & R_x(\xi_x) = (\mathbf{G} + \xi_{\mathbf{G}}, \mathbf{H} + \xi_{\mathbf{H}}), \\ \mathrm{Three-factor}: & R_x(\xi_x) = (\mathrm{uf}(\mathbf{U} + \xi_{\mathbf{U}}), \mathbf{R} + \xi_{\mathbf{R}}, \mathrm{uf}(\mathbf{V} + \xi_{\mathbf{V}})), \end{array} \tag{4.6}$$

where $\xi_x \in \mathcal{H}_x$ is a search direction and $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of the polar decomposition of a full column rank matrix, i.e., $\mathrm{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1/2}$ which is computed efficiently by performing the singular value decomposition of $\mathbf{A}$. The computational cost of a retraction operation is $O(nr^2 + mr^2 + r^3)$. The retraction R_x (4.6) defines a valid retraction on the Riemannian quotient manifold $\mathcal{M}/\sim$ such that $R_{[x]}(\xi_{[x]}) := [R_x(\xi_x)]$, where ξ_x is the horizontal lift of an abstract tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}/\sim)$ in $\mathcal{H}_x$ and $[x]$ and $[R_x(\xi_x)]$ are the equivalence classes defined in Table 4.1.

4.3.3 Vector transport on the manifold

A vector transport $\mathcal{T} : T_x\mathcal{M} \times T_x\mathcal{M} \to T_{R_x(\eta_x)}\mathcal{M} : (\eta_x, \xi_x) \mapsto \mathcal{T}_{\eta_x}\xi_x$ on a manifold $\mathcal{M}$ is a smooth mapping that transports a tangent vector $\xi_x \in T_x\mathcal{M}$ at $x \in \mathcal{M}$ to a vector in the tangent space at $R_x(\eta_x)$ (Absil et al., 2008, Chapter 8). The Riemannian submersion structure of the quotient manifold $\mathcal{M}/\sim$ allows us to compute the matrix representation of the vector transport $\mathcal{T}_{\eta_{[x]}}\xi_{[x]}$ on the quotient manifold $\mathcal{M}/\sim$ with projections operators and the retraction mapping in the total space $\mathcal{M}$ (Absil et al., 2008, sections 8.1.3 and 8.1.4). In particular from the theory of Riemannian submersion, the horizontal lift of the vector transport $\mathcal{T}_{\eta_{[x]}}\xi_{[x]}$ is given as

$$\text{horizontal lift of } \mathcal{T}_{\eta_{[x]}}\xi_{[x]} = \Pi_{R_x(\eta_x)}(\mathcal{T}_{\eta_x}\xi_x),$$

	Two – factor decomposition $\mathbf{X} = \mathbf{G}\mathbf{H}^T$	Three – factor decomposition $\mathbf{X} = \mathbf{U}\mathbf{R}\mathbf{V}^T$
Matrix representation of the ambient space	$(\mathbf{Z_G}, \mathbf{Z_H}) \in \mathbb{R}^{n \times r} \times \mathbb{R}^{m \times r}$	$(\mathbf{Z_U}, \mathbf{Z_R}, \mathbf{Z_V}) \in \mathbb{R}^{n \times r} \times \mathbb{R}^{r \times r} \times \mathbb{R}^{m \times r}$
Normal space	$\emptyset$	$(\mathbf{U}\mathbf{N}_1, 0, \mathbf{V}\mathbf{N}_2)$: $\mathbf{N}_1, \mathbf{N}_2 \in \mathbb{R}^{r \times r}$, $\mathbf{N}_1\mathbf{R}\mathbf{R}^T$ is symmetric, $\mathbf{N}_2\mathbf{R}^T\mathbf{R}$ is symmetric
	Ψ_x, projection onto the tangent space	$\downarrow$
Projection of an ambient vector onto $T_x\mathcal{M}$	$\Psi_x(\mathbf{Z_G}, \mathbf{Z_H})$ $= (\mathbf{Z_G}, \mathbf{Z_H})$	$\Psi_x(\mathbf{Z_U}, \mathbf{Z_R}, \mathbf{Z_V})$ $= (\mathbf{Z_U} - \mathbf{U}\mathbf{B_U}(\mathbf{R}\mathbf{R}^T)^{-1}, \mathbf{Z_R},$ $\mathbf{Z_V} - \mathbf{V}\mathbf{B_V}(\mathbf{R}^T\mathbf{R})^{-1})$, where $\mathbf{B_U}$ and $\mathbf{B_V}$ are symmetric matrices of size $r \times r$ obtained by solving the Lyapunov equations $\mathbf{R}\mathbf{R}^T\mathbf{B_U} + \mathbf{B_U}\mathbf{R}\mathbf{R}^T =$ $2\mathbf{R}\mathbf{R}^T\mathrm{Sym}(\mathbf{U}^T\mathbf{Z_U})\mathbf{R}\mathbf{R}^T$ $\mathbf{R}^T\mathbf{R}\mathbf{B_V} + \mathbf{B_V}\mathbf{R}^T\mathbf{R} =$ $2\mathbf{R}^T\mathbf{R}\mathrm{Sym}(\mathbf{V}^T\mathbf{Z_V})\mathbf{R}^T\mathbf{R}$
	Π_x, projection onto the horizontal space	$\downarrow$
Projection of a tangent vector $\eta_x \in T_x\mathcal{M}$ onto $\mathcal{H}_x$	$\Pi_x(\eta_x)$ $= (\eta_\mathbf{G} + \mathbf{G}\mathbf{\Lambda}, \eta_\mathbf{H} - \mathbf{H}\mathbf{\Lambda}^T)$, where $\mathbf{\Lambda} = 0.5(\eta_\mathbf{H}^T\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}$ $-(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\eta_\mathbf{G})$	$\Pi_x(\eta_x)$ $= (\eta_\mathbf{U} - \mathbf{U}\mathbf{\Omega}_1, \eta_\mathbf{R} - (\mathbf{R}\mathbf{\Omega}_2 - \mathbf{\Omega}_1\mathbf{R}),$ $\eta_\mathbf{V} - \mathbf{V}\mathbf{\Omega}_2)$, where $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are the unique solutions to the coupled Lyapunov equation $\mathbf{R}\mathbf{\Omega}_2\mathbf{R}^T - \mathbf{R}\mathbf{R}^T\mathbf{\Omega}_1 - \mathbf{\Omega}_1\mathbf{R}\mathbf{R}^T =$ $\mathrm{Skew}(\mathbf{U}^T\xi_\mathbf{U}\mathbf{R}\mathbf{R}^T) + \mathrm{Skew}(\mathbf{R}\xi_\mathbf{R}^T)$ $\mathbf{R}^T\mathbf{\Omega}_1\mathbf{R} - \mathbf{R}^T\mathbf{R}\mathbf{\Omega}_2 - \mathbf{\Omega}_2\mathbf{R}^T\mathbf{R} =$ $\mathrm{Skew}(\mathbf{V}^T\xi_\mathbf{V}\mathbf{R}^T\mathbf{R}) + \mathrm{Skew}(\mathbf{R}^T\xi_\mathbf{R})$

TABLE 4.3: The matrix representations of the projection operations using Ψ_x and Π_x. Ψ_x projects a matrix in the Euclidean space onto the tangent space $T_x\mathcal{M}$ by removing the normal component. Π_x further extracts the horizontal component of a tangent vector ξ_x. Here the operators $\mathrm{Sym}(\cdot)$ and $\mathrm{Skew}(\cdot)$ extract the symmetric and skew-symmetric parts of a square matrix and are defined as $\mathrm{Sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^T)/2$ and $\mathrm{Skew}(\mathbf{A}) = (\mathbf{A} - \mathbf{A}^T)/2$ for any square matrix $\mathbf{A}$.

Algorithm 1 The Riemannian conjugate-gradient method for minimizing $f : \mathcal{M} \to \mathbb{R}$ on $\mathcal{M}/\sim$

```
Input: The Riemannian structure on $\mathcal{M}/\sim$ with the metric $g$ (4.5),
    initial iterate $x_0 \in \mathcal{M}$, and the search vector $\eta_0 = 0$.
Ouput: Sequence of iterates $\{x_i\}$.
Compute the Riemannian gradient $\xi_i = \mathrm{grad}_{x_i} f \in \mathcal{H}_{x_i}$.    ▷ (4.11)
Compute the conjugate search direction by Polak-Ribière (PR+) that takes a particular linear combination of the previous search vector with the current Riemannian gradient as
    $\eta_i = -\xi_i + \beta_i \Pi_{x_i}(\Psi_{x_i}(\eta_{i-1})), \eta_i \in \mathcal{H}_{x_i}$.    ▷ Section 4.4.2 and (4.13)
Check whether the search direction is a descent direction, i.e.,
    verify that $g_{x_i}(\eta_i, \xi_i) > 0$. If not, then $\eta_i = -\xi_i$    ▷ Reset
Determine an initial step-size $s_i$.    ▷ Section 4.4.3
Retract with backtracking line search starting from the step-size $s_i$ to arrive at a step-size $s_i/2^p$
    ($p \geq 0$ integer), and the next iterate is $x_{i+1} = R_{x_i}(\frac{s_i}{2^p}\eta_i)$.    ▷ (4.6)
Repeat until convergence.
```

where $\mathcal{T}_{\eta_x}\xi_x$ is the vector transport in the total space $\mathcal{M}$, η_x and ξ_x are horizontal lifts in $\mathcal{H}_x$ of $\xi_{[x]}$ and $\eta_{[x]}$ that belong to $T_{[x]}(\mathcal{M}/\sim)$, $\Pi_{R_x(\eta_x)}(\cdot)$ is the projection operator that extract the horizontal component of a tangent vector (defined in Table 4.3) at $R_x(\eta_x)$, and $R_x(\eta_x) \in \mathcal{M}$ is the retraction along $\eta_x \in \mathcal{H}_x$ defined in (4.6). Exploiting the Riemannian structure further, the vector transport $\mathcal{T}_{\eta_x}\xi_x$ in the total space $\mathcal{M}$ admits the expression

$$\mathcal{T}_{\eta_x}\xi_x = \Psi_{R_x(\eta_x)}(\xi_x),$$

where $\Psi_x(\cdot)$ is the projection operator defined in Table 4.3 that projects an ambient vector onto the tangent space $T_x\mathcal{M}$ and $R_x(\eta_x)$ is the retraction along $\eta_x \in \mathcal{H}_x$ defined in (4.6).

Finally, the horizontal lift of $\mathcal{T}_{\eta_{[x]}}\xi_{[x]}$ in the horizontal space $\mathcal{H}_x$ has the formula

$$\text{horizontal lift of } \mathcal{T}_{\eta_{[x]}}\xi_{[x]} = \Pi_{R_x(\eta_x)}(\Psi_{R_x(\eta_x)}(\xi_x)), \tag{4.7}$$

$\Pi_x(\cdot)$ and $\Psi_x(\cdot)$ are projection operations defined in Table 4.3, xi_x and η_x are horizontal lifts of $\xi_{[x]}$ and $\eta_{[x]}$, and $R_x(\cdot)$ is the retraction mapping (4.6). The computational cost of transporting a vector solely depends on projection and retraction operations which cost $O(nr^2 + mr^2 + r^3)$.

4.4 Algorithmic details

For the sake of illustration, we consider the conjugate-gradient method which can be easily implemented using notions developed in the previous section. Tables 4.1, 4.2, and 4.3; combined with retraction (4.6), and vector transport (4.7) operations give all the necessary ingredients for optimizing any smooth cost function on the Riemannian quotient manifold of fixed-rank matrix factorizations. For example, consider a *smooth* cost function $f : \mathcal{M} \to \mathbb{R} : x \mapsto f(x)$ and the optimization problem (with slight abuse of notations)

$$\begin{array}{ll} \min\limits_{x \in \mathcal{M}} & f(x) \\ \text{subject to} & [x] \in \mathcal{M}/\sim, \end{array} \tag{4.8}$$

where $[x] = \{y \in \mathcal{M} : y \sim x\}$. The characterizations for $\mathcal{M}$ and $\mathcal{M}/\sim$ are in Table 4.1. The function f has the following characterization for the low-rank matrix completion problem (4.1).

$$\begin{array}{l} \mathrm{Two-factor:} \\ f:\mathbb{R}_*^{n\times r}\times\mathbb{R}_*^{m\times r}\rightarrow\mathbb{R}:(\mathbf{G},\mathbf{H})\mapsto\frac{1}{|\Omega|}\|\mathcal{P}_\Omega(\mathbf{G}\mathbf{H}^T)-\mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 \\ \\ \mathrm{Three-factor:} \\ f:\mathrm{St}(r,n)\times\mathrm{GL}(r)\times\mathrm{St}(r,m)\rightarrow\mathbb{R}:(\mathbf{U},\mathbf{R},\mathbf{V})\mapsto\frac{1}{|\Omega|}\|\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T)-\mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2, \end{array} \tag{4.9}$$

where Ω is the set of known indices of the incomplete matrix $\mathbf{X}^\star$ and $\mathcal{P}_\Omega(\cdot)$ is the orthogonal sampling operator. Here $\mathbb{R}_*^{n\times r}$ is the set of full column rank matrices, $\mathrm{St}(r,n)$ is the set of matrices of size $n\times r$ with orthonormal columns, and $\mathrm{GL}(r)$ is the set of $r\times r$ non-singular matrices.

The skeletal version of the Riemannian conjugate-gradient method for the problem (4.8) is shown in Algorithm 1. The basic steps of the method include computing the *Riemannian gradient* of $\mathcal{M}/\sim$, computing the *conjugate search direction* that is a linear combination of the current gradient and β-*scaled* previous search direction, and performing backtracking linesearch to compute the subsequent iterate. Each of these steps is worked out in detail in this section for the cost function (4.9). The structure of f in (4.9) is also exploited to compute an *initial guess* for the step-size along a given search direction.

The convergence of the Riemannian conjugate-gradient algorithm to a critical point of (4.8) follows from the convergence analysis of Ring and Wirth (2012); Sato and Iwai (2013). Step 3 of Algorithm 1, in particular, ensures that the sequence $\{\eta_i\}$, $\eta_i\in\mathcal{H}_{x_i}$ is *gradient-related* (Absil et al., 2008, Definition 4.2.1). Consequently, Algorithm 1 converges to a critical point of (4.8) (Absil et al., 2008, Theorem 4.3.1). The rate of convergence analysis of Algorithm 1, on the other hand, is difficult to establish. Empirically, however, good performance is reported on all our examples.

4.4.1 The Riemannian gradient computation

The horizontal lift (matrix representation) of the Riemannian gradient $\mathrm{grad}_{[x]}f$ of f (4.9) on the quotient manifold $\mathcal{M}/\sim$ at $x\in\mathcal{M}$ is obtained by solving the *convex* quadratic programming problem, presented in (3.8),

$$\begin{array}{rcl} \text{horizontal lift of } \mathrm{grad}_{[x]}f & = & \mathrm{grad}_x f \\ & = & \underset{\zeta_x\in T_x\mathcal{M}}{\arg\min}\, f(x)-\langle f_x(x),\zeta_x\rangle+\frac{1}{2}g_x(\zeta_x,\zeta_x), \end{array} \tag{4.10}$$

where $\mathrm{grad}_x f$ is the gradient on the computational space $\mathcal{M}$, $f_x(x)$ is the first-order derivative of f, $T_x\mathcal{M}$ is the tangent space of $\mathcal{M}$ at x, $\langle\cdot,\cdot\rangle$ is the standard Euclidean derivative, and g_x is the metric (4.5). The equality between the horizontal lift $\mathrm{grad}_{[x]}f$ and $\mathrm{grad}_x f$ in (4.10) is the standard result of the Riemannian submersion theory (Absil et al., 2008, sections 3.6.1 and 3.6.2).

The problem (4.10) admits a closed form solution as we shown below. The tangent space $T_x\mathcal{M}$ characterization follows from Table 4.2.

$$
\begin{aligned}
&\text{Two}-\text{factor}:\\
&f_x(x) = (\mathbf{S}\mathbf{H}^T, \mathbf{S}^T\mathbf{G}),\\
&\qquad\qquad \text{where } \mathbf{S} = 2(\mathcal{P}_\Omega(\mathbf{G}\mathbf{H}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star))/|\Omega|.\\
&\mathrm{grad}_x f \;\; = \underset{\zeta_x \in T_x\mathcal{M}}{\arg\min} \quad f(x) - \langle f_x(x), \zeta_x\rangle + \tfrac{1}{2} g_x(\zeta_x, \zeta_x)\\
&\qquad\qquad = \underset{\substack{\zeta_{\mathbf{G}} \in \mathbb{R}^{n\times r}\\ \zeta_{\mathbf{H}} \in \mathbb{R}^{m\times r}}}{\arg\min} \;\; \begin{array}{l} -\langle \mathbf{S}\mathbf{H}^T, \zeta_{\mathbf{G}}\rangle - \langle \mathbf{S}^T\mathbf{G}, \zeta_{\mathbf{H}}\rangle \\ +\tfrac{1}{2}(\langle \zeta_{\mathbf{G}}, \zeta_{\mathbf{G}}\mathbf{H}^T\mathbf{H}\rangle + \langle \zeta_{\mathbf{H}}, \zeta_{\mathbf{H}}\mathbf{G}^T\mathbf{G}\rangle)\end{array}\\
&\qquad\qquad = (\mathbf{S}\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}, \mathbf{S}^T\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}).
\end{aligned}
\tag{4.11a}
$$

$$
\begin{aligned}
&\text{Three}-\text{factor}:\\
&f_x(x) = (\mathbf{S}\mathbf{V}\mathbf{R}^T, \mathbf{U}^T\mathbf{S}\mathbf{V}, \mathbf{S}^T\mathbf{U}\mathbf{R}),\\
&\qquad\qquad \text{where } \mathbf{S} = 2(\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star))/|\Omega|.\\
&\mathrm{grad}_x f \;\; = \underset{\zeta_x \in T_x\mathcal{M}}{\arg\min} \quad f(x) - \langle f_x(x), \zeta_x\rangle + \tfrac{1}{2} g_x(\zeta_x, \zeta_x)\\
&\qquad\qquad = \underset{\substack{\zeta_{\mathbf{U}} \in \mathbb{R}^{n\times r},\\ \zeta_{\mathbf{R}} \in \mathbb{R}^{r\times r},\\ \zeta_{\mathbf{V}} \in \mathbb{R}^{m\times r}}}{\arg\min} \;\; \begin{array}{l} -\langle \mathbf{S}\mathbf{V}\mathbf{R}^T, \zeta_{\mathbf{U}}\rangle - \langle \mathbf{U}^T\mathbf{S}\mathbf{V}, \zeta_{\mathbf{R}}\rangle - \langle \mathbf{S}^T\mathbf{U}\mathbf{R}, \zeta_{\mathbf{V}}\rangle \\ +\tfrac{1}{2}(\langle \zeta_{\mathbf{U}}, \zeta_{\mathbf{U}}\mathbf{R}\mathbf{R}^T\rangle + \langle \zeta_{\mathbf{R}}, \zeta_{\mathbf{U}}\rangle + \langle \zeta_{\mathbf{V}}, \zeta_{\mathbf{V}}\mathbf{R}^T\mathbf{R}\rangle)\end{array}\\
&\qquad\qquad\quad \text{subject to} \quad \mathbf{U}^T\zeta_{\mathbf{U}} + \zeta_{\mathbf{U}}^T\mathbf{U} = 0, \mathbf{V}^T\zeta_{\mathbf{V}} + \zeta_{\mathbf{V}}^T\mathbf{V} = 0\\
&\qquad\qquad = (\mathbf{S}\mathbf{V}\mathbf{R}^T(\mathbf{R}\mathbf{R}^T)^{-1} - \mathbf{U}\mathbf{B}_{\mathbf{U}}(\mathbf{R}\mathbf{R}^T)^{-1}, \mathbf{U}^T\mathbf{S}\mathbf{V},\\
&\qquad\qquad\quad \mathbf{S}^T\mathbf{U}\mathbf{R}(\mathbf{R}^T\mathbf{R})^{-1} - \mathbf{V}\mathbf{B}_{\mathbf{V}}(\mathbf{R}^T\mathbf{R})^{-1}),\\
&\qquad\qquad\quad \text{where } \mathbf{B}_{\mathbf{U}} \text{ and } \mathbf{B}_{\mathbf{V}} \text{ are solutions to the Lyapunov equations}\\
&\qquad\qquad\quad \mathbf{R}\mathbf{R}^T\mathbf{B}_{\mathbf{U}} + \mathbf{B}_{\mathbf{U}}\mathbf{R}\mathbf{R}^T = 2\mathrm{Sym}(\mathbf{R}\mathbf{R}^T\mathbf{U}^T\mathbf{S}\mathbf{V}\mathbf{R}^T)\\
&\qquad\qquad\quad \mathbf{R}^T\mathbf{R}\mathbf{B}_{\mathbf{V}} + \mathbf{B}_{\mathbf{V}}\mathbf{R}^T\mathbf{R} = 2\mathrm{Sym}(\mathbf{R}^T\mathbf{R}\mathbf{V}^T\mathbf{S}^T\mathbf{U}\mathbf{R}).
\end{aligned}
\tag{4.11b}
$$

Here Sym$(\cdot)$ extracts the symmetric part of a square matrix, i.e., $\mathrm{Sym}(\mathbf{A}) = (\mathbf{A} + \mathbf{A}^T)/2$. The total numerical cost of computing the Riemannian gradient (4.11) is $O(|\Omega|r + nr^2 + mr^2 + r^3)$.

4.4.2 The conjugate direction computation

As mentioned earlier, at the i^{th} iteration the conjugate direction η_i in Algorithm 1 is obtained by a linearly combining the Riemannian gradient $\mathrm{grad}_{x_i} f$ with previous search direction η_{i-1}. However, it should be stressed that $\mathrm{grad}_{x_i} f \in \mathcal{H}_{x_i}$ and $\eta_{i-1} \in \mathcal{H}_{x_{i-1}}$ belong to different horizontal spaces on $\mathcal{M}$. This is tackled by invoking the concept of vector transport on manifold (Section 4.3.3) and η_{i-1} is transported to from x_{i-1} to x_i by using the vector transport operation $\mathcal{T}(\cdot)$ in Section 4.3.3 that, by definition, produces a vector in the horizontal space $\mathcal{H}_{x_i}$ at x_i. Finally using the formula (4.7), the update proposed is

$$\eta_i = -\mathrm{grad}_{x_i} f + \beta_i \Pi_{x_i}(\Psi_{x_i}(\eta_{i-1})), \quad \eta_i \in \mathcal{H}_{x_i}, \tag{4.12}$$

where β_i is the scaling parameter at the i^{th} iteration that ensures approximate conjugacy of search directions. Out of many choices for β, a popular choice is the Polak-Ribière (PR$_+$) (Absil et al., 2008, Section 8.3) formula with automatic restart property (Nocedal and Wright, 2006, Chapter 5)

$$\beta_i = \max(\frac{g_{x_i}(\mathrm{grad}_{x_i} f, \mathrm{grad}_{x_i} f - \Pi_{x_i}(\Psi_{x_i}(\mathrm{grad}_{x_{i-1}} f)))}{g_{x_{i-1}}(\mathrm{grad}_{x_{i-1}} f, \mathrm{grad}_{x_{i-1}} f)}, 0), \tag{4.13}$$

where $\mathrm{grad}_{x_i} f$ at x_i is the Riemannian gradient at i^{th} iteration, $\Pi_{x_i}(\Psi_{x_i}(\mathrm{grad}_{x_{i-1}} f))$ is the matrix representation of the vector transport of the Riemannian gradient $\mathrm{grad}_{x_{i-1}}$ at x_{i-1} to x_i, and $g_x(\cdot,\cdot)$ is the Riemannian metric (4.5).

It should be noted that if β is fixed to 0, then the conjugate direction is simply the *negative* Riemannian gradient direction, turning Algorithm 1 into the standard steepest-descent algorithm.

The computational cost of computing the conjugate direction at each iteration is equal to cost of computing the vector transport and the metric,, the sum total of which costs $O(nr^2 + mr^2 + r^3)$.

4.4.3 Initial guess for the step-size

Computing a good step-size guess has a significant effect on the performance of a nonlinear conjugate-gradient algorithm (Nocedal and Wright, 2006, Chapter 5). The extra cost of computing an approximate step-size is *usually* compensated by a faster rate of convergence. To this end, we exploit the least-squares nature of the matrix completion cost function to compute a *linearized* step-size guess efficiently (Vandereycken, 2013). Given a search direction $\eta_x \in \mathcal{H}_x$, the optimization problems that we solve at each iteration are

$$\begin{array}{rl} \mathrm{Two-factor:} & s_* = \underset{s\in\mathbb{R}}{\arg\min} \|\mathcal{P}_\Omega((\mathbf{G} - s\eta_{\mathbf{G}})(\mathbf{H} - s\eta_{\mathbf{H}})^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 \\ \mathrm{Three-factor:} & s_* = \underset{s\in\mathbb{R}}{\arg\min} \|\mathcal{P}_\Omega((\mathbf{U} - s\eta_{\mathbf{U}})(\mathbf{R} - s\eta_{\mathbf{R}})(\mathbf{V} - s\eta_{\mathbf{V}})^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2. \end{array} \tag{4.14}$$

For the two-factor factorization in (4.14), it should be noted that the cost function of the optimization problem is a degree 4 polynomial in s. The minima are, therefore, the roots of its first derivative which is a degree 3 polynomial. Efficient algorithms exist (including closed-form expressions) for finding the roots of a degree 3 polynomial and hence, finding the optimal s_* is numerically straightforward. Total numerical cost is $O(|\Omega|r)$.

Similarly for the three-factor factorization in (4.14), the cost function is a degree 6 polynomial in s and thus, the global minimum s_* can be obtained numerically (and computationally efficiently) by finding the roots of its degree 5 derivative polynomial. However, this can be further relaxed by considering a degree 2 polynomial *approximation*, i.e.,

$$s_*^{\mathrm{accel}} = \underset{s\in\mathbb{R}}{\arg\min} \|\mathcal{P}_\Omega(\mathbf{URV}^T + s\eta_{\mathbf{U}}\mathbf{RV}^T + s\mathbf{U}\eta_{\mathbf{R}}\mathbf{V}^T + s\mathbf{UR}\eta_{\mathbf{V}}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2 \tag{4.15}$$

that has a closed form solution. Computing s_*^{accel} (4.15) for the three-factor factorization is about *three times* faster than computing s_* in (4.14) with a numerical cost of $O(|\Omega|r)$.

4.5 Updating rank

In many problems a good rank of the solution is either not known a priori or the notion of numerical rank is too vague to define it precisely, e.g., matrices with exponential decay of singular values. In such instances, it makes sense to traverse through a number of ranks, and not just one, in a systematic manner while ensuring that the cost function of (4.1) is minimized. One way is to use fixed-rank optimization (by fixing the rank) in conjunction with a rank-update strategy. Such schemes have been quite popular in solving large-scale semidefinite programming problems (Burer and Monteiro, 2003; Journée et al., 2010). To this end, we propose the meta scheme shown in Table (4.4) for low-rank matrix completion that alternates between fixed-rank optimization (with a Riemannian conjugate-gradient algorithm) and *rank-one* updates. The rank-one update is based on the idea of moving along the *dominant* rank-one projection of the negative gradient of the mean square error $\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ in the space $\mathbb{R}^{n\times m}$. The scheme in Table 4.4 ensures a *monotonic* decrease of the cost function $(\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2/)/|\Omega|$ in (4.1).

Given an $n\times m$ rank-r matrix $\mathbf{X}$, the rank-one update corresponds to computing (fixed-rank factorizations of) a rank-$(r+1)$ matrix $\mathbf{X}_+$ such that

$$\begin{array}{rcl}
\mathrm{Two-factor:} & & \\
\mathbf{X}_+ & = & \mathbf{X} - \sigma u v^T \\
\mathbf{G}_+\mathbf{H}_+^T & = & \mathbf{G}\mathbf{H}^T - \sigma u v^T \\
\mathrm{Three-factor:} & & \\
\mathbf{X}_+ & = & \mathbf{X} - \sigma u v^T \\
\mathbf{U}_+\mathbf{R}_+\mathbf{V}_+^T & = & \mathbf{U}\mathbf{R}\mathbf{V}^T - \sigma u v^T,
\end{array} \tag{4.16}$$

where $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are the dominant unit-norm left and right singular vectors of $2(\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star))/|\Omega|$ and $\sigma > 0$ is the dominant singular value, $(\mathbf{G}_+, \mathbf{H}_+) \in \mathbb{R}_*^{n\times(r+1)} \times \mathbb{R}_*^{m\times(r+1)}$, and $(\mathbf{U}_+, \mathbf{R}_+, \mathbf{V}_+) \in \mathrm{St}(r+1, n) \times \mathrm{GL}(r) \times \mathrm{St}(r+1, m)$. Here $\mathbb{R}_*^{n\times r}$ is the set of full column rank matrices, $\mathrm{St}(r, n)$ is the set of matrices of size $n \times r$ with orthonormal columns, and $\mathrm{GL}(r)$ is the set of $r\times r$ non-singular matrices. The rank-one updating (4.16) can be done numerically efficiently (Brand, 2006). The total computational cost is $O(|\Omega| + nr^2 + mr^2 + r^3)$.

The rank-update procedure shown in Table 4.4 is continued till a satisfied mean square error $\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ is identified. Not surprisingly, this procedure of alternating between fixed-rank optimization and rank-updates is computationally more intensive as we traverse through a number of ranks one by one, minimizing $\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ at each rank. On the other hand, it leads to better accuracy (smaller errors $\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ at lower ranks) in matrix completion problems which are *approximately* low-rank, i.e., their singular values decay exponentially but are not zero. This scenario is discussed in Case (d) of Section 4.6. Secondly, the rank-update procedure also plays a pivotal role in trace norm regularization problems, the convex alternative to fixing the rank, where it helps in computing a series of convex solutions efficiently. This is discussed in Chapter 5.

Given	• Initial rank r_0, e.g., $r_0 = 1$ and initial iterate $\mathbf{X}_0 \in \mathbb{R}^{n \times m}_{r_0}$.
Scheme	We alternate between the following two steps until a threshold is reached.
Step i)	Compute a stationary point $\mathbf{X} \in \mathbb{R}^{n \times m}_{r}$ of the fixed-rank optimization problem (4.1) with the Riemannian conjugate-gradient algorithm proposed in Section 4.4 initialized from $\mathbf{X}_0$.
Step ii)	Update the rank r to $r+1$ and initialize $\mathbf{X}_0 = \mathbf{X} - \sigma u v^T$, where $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are the dominant left and right singular vectors of $2(\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star))/\lvert\Omega\rvert$ and $\sigma > 0$ is the dominant singular value, and $t > 0$ is an appropriate step-size computed by backtracking.

Table 4.4: A meta scheme for minimizing $\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ with a smaller rank solution $\mathbf{X} \in \mathbb{R}^{n \times m}$.

4.6 Numerical comparisons

Our Riemannian conjugate-gradient algorithms based the two-factor and three-factor factorizations are referred to as R2MC and R3MC, respectively. The Matlab implementations are available from the website http://www.montefiore.ulg.ac.be/~mishra and the generic implementations of the two fixed-rank geometries are provided in the Manopt optimization toolbox (Boumal et al., 2014) which has additional algorithmic implementations, e.g., the trust-regions.

In this section, we show numerical comparisons of R2MC and R3MC with a number of state-of-the-art algorithms. We show that our proposed algorithms connect closely to a number of competing methods. In addition to this, we bring out a few conceptual differences between the competing algorithms and ours. Finally, the numerical comparisons suggest that our geometric algorithms compete favorably with state-of-the-art algorithms.

State-of-the-art algorithms considered for comparisons are the following. In ScGrassMC, LRGeom, and Polar Factorization, we use their conjugate-gradient implementations together with *linearized* step-size guesses. R3MC uses the accelerated step-size computation (4.15) while R2MC uses the step-size computation (4.14).

1. RTRMC (Boumal and Absil, 2011): It considers the decomposition of a rank-r matrix $\mathbf{X}$ into $\mathbf{X} = \mathbf{U}\mathbf{Y}^T$, where $\mathbf{U} \in \mathrm{St}(r, n)$ ($n \times r$ matrices with orthonormal columns) and $\mathbf{Y} \in \mathbb{R}^{m \times r}_*$ (full column-rank $m \times r$ matrices). The fixed-rank optimization problem (4.1) is reformulated as an optimization problem on the *Grassmann manifold* $\mathrm{Gr}(r, n)$ of dimension $nr - r^2$ by eliminating the variable $\mathbf{Y}$. This is done by exploiting the least-squares structure of the matrix-completion problem (Dai et al., 2012). Consequently, the resulting algorithm is efficient in situations where $n \ll m$ as optimization is on a smaller search space of dimension $nr - r^2$ instead of $nr + mr - r^2$. Optimization on the Grassmann manifold is performed in the Riemannian optimization framework which enables to develop both first-order and second-order algorithms. For numerical comparisons in this section, we use the second-order Riemannian trust-region code of RTRMC with default parameters as suggested by Boumal and Absil (2012).

2. LMaFit (Wen et al., 2012): It relies on the factorization $\mathbf{X} = \mathbf{G}\mathbf{H}^T$ (4.2) of a fixed-rank matrix $\mathbf{X}$ to alternatively update the matrices $\mathbf{X}$, $\mathbf{G}$ and $\mathbf{H}$ while better minimizing the mean square error $\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$. The LMaFit algorithm has been a popular benchmark owing to simpler updates of iterates and tuned step-size updates in turn leading to a superior time per iteration complexity. More details are in Section 4.6.2.
3. ScGrassMC (Ngo and Saad, 2012): It relies on the factorization $\mathbf{X} = \mathbf{U}\mathbf{R}\mathbf{V}^T$ of the form (4.2). It alternatively updates $\mathbf{R}$ and $(\mathbf{U}, \mathbf{V})$ (Keshavan et al., 2010). Fixing $\mathbf{U} \in \mathrm{St}(r, n)$ and $\mathbf{V} \in \mathrm{St}(r, m)$, the factor $\mathbf{R}$ is updated by solving the least-squares problem

$$\min_{\mathbf{R}\in\mathbb{R}^{r\times r}} \|\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$$

that has a closed form solution. However, a practical implementation of ScGrassMC computes $\mathbf{R}$ approximately. Fixing $\mathbf{R}$, the factors $(\mathbf{U}, \mathbf{V})$ are updated on the *bi-Grassmann* search space $\mathrm{Gr}(r, n) \times \mathrm{Gr}(r, m)$ using an iteration for the problem

$$\min_{(\mathbf{U},\mathbf{V})\in\mathrm{Gr}(r,n)\times\mathrm{Gr}(r,m)} \|\mathcal{P}_\Omega(\mathbf{U}\mathbf{R}\mathbf{V}^T) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2, \tag{4.17}$$

where $\mathrm{Gr}(r, n)$ is the set of r-dimensional subspaces in $\mathbb{R}^n$. ScGrassMC specifically considers *scaled* conjugate-gradient algorithm for (4.17), where the scaling is interpreted as an adaptive preconditioner that well-conditions the level sets of the cost function. More details are in Section 4.6.2.
4. LRGeom (Vandereycken, 2013): Many works, including ours, view the set of fixed-rank matrices as the product space of well-studied matrix manifolds. A different viewpoint is that of an *embedded submanifold*, i.e, the search space $\mathbb{R}_r^{n\times m}$ has the structure of a Riemannian submanifold in the space $\mathbb{R}^{n\times m}$ (Vandereycken, 2013, Proposition 2.1). The metric imposed on $\mathbb{R}_r^{n\times m}$ is the restriction of the standard Euclidean inner product from $\mathbb{R}^{n\times m}$. The recent papers by Shalit et al. (2010); Vandereycken (2013) investigate the search space in detail and develop the notions of minimizing a smooth cost function. While conceptually the iterates of an iterative algorithm move on the embedded submanifold, numerically the implementation is done *efficiently* using fixed-rank factorizations, e.g, the two-factor factorization is used by Shalit et al. (2010) and the compact singular value decomposition by Vandereycken (2013). For numerical comparisons, we consider the conjugate-gradient implementation of Vandereycken (2013).
5. Polar Factorization (Meyer et al., 2011a): It considers a Riemannian quotient geometry based on a three-factor factorization model similar to ours in (4.2) but with the additional constraint that the factor $\mathbf{R}$ is *symmetric and positive definite*, i.e, $\mathbf{R} \succ 0$. The Riemannian metric defined on the space is the geometric product metric, i.e., it is a summation of the natural metric on $\mathrm{St}(r, n)$ (Absil et al., 2008, Chapter 3) and the *bi-invariant* metric on the positive cone $\mathbf{R} \succ 0$ (Bhatia, 2007, Section 6.1). The imposed geometry thus generalizes the Riemannian geometry for the symmetric positive *semidefinite* matrices proposed by Bonnabel and Sepulchre (2009). For numerical comparisons, we consider the conjugate-gradient implementation of (Meyer et al., 2011a).

The choice of these algorithms as state-of-the-art rests on recent publications (Boumal and Absil, 2011; Meyer et al., 2011a; Ngo and Saad, 2012; Vandereycken, 2013; Wen et al., 2012) that solve the low-rank matrix completion problem with a fixed-rank constraint, e.g., (4.1). Other problems formulations like matrix completion with nuclear norm regularization (Cai et al., 2010; Toh and Yun, 2010) have not been discussed here as the main motivation of the present chapter is to look specifically look at algorithms that exploit the search space of fixed-rank matrices. It should, however, be stated that all the considered algorithms can be exploited, though it is not trivial, to have online, stochastic, and parallel implementations (Balzano et al., 2010; Recht and Ré, 2013).

Implementation details and stopping criteria

All simulations in this section are performed in Matlab on a 2.53 GHz Intel Core i5 machine with 4 GB RAM. We use Matlab codes for all the considered algorithms. Each simulation shows the behavior of the cost function $\frac{1}{|\Omega|}\|\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star)\|_F^2$ as a function of time taken or iterations used. For each example considered here, an $n \times m$ random matrix of rank r is generated as proposed by Cai et al. (2010). Two matrices $\mathbf{A} \in \mathbb{R}^{n\times r}$ and $\mathbf{B} \in \mathbb{R}^{m\times r}$ are generated according to a Gaussian distribution with zero mean and unit standard deviation. The matrix product $\mathbf{AB}^T$ gives a random matrix of rank r. A fraction of the entries are randomly removed with uniform probability. It should be noted that the dimension of the space of $n \times m$ matrices of rank r is $(n + m - r)r$ and the number of entries known is a *multiple* of this dimension. This multiple or ratio is called the *over-sampling ratio* or simply, *over-sampling* (OS). The over-sampling ratio (OS) determines the number of entries that are known. A OS $= 6$ means that $6(n+m-r)r$ of randomly and uniformly selected entries are known a priori out of a total of nm entries.

The maximum number of iterations of all except RTRMC is set to 500. For RTRMC, the maximum number of outer iterations is set to 200 (we expect a better rate of convergence) and the number of inner iterations (for the trust-regions subproblem) is set to 100. Finally, all the algorithms are terminated when the cost function value is below 10^{-20}. The algorithms are initialized randomly.

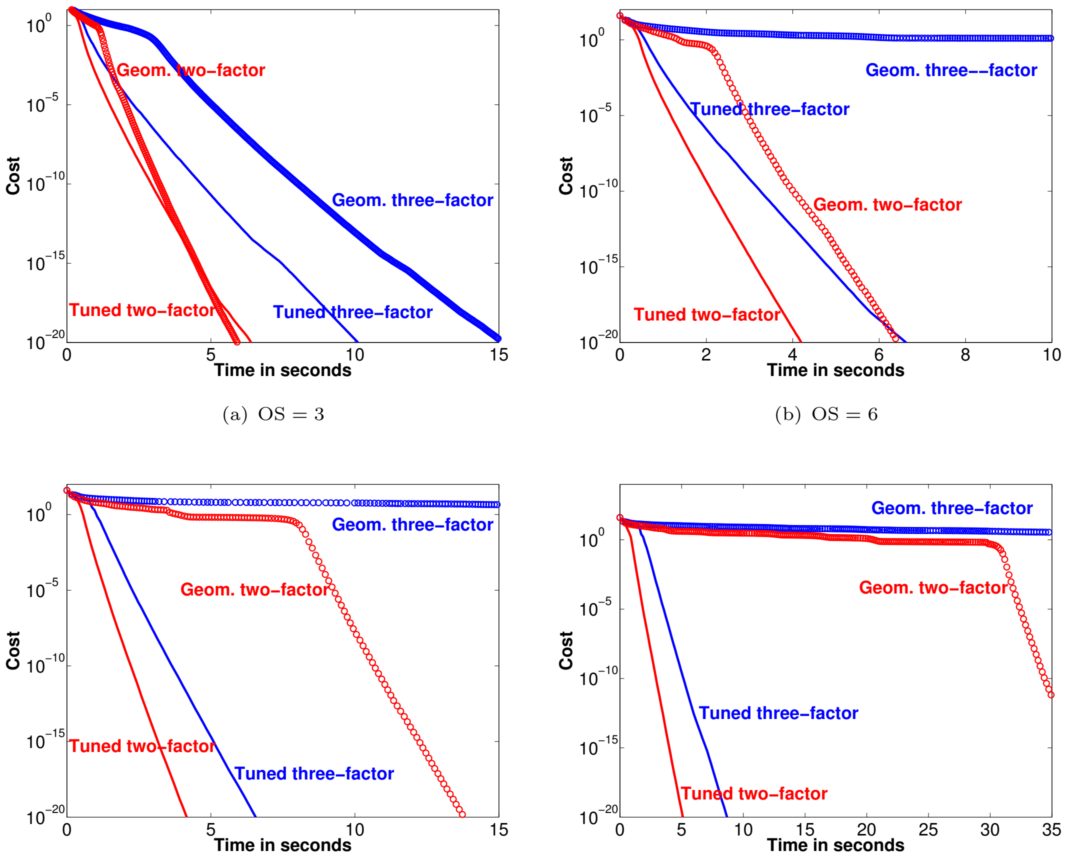
4.6.1 Comparison of Riemannian metrics

In contrast to the tuned Riemannian metrics proposed in (4.5), an alternative choice is to use the *conventional* approach of exploiting the product structure of $\mathcal{M}$ to propose Riemannian metrics. A valid Riemannian metric on $\mathcal{M}$ is defined by combining the individual natural metrics for $\mathbb{R}_*^{n\times r}$ (Absil et al., 2008, Example 3.6.4), $\mathrm{St}(r, n)$ (Absil et al., 2008, Example 3.6.2), and the *right-invariant* metric on $\mathrm{GL}(r)$ (Vandereycken et al., 2013, (3.10)). The resulting metric on $\mathcal{M}$ is called the *geometric product metric* that is a summation of the Riemannian metrics on the product spaces (Lee, 2003, Example 13.2);

$$\text{Two} - \text{factor}: \quad g_x(\xi_x, \eta_x) = \langle \xi_{\mathbf{G}}, \eta_{\mathbf{G}}(\mathbf{G}^T\mathbf{G})^{-1}\rangle + \langle \xi_{\mathbf{H}}, \eta_{\mathbf{H}}(\mathbf{H}^T\mathbf{H})^{-1}\rangle \tag{4.18a}$$

$$\text{Three} - \text{factor}: \quad g_x(\xi_x, \eta_x) = \langle \xi_{\mathbf{U}}, \eta_{\mathbf{U}}\rangle + \langle \xi_{\mathbf{R}}\mathbf{R}^{-1}, \eta_{\mathbf{R}}\mathbf{R}^{-1}\rangle + \langle \xi_{\mathbf{V}}, \zeta_{\mathbf{V}}\rangle \tag{4.18b}$$

where $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product and $(\mathbf{G}, \mathbf{H})$ and $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ are the matrix representations of the two-factor and three-factor fixed-rank factorizations, respectively. Similarly, $(\xi_{\mathbf{G}}, \xi_{\mathbf{H}})$ and $(\xi_{\mathbf{U}}, \xi_{\mathbf{R}}, \xi_{\mathbf{V}})$ are the matrix representations of the tangent vector $\xi_x \in T_x\mathcal{M}$ the characterization is shown in Table 4.2. Likewise for $\zeta_x \in T_x\mathcal{M}$. Consequently, the manifold $\mathcal{M}$ (the characterization is shown in Table 4.1) with the geometric product metrics (4.18) has also the structure of a Riemannian submersion.



(a) OS = 3

(b) OS = 6

(c) OS = 12

(d) OS = 20

FIGURE 4.1: Comparison of between different choices of Riemannian metrics. $n = m = 1000$ and $r = 20$. The gradient descent algorithms based on the tuned Riemannian metric (4.5) consistently perform better than their counterparts based on the conventional Riemannian metric (4.18). The matrix scaling in the Riemannian gradient obtained from one is the inverse of the scaling obtained from the other, as shown in (4.19).

It should be stressed that the matrix scaling obtained by (4.5) is quite different from the one obtained from the metrics (4.18). To see this consider the two-factor fixed-rank factorization case. The matrix representations of the Riemannian gradient $\mathrm{grad}_x f$ for the cost function (4.9) has the matrix expressions

$$\begin{array}{ll} \text{Metric } (4.5a): & \mathrm{grad}_x f = (\mathbf{SH}(\mathbf{H}^T\mathbf{H})^{-1}, \mathbf{S}^T\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}) \\ \text{Metric } (4.18a): & \mathrm{grad}_x f = (\mathbf{SH}(\mathbf{G}^T\mathbf{G}), \mathbf{S}^T\mathbf{G}(\mathbf{H}^T\mathbf{H})), \end{array} \tag{4.19}$$

where $\mathbf{S} = 2(\mathcal{P}_\Omega(\mathbf{X}) - \mathcal{P}_\Omega(\mathbf{X}^\star))/|\Omega|$ and $x = (\mathbf{G}, \mathbf{H}) \in \mathcal{M}$. Moving to a different matrix representative of the same equivalence class $[\mathbf{G}, \mathbf{H}] := \{(\mathbf{GM}^{-1}, \mathbf{HM}^T) : \mathbf{M} \in \mathrm{GL}(r)\}$ with the property $\mathbf{G}^T\mathbf{G} = \mathbf{H}^T\mathbf{H}$ (there exists a continuum of such matrix representations in $[\mathbf{G}, \mathbf{H}]$), we see that in (4.19) the matrix scalings produced by (4.5) and (4.18) are *inverse* to each other. This also suggests that the geometric product metric (4.18) is expected to perform *poorly* in matrix completion instances where a large number of entries are already known where our proposed metric (4.5) captures a part of the second-order information of (4.1).

We illustrate here the empirical evidence that the metric (4.5) tailored to the cost function performs better than the conventional geometric product metric (4.18). To this end, we consider the simplest implementation of a *gradient descent* algorithm for the matrix completion problem (4.1) by setting the parameter $\beta = 0$ in Algorithm 1. A 1000×1000 matrix of rank 20 is generated. The entries are removed with different over-sampling ratios (OS). As the OS is increased, the behavior of steepest-descent algorithms based on the metrics (4.18) and (4.5) separate out in Figure 4.1, thereby confirming the observation made in the previous paragraph.

4.6.2 Connection with LMaFit and ScGrassMC

In this section we connect our Riemannian conjugate-gradient algorithms R2MC and R3MC to LMaFit (Wen et al., 2012) and ScGrassMC (Ngo and Saad, 2012), respectively.

As mentioned earlier, the Gauss-Seidel algorithm LMaFit of Wen et al. (2012) is an alternating minimization scheme on the search space $(\mathbf{X}, \mathbf{G}, \mathbf{H}) \in \mathbb{R}_r^{n\times m} \times \mathbb{R}^{n\times r} \times \mathbb{R}^{m\times r}$ for the optimization problem

$$\begin{array}{ll} \min\limits_{\substack{\mathbf{G}\in\mathbb{R}^{n\times r}\\ \mathbf{H}\in\mathbb{R}^{m\times r}\\ \mathbf{X}\in\mathbb{R}_r^{n\times m}}} & \|\mathbf{G}\mathbf{H}^T - \mathbf{X}\|_F^2 \\ \text{subject to} & \mathbf{X}_{ij} = \mathbf{X}^\star_{ij}, \quad (i,j)\in\Omega \end{array} \tag{4.20}$$

where $\mathbb{R}_r^{n\times m}$ is the set of rank-r matrices of size $n \times m$ and Ω is the set of indices for which the entries in $\mathbf{X}^\star$ are known. Each variable in $(\mathbf{X}, \mathbf{G}, \mathbf{H})$ is updated *sequentially* by fixing the others, as is common in an alternating minimization algorithm. The specific update proposed by Wen et al. (2012, Equation (2.8)) to compute a new iterate $(\mathbf{X}_+, \mathbf{G}_+, \mathbf{H}_+)$ are

$$\begin{aligned} \mathbf{G}_+(\omega) &= (1-\omega)\mathbf{G} + \omega\mathbf{X}\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1} \\ \mathbf{H}_+(\omega) &= (1-\omega)\mathbf{H} + \omega\mathbf{X}^T\mathbf{G}_+(\omega)(\mathbf{G}_+(\omega)^T\mathbf{G}_+(\omega))^{-1} \\ \mathbf{X}_+(\omega) &= \mathbf{G}_+(\omega)\mathbf{H}_+(\omega)^T + \mathcal{P}_\Omega(\mathbf{X}^\star - \mathbf{G}_+(\omega)\mathbf{H}_+(\omega)^T), \end{aligned} \tag{4.21}$$

where $\mathcal{P}_\Omega(\cdot)$ is the orthogonal sampling operator defined in (4.1) and the weight $\omega \geq 1$ is updated using a strategy that is similar to adjusting the radius in the trust-region method (Nocedal and Wright, 2006, Chapter 4). After eliminating the variable $\mathbf{X}(\omega)$ from (4.21), the *equivalent updates* are

$$\begin{aligned} \mathbf{G}_+(\omega) &= (1-\omega)\mathbf{G} + \omega(\mathbf{G}\mathbf{H}^T + \mathcal{P}_\Omega(\mathbf{X}^\star - \mathbf{G}\mathbf{H}^T))\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1} \\ \mathbf{H}_+(\omega) &= (1-\omega)\mathbf{H} + \omega(\mathbf{H}\mathbf{G}^T + \mathcal{P}_\Omega(\mathbf{X}^{\star T} - \mathbf{H}\mathbf{G}^T))\mathbf{G}_+(\omega)(\mathbf{G}_+(\omega)^T\mathbf{G}_+(\omega))^{-1}. \end{aligned} \tag{4.22}$$

If now instead of the sequential update of the variables $(\mathbf{G}, \mathbf{H})$ as in (4.22) a *simultaneous* update of the variables $(\mathbf{G}, \mathbf{H})$ is allowed, then the update (4.22) becomes

$$\begin{aligned} \mathbf{G}_+(\omega) &= \mathbf{G} - \omega\mathbf{S}\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1} \\ \mathbf{H}_+(\omega) &= \mathbf{H} - \omega\mathbf{S}^T\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}, \end{aligned} \tag{4.23}$$

where $\mathbf{S} = \mathcal{P}_\Omega(\mathbf{G}\mathbf{H}^T - \mathbf{X}^\star)$.

It should be noted that the update (4.23) of the variables $(\mathbf{G}, \mathbf{H})$ is *precisely* the same update (up to a normalization constant) that results from a Riemannian steepest-descent algorithm update for the two-factor factorization with respect to the metric (4.5), i.e., R2MC with $\beta = 0$, with the step-size ω, i.e.,

$$\begin{array}{rcl} x_+ & = & x - \omega \mathrm{grad}_x f \\ \Rightarrow \quad (\mathbf{G}_+, \mathbf{H}_+) & = & (\mathbf{G}, \mathbf{H}) - \omega(\mathbf{SH}(\mathbf{H}^T\mathbf{H})^{-1}, \mathbf{S}^T\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}), \end{array}$$

where $x = (\mathbf{G}, \mathbf{H}) \in \mathcal{M}$ (the characterization is shown in Table 4.1) and $\mathbf{S} = 2\mathcal{P}_\Omega(\mathbf{G}\mathbf{H}^T - \mathbf{X}^\star)/|\Omega|$. This leads us to the conclusion that the LMaFit algorithm is the *sequential version* of our geometric algorithm R2MC with $\beta = 0$ (steepest-descent). In other words, the Riemannian metric (4.5) confers a geometric foundation to LMaFit and at the same time allows to retain its good properties.

ScGrassMC alternatively updates $\mathbf{R} \in \mathbb{R}^{r \times r}$ and $(\mathbf{U}, \mathbf{V}) \in \mathrm{Gr}(r, n) \times \mathrm{Gr}(r, m)$ using the three-factor factorization (4.2). The connection with ScGrassMC is apparent from the fact that it considers a matrix scaling of the Riemannian gradient for the variables $(\mathbf{U}, \mathbf{V})$ on $\mathrm{Gr}(r, n) \times \mathrm{Gr}(r, m)$ by $((\mathbf{R}\mathbf{R}^T)^{-1}, (\mathbf{R}^T\mathbf{R})^{-1})$. It should be emphasized that this is the same matrix scaling that is obtained by our metric (4.5) in the computation of the Riemannian gradient (4.11). Additionally, we have linear projections to respect orthogonality of the factors $(\mathbf{U}, \mathbf{V})$. On the other hand, the difference with respect to ScGrassMC is on two fronts.

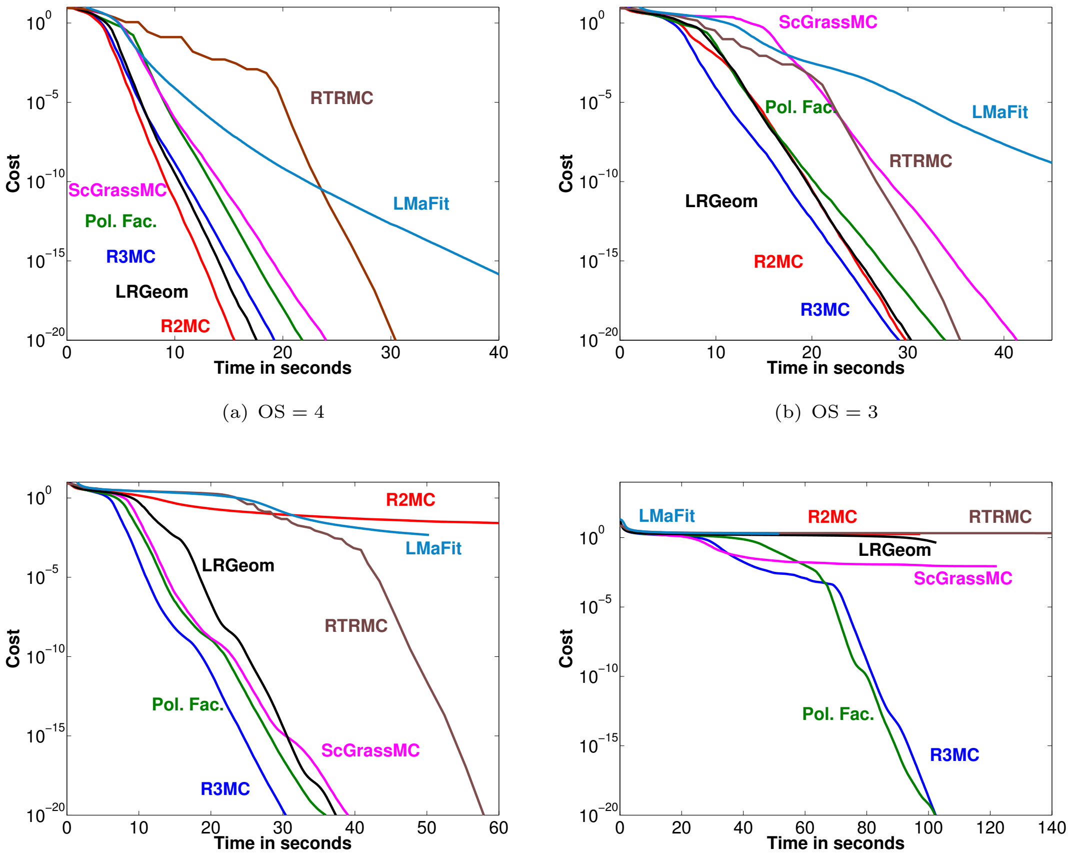
- First, R3MC performs a simultaneous update of the variables $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ while ScGrassMC alternates between updating $(\mathbf{U}, \mathbf{V})$ and $\mathbf{R}$.
- Second, while *preconditioning* in ScGrassMC is motivated by Ngo and Saad (2012) as a way to accelerate the algorithm of Keshavan et al. (2010), we view preconditioning as an outcome of a particular Riemannian metric that can be employed in an arbitrary unconstrained optimization algorithm. This is a fundamentally different view from the one of Ngo and Saad (2012).

4.6.3 Comparisons with state-of-the-art

In this section, we provide a comparative review of the algorithms across different scenarios of low-rank matrix completion instances including on the popular MovieLens-1M dataset `http://grouplens.org/datasets/movielens/`.

Case (a): influence of over-sampling

We consider a moderate size matrix of size 10000×10000 of rank 10, generated randomly. Four instances of incomplete matrices with different over-sampling ratios, i.e., with different proportion of known entries, have been considered. For larger values of OS, most of the algorithms perform similarly and show a nice behavior; and both R2MC and R3MC compete effectively with state-of-the-art. With smaller OS ratios, the algorithms, however, perform differently. In fact in Figure 4.2, for the case of OS = 2.1 only R3MC and Polar Factorization algorithms converged. It should be added that second-order algorithms like RTRMC exhibit a *modestly* better behavior with *non-random* initialization, e.g., by taking the dominant r singular value decomposition of the sparse matrix $\mathcal{P}_\Omega(\mathbf{X}^\star)$. However, it does not alter the main observation, i.e., the algorithms that exploit separation of *scaling* and *subspace* information of a matrix, e.g., SVD-based factorizations, exhibit better behavior.



(a) OS = 4

(b) OS = 3

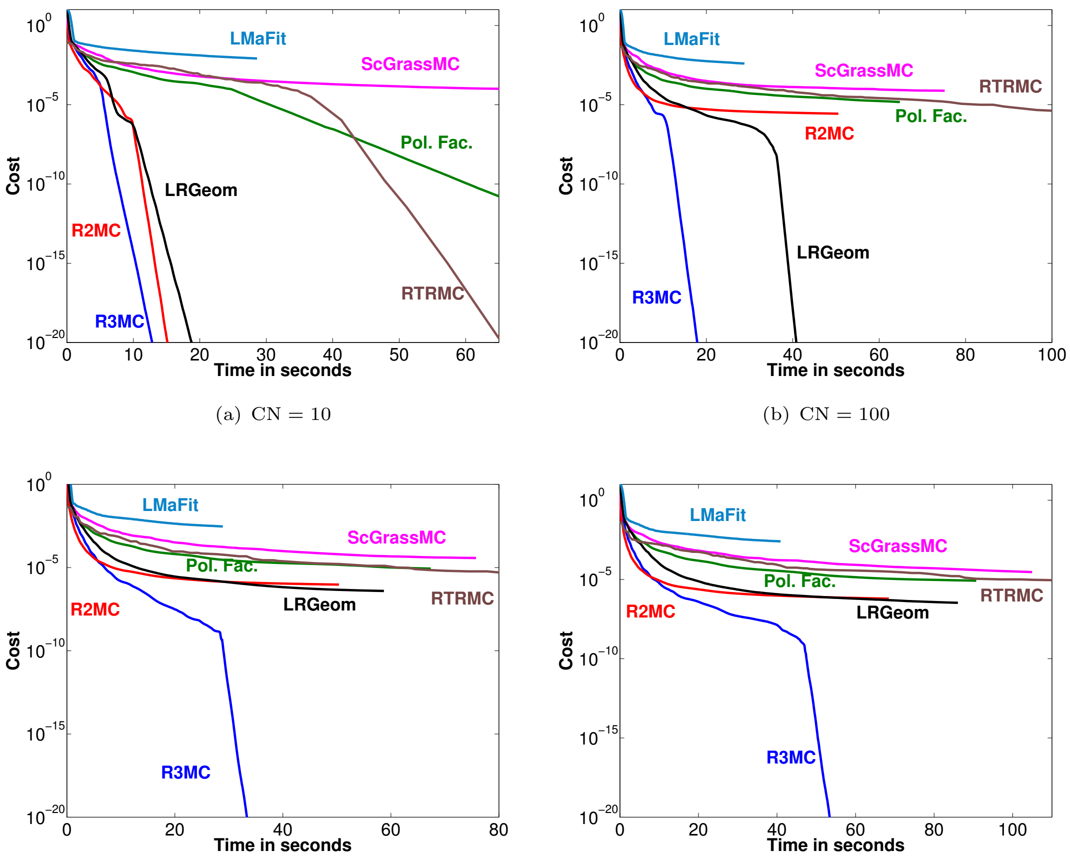
(c) OS = 2.5

(d) OS = 2.1

Figure 4.2: Case (a). For higher OS ratios, the performance behavior of all algorithms is similar. However under low sampling (smaller OS ratios), the Riemannian algorithms based on three-factor matrix factorization, including Polar Factorization, that separated scaling and subspace information perform better. R3MC is particularly efficient in a number of instances.

Case (b): influence of conditioning

We consider matrices of size 5000 × 5000 of rank 10 and impose an exponential decay of singular values. The ratio of the largest to the lowest singular value is known as the condition number (CN) of the matrix. At rank 10 the singular values with condition number 100 is obtained using the Matlab command `logspace(-2,0,10)`. The over-sampling ratio for these instances is 3. Matrix completion instances become challenging as the CN of matrices is increased. In Figure 4.3 we consider four different ill-conditioned matrices with different condition numbers. For low condition numbers, both R2MC and R3MC perform than others. For CN = 100, only R3MC, LRGeom, and RTRMC converged with RTRMC taking a much longer time. In instances with higher condition numbers, R3MC outperforms both LRGeom and RTRMC. It should be noted that, for lower accuracies (e.g., the cost function less than 10^{-6}) both R2MC and R3MC perform better than others.



(a) CN = 10

(b) CN = 100

(c) CN = 300

(d) CN = 500

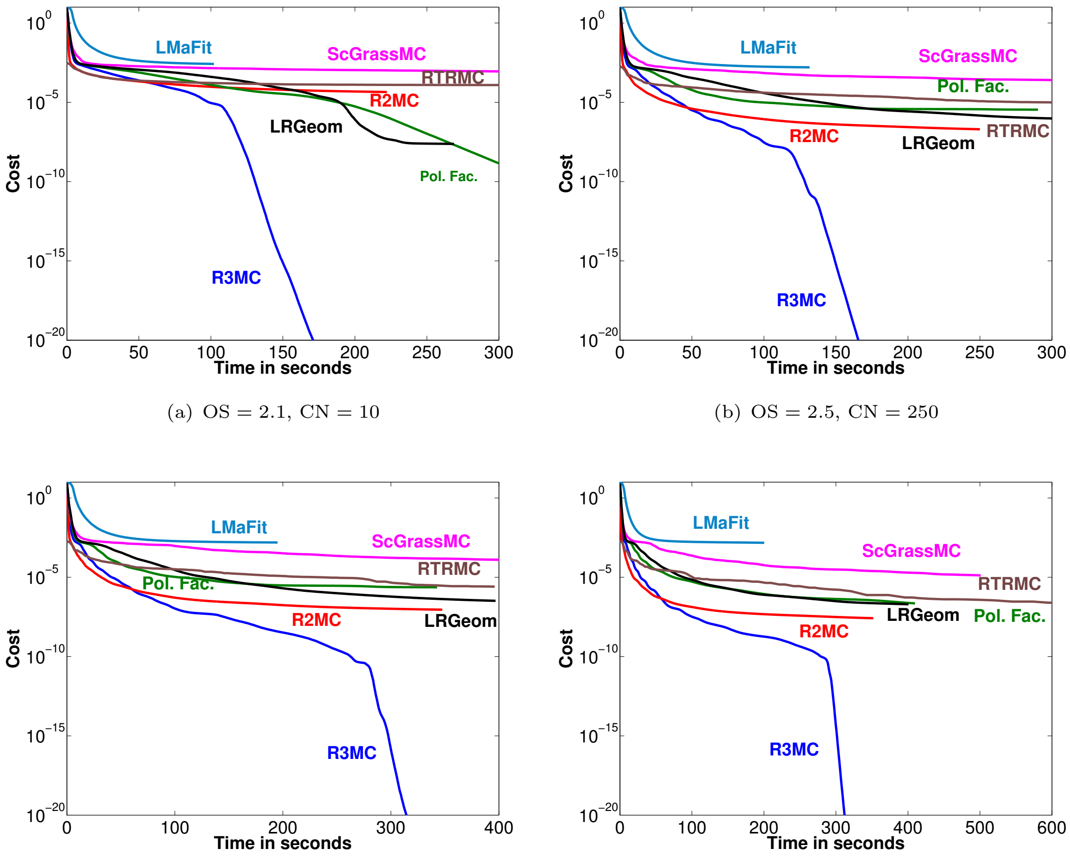
FIGURE 4.3: Case (b). R3MC is robust to different instances of ill-conditioned data. When solutions of lower accuracies are required, both R2MC and R3MC perform better than others.

Case (c): influence of low sampling + ill-conditioning

In this test, we look at problem instances that result from both scarcely sampled and ill-conditioned data. The test requires completing relatively large matrices of size 25000 × 25000 of rank 10 with different condition numbers and OS ratios. In Figure 4.4 R3MC outperforms all other algorithms. For lower accuracies, however, R2MC performs slightly better than R3MC. Both compete favorably to others.

Case (d): ill-conditioning + rank-one updates

We create a random matrix of size 5000 × 5000 of rank 20 with exponentially decaying singular values so that the condition number is 10^{10} and over-sampling ratio 2 (computed for rank 10). In Figure 4.5 we show the mean square error obtained on a set of entries Γ that is different from the set Ω (on which we minimize the cost function). First, in Figure 4.5(a) we compare the algorithms for rank 10 directly where R3MC shows a significantly better performance than others. The performance of RTRMC improves modestly with regularization. Second, in Figure 4.5(b) we use R3MC with the rank-one updating procedure in Section 4.5 starting from rank 1. It results in a better recovery at rank 10 and almost complete recovery



(a) OS = 2.1, CN = 10

(b) OS = 2.5, CN = 250

(c) OS = 3, CN = 300

(d) OS = 4, CN = 500

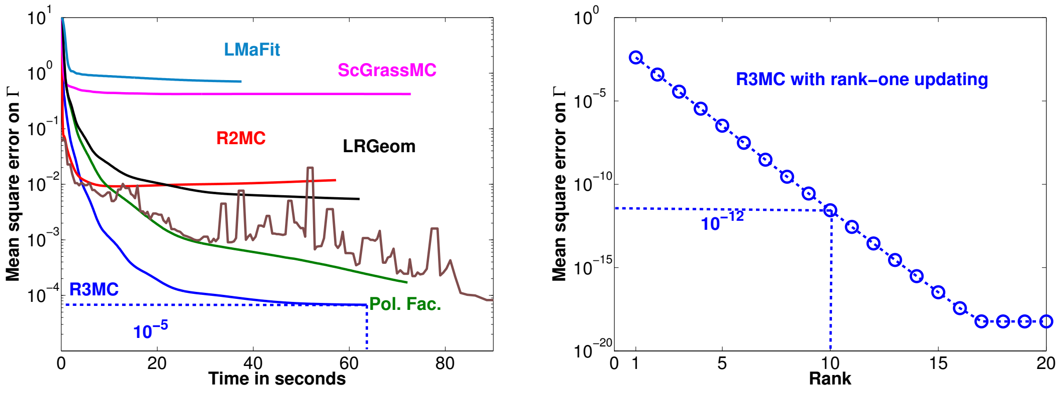
Figure 4.4: Case (c). R3MC especially exhibits a robust behavior in scenarios that combine scarcely sampled and ill-conditioned data.

at rank 17. This also suggests that fixed-rank algorithms including ours perform better with the rank updating procedure for very ill-conditioned data.

Case (e): rectangular matrices

Here we are particularly interested in instances with $n \ll m$, i.e., *rectangular* matrices. For these instances, most simulations suggest that the algorithm RTRMC of Boumal and Absil (2011) performs numerically very efficiently than others. This is not surprising as the underlying geometry of RTRMC exploits the fact that the least-square formulation (4.1) is solvable by fixing one of the *fixed-rank* factors. Consequently, the resulting problem becomes an optimization problem on the Grassmann manifold $\mathrm{Gr}(r, n)$ in the lower dimension n.

To adapt algorithms like R2MC and R3MC (including RTRMC) to rectangular matrices under the standard assumption of the Gaussian distribution of the known entries, we propose to deal with smaller size matrices with fewer columns. Instead of considering the entire matrix $\mathbf{X}^{\star}$ of size $n \times m$, we consider a *truncated* sub-matrix of size $n \times p$, shown in Figure 4.6. The sub-matrix matrix consists of all the rows of $\mathbf{X}^{\star}$ but contains only p (out of m) columns. The columns are randomly chosen. A simple analysis shows



(a) Direct rank 10 recovery

(b) Recovery with rank-one updates

FIGURE 4.5: Case (d). In general fixed-rank algorithms along with the rank-one updating (Section 4.5) show a much better performance for highly ill-conditioned data.

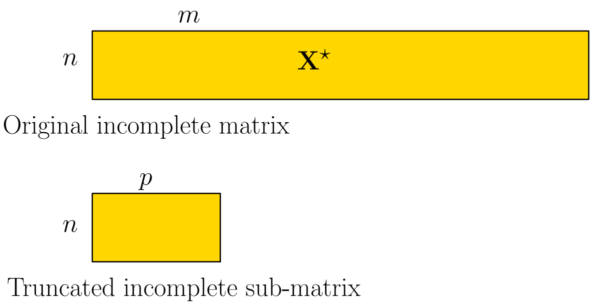


FIGURE 4.6: Case (e). In matrix completion instances with $n \ll m$, we propose to construct a sub-matrix that consists of only $p > n$ randomly chosen columns from the original incomplete matrix. In scenarios where an exact rank-r matrix completion for the full problem is expected, the smaller truncated problem gives concrete information about the rank-r *left subspace* spanning the columns. In other scenarios, the rank-r left subspace of the smaller truncated problem provides a good starting point to deal with the full problem.

that if the over-sampling ratio of the original problem is OS, then the OS for the truncated problem, i.e., matrix completion of the truncated sub-matrix, is $\mathrm{OS}_{\mathrm{truncated}} = \mathrm{OS}\alpha/(1+\alpha)$, where $\alpha = p/n$.

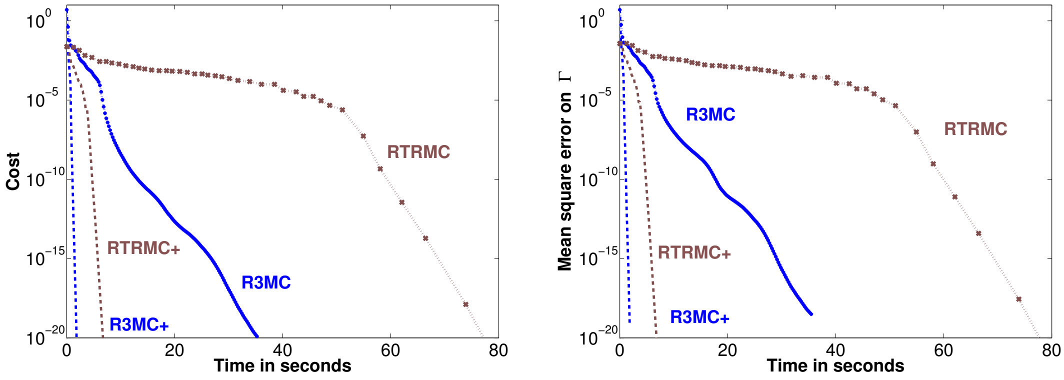
Although the dimension of the truncated problem $(nr + \alpha nr - r^2)$ is much smaller than that of the full problem $(nr + mr - r^2)$, the truncated problem is much more *challenging* to solve for smaller values α. However by tuning $\alpha > 1$, i.e., $p > n$ it is possible to have a competitive trade-off between difficulty and computational efficiency. It should, however, be noted that both the incomplete matrices (original and the truncated version) share the same *left subspace* under the assumption that an exact matrix reconstruction of the original problem is indeed possible. A different point of view is that if columns of the original incomplete matrix $\mathbf{X}^\star$ belong to a *time-invariant* left subspace, then the same subspace also appears in the truncated sub-matrix. This basic assumption has led to subspace tracking algorithms being successfully applied to low-rank matrix completion problems (Balzano et al., 2010; Dai et al., 2011). With the basic assumption of time-invariance the left subspace, we show our meta scheme in Table 4.5. While the algorithm of Balzano et al. (2010) *tracks* the r-dimensional subspace column-by-column (in a theoretical setting m tends to infinity but in a practical setting m is usually a large number), we propose

Rank-r matrix completion of an $n \times m$ $\mathbf{X}^\star$ with $n \ll m$.

1. Construct a sub-matrix of dimensions $n \times p$ $(> n)$ by picking only p columns out of m *randomly*.
2. Compute the rank-r left subspace $\mathbf{U} \in \mathrm{St}(r, n)$ (with orthonormal columns) that *best* solves the truncated sub-matrix completion problem
3. Once the left subspace $\mathbf{U} \in \mathrm{St}(r, n)$ is identified from the truncated sub-matrix, the weighting factor, e.g., the matrix $\mathbf{Y} \in \mathbb{R}_*^{n \times r}$ of the factorization $\mathbf{X} = \mathbf{U}\mathbf{Y}^T$ of the full problem is obtained by solving a least-squares problem by fixing $\mathbf{U}$ (Boumal and Absil, 2011). $(\mathbf{U}, \mathbf{Y})$ provides a good initialization to algorithms like R2MC and RTRMC for the full problem. Similarly, an additional QR factorization of $\mathbf{Y}$ gives an initialization to the factors $\mathbf{R} \in \mathrm{GL}(r)$ and $\mathbf{V} \in \mathrm{St}(r, m)$, such that $\mathbf{Y} = \mathbf{V}\mathbf{R}^T$. The resulting $(\mathbf{U}, \mathbf{R}, \mathbf{V})$ provides a good initialization to algorithms like R3MC for the full problem.

Table 4.5: A meta scheme for rectangular matrix completion problems under the standard assumption of the Gaussian distribution of the known entries.

to compute the left subspace of the smaller sub-matrix as a way to *initialize* algorithms for the full problem.



(a) Means square error

(b) Mean square test error

Figure 4.7: Case (e). We consider a random rank-5 matrix completion instance of a moderately ill-conditioned with CN = 10 rectangular matrix of size 1000×50000 and over-sampling ratio OS = 5. The truncated matrix completion problem is obtained by choosing $2 \times 1000 = 2000$ columns out of 50000. Consequently, the over-sampling ratio of the truncated problem is reduced to $\approx 5\alpha/(1+\alpha) = (2/3)5 = 3.3$. The plots show that both R3MC and RTRMC that solve the truncated problem first to initialize the full problem (appended by +) are significantly faster than their full problem counterparts. Under moderate ill-conditioning of the data, R3MC shows a better performance than RTRMC. In rectangular problems with high ill-conditioned data, the performance of R3MC amd RTRMC are similar in a number of situations.

To demonstrate the idea, we consider a rank 5 matrix of size 1000×50000. The over-sampling ratio is 5. An incomplete truncated matrix of size 1000×2000 is formed by picking 2000 columns randomly out of 50000, i.e., $\alpha = 2$ and $\mathrm{OS}_{\text{truncated}} = \mathrm{OS}\alpha/(1+\alpha) = 5(2/3) = 3.3$. Figure 4.7 shows that both R3MC and RTRMC with the meta-scheme above are significantly faster than the ones which deal directly with the full incomplete matrix. To show the efficacy of the scheme to recover the unknown entries, we show the mean square error obtained on a set of entries Γ that is different from the set Ω in Figure 4.7(b).

Rank r	R3MC	R2MC	Pol. Fac.	ScGrassMC	LRGeom	LMaFit	RTRMC
3	0.7713	0.7771	0.7710	0.7967	0.7723	0.7762	0.7858
4	0.7677	0.7758	0.7675	0.7730	0.7689	0.7727	0.8022
5	0.7666	0.7781	0.7850	0.8280	0.7660	0.8224	0.8314
6	**0.7634**	0.7893	0.7651	0.7910	0.7698	0.8194	0.8802
7	0.7684	0.7996	0.7980	0.8368	0.7810	0.8074 (max. iters.)	0.8241
With rank updating strategy (optimal rank)	0.7370 (9)	0.7434 (8)	0.7435 (20 max. rank)	**0.7323** (10)	0.7381 (9)	0.7435 (9)	Did-not give better results

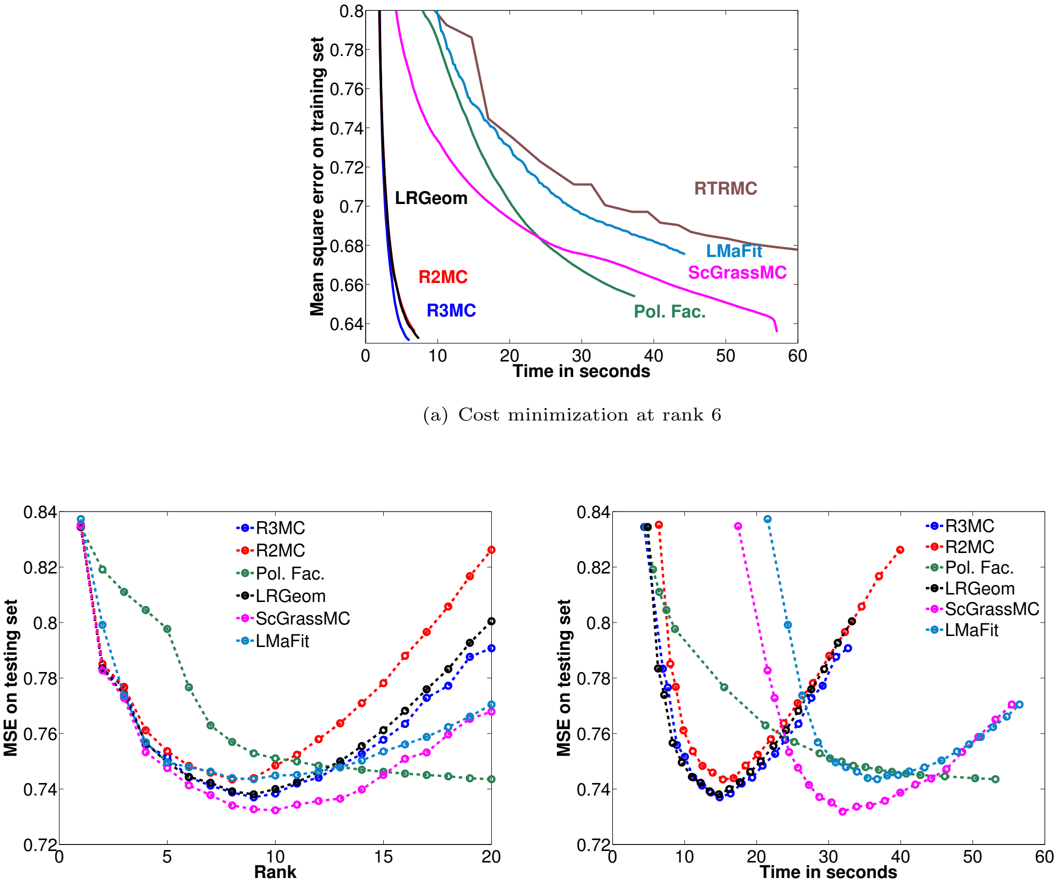
Table 4.6: MSEs obtained on the test set of the MovieLens-1M dataset.

Case (f): MovieLens dataset

As a final test, we compare all the considered algorithms on the popular MovieLens-1M dataset downloaded from the website http://grouplens.org/datasets/movielens/. The dataset has a million ratings corresponding to 6040 users and 3952 movies. We perform 10 random 80/10/10 - train/validation/test partitions of the rating data. The algorithms are run one each of these partitions and the results are averaged. Minimization of the cost function is on the training set. The algorithms are run for a maximum 1000 iterations (200 for RTRMC) and are stopped before if the mean square error (MSE) on the validation set (not the test set) starts to increase. Finally, the results are reported on the testing set. In RTRMC we also set the parameter λ to 10^{-6} to avoid an error due to non-uniqueness of the least-squares solution that it considers at each iteration.

Table 4.6 shows the MSEs on the testing set of ratings with standard deviations $\pm 10^{-5}$ for fixed-rank problems (the rank updating procedure is not used) at different ranks with random initialization. The best score of 0.7634 is obtained by R3MC at rank 6. The timing performance of different algorithms at rank 6 are shown in Figure 4.8(a) where R3MC, R2MC, and LRGeom are *considerably* faster than others. Similar plots are also observed at other ranks.

We also run the algorithms with the rank-updating procedure in Section 4.5 to simultaneously find a "good" rank. We traverse through all ranks from rank 1 to 20. Rank is updated only when the error on the validation set starts to increase. RTRMC with rank-one updating did not give better results than the one without the procedure and is omitted. Fig 4.8(b) shows all the MSEs on the testing set across ranks that are traversed with the rank-updating procedure. The last row of Table 4.6 compiles the *best* MSEs on the test set where the optimal ranks are shown in brackets. The best test score of 0.7323 is obtained by ScGrassMC at rank 10 followed by the score 0.7370 of R3MC at rank 9. As for the timing performance, R3MC is *twice* faster than ScGrassMC as shown in Figure 4.8(c) which also validates the observation in Figure 4.8(a) that R3MC, R2MC, and LRGeom are, in general, faster than others. Overall, employing



(a) Cost minimization at rank 6

(b) MSE on the testing set with the rank-updating procedure in Section 4.5

(c) Timing performance with rank-updating

FIGURE 4.8: Case (f), MovieLens-1M dataset. *Above*: R2MC, R3MC, and LRGeom are faster than others on this dataset. *Below*: with respect to accuracy, algorithms with the rank-updating procedure (Section 4.5) give better scores.

the rank-updating strategy in Section 4.5 on the MovieLens-1M dataset improves the test scores for all the considered algorithms.

Remarks

The case studies (a) to (f) are *challenging* instances of low-rank matrix completion because they combine ill-conditioning and low sampling in the data. Even though the above case studies are not fully exhaustive, they show a general trend of the performance of algorithms. The conclusions drawn from each case study are based on a number of runs. Each figure, however, shows a *typical* run. Similarly, even though we have shown convergence of the algorithms to very high accuracies, the conclusions drawn are equally valid for lower accuracies. In all the examples, R3MC and R2MC have shown faster convergence. R3MC has particularly shown a more robust performance.

4.7 Chapter summary

We have discussed two efficient Riemannian conjugate-gradient algorithms namely, R2MC and R3MC, for the low-rank matrix completion problem. Concrete matrix formulas have been summarized in tables. The algorithms stem from two specific Riemannian quotient geometries, by virtue of specially chosen Riemannian metrics on the set of two fixed-rank matrix factorizations. Various numerical comparisons suggest a competitive performance of our proposed algorithms. On a theoretical level, we have shown that the versatile nature of the Riemannian optimization framework not only lends itself excellently to the search space of fixed-rank matrices, but it is also possible to customize it to the least-squares nature of the cost function at hand.

The results of this chapter have been presented in (Mishra and Sepulchre, 2014a; Mishra et al., 2012).

Chapter 5

Solving large-scale convex problems with low-rank optimization

The present chapter focuses on the convex program

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} \quad F(\mathbf{X}) + \lambda\|\mathbf{X}\|_*, \tag{5.1}$$

where $F:\mathbb{R}^{n\times m}\to\mathbb{R}$ is a smooth *convex* function, $\|\mathbf{X}\|_*$ is the *trace norm* (also known as nuclear norm) which is the sum of the singular values of $\mathbf{X}$ (Cai et al., 2010; Fazel, 2002; Recht et al., 2010), and $\lambda > 0$ is the regularization parameter. Convex programs of this type have attracted much attention in recent years as efficient *convex relaxations* of intractable rank minimization problems (Fazel, 2002). The rank of the optimal minimizer of (5.1) as a function of the regularization parameter λ, i.e., $\mathbf{X}^*(\lambda)$, decreases to zero as the regularization parameter grows unbounded (Bach, 2008). As a consequence, generating efficiently a regularization path $\{\mathbf{X}^*(\lambda_i)\}$ for $i = \{1,\dots,N\}$, for a whole range of values of λ_i-minimizers, is a convenient convex proxy to obtain sub-optimal low-rank minimizers of F.

The chapter is organized as follows. In Section 5.2 the problem of fixed-rank optimization is related to the trace norm minimization problem. Section 5.3 proposes a Riemannian second-order geometry for the fixed-rank problem with a detailed numerical complexity analysis. An algorithm for (5.1) that alternates between fixed-rank optimization and rank-one updates is proposed in Section 5.4. A novel predictor-corrector approach to generate a regularization path of (5.1) for a grid of values of λ is discussed in Section 5.5. For the sake of illustration and empirical comparison with state-of-the-art algorithms we consider two particular applications, low-rank matrix completion (Candès and Plan, 2009) and multivariate linear regression (Yuan et al., 2007) in Section 5.6. In both cases, we obtain iterative algorithms with a numerical complexity that is linear in the number of observations and with favorable convergence and precision properties.

5.1 Motivation

Motivated by machine learning and statistical large-scale regression problems (Ma et al., 2011; Recht et al., 2010; Toh and Yun, 2010; Vounou et al., 2010; Yuan et al., 2007), we are interested in very low-rank solutions ($p < 10^2$) of very high-dimensional problems ($n > 10^6$). To this end, we propose an algorithm that guarantees *locally* second-order convergence to the solutions of (5.1) while ensuring a tight control on the data storage requirements (storage is linear in n and m) and on the numerical complexity of each iteration. In addition, we show an efficient computation of a regularization path of solutions to (5.1) by exploiting the problem structure and the geometry of rank constraints.

The proposed algorithm is based on the fixed-rank factorization (2.2.1.2) of a rank-p matrix, similar to the *compact* singular value decomposition (SVD), $\mathbf{X} = \mathbf{UBV}^T$. Like in SVD, $\mathbf{U} \in \mathrm{St}(p, n)$ (with orthonormal columns) and $\mathbf{V} \in \mathrm{St}(p, m)$ that span row and column spaces of $\mathbf{X}$, respectively. In contrast, the $p \times p$ scaling factor $\mathbf{B} = \mathbf{B}^T \succ 0$ is allowed to be non-diagonal which makes the factorization non-unique. Our algorithm alternates between fixed-rank optimization and rank-one updates. When the rank is fixed, the problem is no longer convex but the search space has a Riemannian structure. We use the framework of optimization on Riemannian quotient manifolds to propose a Riemannian trust-region algorithm that generates low-cost (linear in n) iterates that converge super-linearly to a local minimum. Local minima are escaped by incrementing the rank until the global minimum in reached. The rank-one update is always selected to ensure a decrease of the cost.

Implementing the complete algorithm for a fixed value of the regularization parameter λ leads to a monotone convergence to the global minimum through a sequence of local minima of increasing ranks. Instead, we also modify λ along the way with a predictor-corrector method, thereby transforming most local minima of (5.1) (for fixed λ and fixed rank) into global minima of (5.1) for different values of λ. The resulting procedure, thus, provides a full regularization path of solutions at a very efficient numerical cost.

Not surprisingly, the proposed approach has links with several earlier contributions in the literature. Primarily, the idea of interlacing fixed-rank optimization with rank-one updates has been used in semidefinite programming (Burer and Monteiro, 2003; Journée et al., 2010). It is here extended to a nonsymmetric framework using the Riemannian geometry recently developed by Bonnabel and Sepulchre (2009); Meyer (2011); Meyer et al. (2011a). An improvement with respect to the earlier work of Burer and Monteiro (2003); Journée et al. (2010) is the use of duality gap certificate to discriminate between local and global minima and its efficient computation thanks to the chosen parameterization.

Schemes that combine fixed-rank optimization and special rank-one updates have appeared recently in the particular context of matrix completion (Keshavan et al., 2010; Wen et al., 2012). The framework presented here is in the same spirit but in a more general setting and with a global convergence analysis. Most other fixed-rank algorithms (Boumal and Absil, 2011; Keshavan et al., 2010; Meka et al., 2009; Meyer, 2011; Simonsson and Eldén, 2010; Srebro and Jaakkola, 2003; Vandereycken, 2013; Wen et al., 2012) for matrix completion fix the rank before. It is difficult to provide a tight comparison of the proposed algorithm to trace norm minimization algorithms that do not fix the rank a priori (Amit et al., 2007; Cai et al., 2010; Mazumder et al., 2010; Yuan et al., 2007). It should be emphasized, however, that most trace norm minimization algorithms use singular value thresholding operation at each iteration. This is the most numerically demanding step for these algorithms. For the matrix completion application, it involves computing (potentially all) the singular values of a *low-rank* + *sparse* matrix (Cai et al., 2010). In contrast, the proposed approach requires only dense linear algebra (linear in n) and rank-one updates using only dominant singular vectors and value of a sparse matrix. The main potential of the algorithm appears when computing the solution not for a single parameter λ but for a number of values of λ. We compute the entire regularization path with an efficient predictor-corrector strategy that convincingly outperforms the warm-restart strategy.

5.2 Relationship between convex program and fixed-rank formulation

Among different factorizations that exist to represent low-rank matrices, we use the factorization of Bonnabel and Sepulchre (2009); Meyer et al. (2011a) that decomposes a rank-p matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$ into

$$\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T, \tag{5.2}$$

where $\mathbf{U} \in \mathrm{St}(p, n)$, $\mathbf{V} \in \mathrm{St}(p, m)$ and $\mathbf{B} \in \mathrm{S}_{++}(p)$. $\mathrm{St}(p, n)$ is the Stiefel manifold or the set of $n \times p$ matrices with orthonormal columns. $\mathrm{S}_{++}(p)$ is the cone of $p \times p$ positive definite matrices. We stress that the scaling $\mathbf{B} = \mathbf{B}^T \succ 0$ is not required to be diagonal. The redundancy of this parameterization has non-trivial algorithmic implications (in Section 5.3) but we believe that it is also the key to success of the approach. Refer to (Keshavan et al., 2010; Meyer et al., 2011a) for earlier algorithms advocating matrix scaling and Section 5.6.1 for a numerical illustration. With the factorization $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$, the trace norm is $\|\mathbf{X}\|_* = \mathrm{Trace}(\mathbf{B})$ which is now *differentiable*. For a fixed rank p, the optimization problem (5.1) is recast as

$$\min_{\substack{\mathbf{U} \in \mathrm{St}(p,n) \\ \mathbf{B} \in \mathrm{S}_{++}(p) \\ \mathbf{V} \in \mathrm{St}(p,m)}} \quad F(\mathbf{U}\mathbf{B}\mathbf{V}^T) + \lambda \mathrm{Trace}(\mathbf{B}). \tag{5.3}$$

The search space of (5.3) is not Euclidean but the product space of two well-studied manifolds, namely, the Stiefel manifold $\mathrm{St}(p, n)$ (Edelman et al., 1998) and the cone of positive definite matrices $\mathrm{S}_{++}(p)$ (Bhatia, 2007; Smith, 2005). The column and row spaces of $\mathbf{X}$ are represented on $\mathrm{St}(p, n)$ and $\mathrm{St}(p, m)$ whereas the scaling factor is absorbed into $\mathrm{S}_{++}(p)$. A proper metric on the space takes into account both *rotational* and *scaling* invariance. We propose one that respects the rich geometry of the cone of positive definite matrices.

5.2.1 First-order optimality conditions

In order to relate the fixed-rank problem (5.3) to the convex optimization problem (5.1) we look at the necessary and sufficient optimality conditions that govern the solutions. The first-order necessary and sufficient optimality condition for the convex program (5.1) is (Bach, 2008; Recht et al., 2010)

$$\mathbf{0} \in \mathrm{Grad}_{\mathbf{X}} F + \lambda \partial \|\mathbf{X}\|_*, \tag{5.4}$$

where $\mathrm{Grad}_{\mathbf{X}}F$ is the Euclidean gradient of F at $\mathbf{X} \in \mathbb{R}^{n\times m}$ and $\partial\|\mathbf{X}\|_*$ is the sub-differential of the trace norm (Boyd and Vandenberghe, 2004; Cai et al., 2010; Toh and Yun, 2010). The first-order optimality conditions for the fixed-rank optimization problem (5.3) are given in Proposition 5.1 below. Subsequently, Proposition 5.2 gives the criterion under which a critical point of (5.3) is identified with the global minimizer of (5.1).

Proposition 5.1. *The first-order necessary optimality conditions of (5.3) are*

$$\begin{array}{lcl} \mathbf{SVB} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{SVB}) & = & \mathbf{0} \\ \mathrm{Sym}(\mathbf{U}^T\mathbf{SV} + \lambda\mathbf{I}) & = & \mathbf{0} \\ \mathbf{S}^T\mathbf{UB} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T\mathbf{S}^T\mathbf{UB}) & = & \mathbf{0}, \end{array} \tag{5.5}$$

where $\mathbf{X} = \mathbf{UBV}^T$ *(5.2),* $\mathrm{Sym}(\mathbf{\Delta}) = (\mathbf{\Delta} + \mathbf{\Delta}^T)/2$ *for any square matrix* $\mathbf{\Delta}$ *and* $\mathbf{S} = \mathrm{Grad}_{\mathbf{X}}F$*.* $\mathbf{S}$ *is referred to as the* dual variable *throughout the chapter.*

Proof. The first-order optimality conditions are derived either by writing the Lagrangian function of the problem (5.3) and looking at the *KKT first-order conditions* or by deriving the (Riemannian) gradient of the cost function (5.5) on the product space $\mathrm{St}(p, n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p, m)$ with the metric (5.12) proposed in Section 5.3. These two ways correspond to the two different viewpoints on equality constrained optimization shown in Figure 3.1.

□

Proposition 5.2. *A local minimum of (5.3)* $\mathbf{X} = \mathbf{UBV}^T$ *is also the global optimum of (5.1) iff* $\|\mathbf{S}\|_{op} = \lambda$ *where* $\mathbf{S} = \mathrm{Grad}_{\mathbf{X}}F$ *and* $\|\mathbf{S}\|_{op}$ *is the operator norm, i.e., the dominant singular value of* $\mathbf{S}$*. Moreover,* $\|\mathbf{S}\|_{op} \geq \lambda$ *and equality holds only at optimality. Consequently, a local minimum of (5.3) is identified with the global minimum of (5.1) if* $\|\mathbf{S}\|_{op} - \lambda \leq \epsilon$ *where* ϵ *is a defined threshold.*

Proof. This is in fact rewriting the first-order optimality condition of (5.1) (Cai et al., 2010; Ma et al., 2011). The proof is as follows.

From the characterization of sub-differential of trace norm (Recht et al., 2010) we have the following.

$$\partial\|\mathbf{X}\|_* = \{\mathbf{UV}^T + \mathbf{W} : \begin{array}{l} \mathbf{W} \text{ and } \mathbf{X} \text{ have orthogonal column and row spaces,} \\ \mathbf{W} \in \mathbb{R}^{n\times m} \text{ and } \|\mathbf{W}\|_{op} \leq 1\}, \end{array} \tag{5.6}$$

where $\mathbf{X} = \mathbf{UBV}^T$ (5.2). Since $\mathbf{X}$ is also a stationary point for the problem (5.3), the conditions (5.5) are satisfied including $\mathrm{Sym}(\mathbf{U}^T\mathbf{SV}) + \lambda\mathbf{I} = \mathbf{0}$, where $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a matrix, i.e., $\mathrm{Sym}(\mathbf{\Delta}) = (\mathbf{\Delta} + \mathbf{\Delta}^T)/2$ for any square matrix $\mathbf{\Delta}$. Based on the properties of a matrix norm we also have

$$\begin{array}{rcl} \lambda\mathbf{I} & = & -\mathrm{Sym}(\mathbf{U}^T\mathbf{SV}) \\ \Rightarrow \quad \lambda & = & \|\mathrm{Sym}(\mathbf{U}^T\mathbf{SV})\|_{op} \leq \|\mathbf{U}^T\mathbf{SV}\|_{op} \leq \|\mathbf{S}\|_{op}, \end{array}$$

where equality holds *if and only if* $\mathbf{U}$ and $\mathbf{V}$ correspond to the dominant row and column subspace of $\mathbf{S}$, respectively. That is, if $\mathbf{S} = -\lambda\mathbf{UV}^T + \mathbf{U}_\perp\mathbf{\Sigma}\mathbf{V}_\perp{}^T$ where $\mathbf{U}^T\mathbf{U}_\perp = \mathbf{0}$, $\mathbf{V}^T\mathbf{V}_\perp = \mathbf{0}$, $\mathbf{U}_\perp \in \mathrm{St}(n-p, n)$, $\mathbf{V}_\perp \in \mathrm{St}(m-p, m)$ and $\mathbf{\Sigma} \in \mathbb{R}^{(n-p)\times(m-p)}$ is a *diagonal* matrix with positive entries with $\|\mathbf{\Sigma}\|_{op} \leq \lambda$. For example, if $n \leq m$, then the left $(n-p)\times(n-p)$ part $\mathbf{\Sigma}$ is diagonal with positive entries and the rest all are zeros. It should be noted that this also means that $\mathbf{S} \in \lambda\partial\|\mathbf{X}\|_*$ such that $\mathbf{W} = \mathbf{U}_{\perp}\mathbf{\Sigma}\mathbf{V}_{\perp}{}^T$ which satisfies (5.6), and the global optimality condition (5.4) is attained. This completes the proof.

□

5.2.2 Duality gap computation

Proposition 5.2 provides a criterion to check global optimality of a local solution of (5.3). However, it provides no guarantees on *closeness* to the global solution. A better way of certifying optimality for the optimization problem (5.1) is provided by the notion of *duality gap*. The duality gap characterizes the difference of the obtained solution from the optimal solution and is always non-negative (Boyd and Vandenberghe, 2004, Chapter 5).

Proposition 5.3. *The Lagrangian dual formulation of (5.1) is*

$$\begin{array}{ll} \max\limits_{\mathbf{M}\in\mathbb{R}^{n\times m}} & -F^*(\mathbf{M}) \\ \text{subject to} & \|\mathbf{M}\|_{op} \leq \lambda, \end{array} \tag{5.7}$$

where $\mathbf{M}$ *is the* dual variable *and* $\|\cdot\|_{op}$ *is the* operator norm *of a matrix, i.e.,* $\|\mathbf{M}\|_{op}$ *is the* largest *singular value of* $\mathbf{M}$*. It should be stated that the operator norm* $\|\cdot\|_{op}$ *is the dual norm of the trace norm* $\|\cdot\|_*$*.* $F^* : \mathbb{R}^{n\times m} \to \mathbb{R}$ *is the Fenchel (convex) conjugate (Bach et al., 2011; Boyd and Vandenberghe, 2004) of* F*, defined as* $F^*(\mathbf{M}) = \sup\limits_{\mathbf{X}\in\mathbb{R}^{n\times m}} \left[\text{Trace}(\mathbf{M}^T\mathbf{X}) - F(\mathbf{X})\right]$*.*

Proof. Without loss of generality we introduce a dummy variable $\mathbf{Z} \in \mathbb{R}^{n\times m}$ to rephrase the optimization problem (5.1) as

$$\begin{array}{ll} \min\limits_{\mathbf{X},\mathbf{Z}\in\mathbb{R}^{n\times m}} & F(\mathbf{X}) + \lambda\|\mathbf{Z}\|_* \\ \text{subject to} & \mathbf{Z} = \mathbf{X}, \end{array}$$

where $\lambda > 0$ is the regularization parameter. The Lagrangian of the problem with dual variable $\mathbf{M} \in \mathbb{R}^{n\times m}$ is $\mathcal{L} : (\mathbb{R}^{n\times m}, \mathbb{R}^{n\times m}, \mathbb{R}^{n\times m}) \to \mathbb{R} : (\mathbf{X}, \mathbf{Z}, \mathbf{M}) \mapsto \mathcal{L}(\mathbf{X}, \mathbf{Z}, \mathbf{M}) = F(\mathbf{X}) + \lambda\|\mathbf{Z}\|_* + \text{Trace}(\mathbf{M}^T(\mathbf{Z} - \mathbf{X}))$. The *Lagrangian dual* function $G : \mathbb{R}^{n\times m} \to \mathbb{R}$ of the Lagrangian $\mathcal{L}$ is, computed as (Bach et al., 2011; Boyd and Vandenberghe, 2004)

$$\begin{array}{rcl} G(\mathbf{M}) & = & \min\limits_{\mathbf{X},\mathbf{Z}\in\mathbb{R}^{n\times m}} \quad F(\mathbf{X}) - \text{Trace}(\mathbf{M}^T\mathbf{X}) + \text{Trace}(\mathbf{M}^T\mathbf{Z}) + \lambda\|\mathbf{Z}\|_* \\ \Rightarrow\ G(\mathbf{M}) & = & \min\limits_{\mathbf{X}\in\mathbb{R}^{n\times m}} \quad \{F(\mathbf{X}) - \text{Trace}(\mathbf{M}^T\mathbf{X})\} + \min\limits_{\mathbf{Z}\in\mathbb{R}^{n\times m}} \quad \{\text{Trace}(\mathbf{M}^T\mathbf{Z}) + \lambda\|\mathbf{Z}\|_*\} \end{array}$$

Using the operator norm $\|\cdot\|_{op}$ as the dual of the nuclear norm $\|\cdot\|_*$,we have

$$\min_{\mathbf{Z}\in\mathbb{R}^{n\times m}} \quad \text{Trace}(\mathbf{M}^T\mathbf{Z}) + \lambda\|\mathbf{Z}\|_* = 0 \quad \text{if} \quad \|\mathbf{M}\|_{op} \leq \lambda.$$

Similarly, using the concept of Fenchel conjugate of a function we have

$$\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} \quad F(\mathbf{X}) - \text{Trace}(\mathbf{M}^T\mathbf{X}) = -F^*(\mathbf{M})$$

where F^* is the Fenchel conjugate of F, defined as $F^*(\mathbf{M}) = \sup_{\mathbf{X}\in\mathbb{R}^{n\times m}} \left[\mathrm{Trace}(\mathbf{M}^T\mathbf{X}) - F(\mathbf{X})\right]$ (Bach et al., 2011; Boyd and Vandenberghe, 2004). Equivalently, subject to $\|\mathbf{M}\|_{op} \leq \lambda$, the final expression for the dual function is $G(\mathbf{M}) = -F^*(\mathbf{M})$ (Bach et al., 2011) and the Lagrangian dual formulation is

$$\max_{\mathbf{M}\in\mathbb{R}^{n\times m}} \; -F^*(\mathbf{M}) \quad \text{such that} \quad \|\mathbf{M}\|_{op} \leq \lambda.$$

This proves the proposition.

□

Given a primal candidate $\mathbf{X} \in \mathbb{R}^{n\times m}$ and a *dual feasible candidate* $\mathbf{M} \in \mathbb{R}^{n\times m}$ such that $\|\mathbf{M}\|_{op} \leq \lambda$, the effective expression of duality gap is

$$F(\mathbf{X}) + \lambda\|\mathbf{X}\|_* + F^*(\mathbf{M}). \tag{5.8}$$

A good choice for the dual candidate $\mathbf{M}$ is $\mathbf{S}$ ($=\mathrm{Grad}_{\mathbf{X}}F$) with appropriate scaling to satisfy the operator norm constraint, i.e., $\mathbf{M} = \min(1, \frac{\lambda}{\|\mathbf{S}\|_{op}})\mathbf{S}$ (Bach et al., 2011).

5.3 A Riemannian optimization approach for the fixed-rank optimization problem (5.3)

In this section we propose an algorithm for the problem (5.3). In contrast to first-order optimization algorithms proposed earlier by Keshavan et al. (2010); Meyer et al. (2011a,b), we develop a second-order trust-region algorithm that has a provably quadratic rate of convergence (Absil et al., 2008, Chapter 7). We rewrite (5.3) as

$$\min_{\substack{\mathbf{U}\in\mathrm{St}(p,n)\\ \mathbf{B}\in\mathrm{S}_{++}(p)\\ \mathbf{V}\in\mathrm{St}(p,m)}} \quad f(\mathbf{U},\mathbf{B},\mathbf{V}), \tag{5.9}$$

where $f : \mathrm{St}(p,n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p,m) \rightarrow \mathbb{R} : (\mathbf{U},\mathbf{B},\mathbf{V}) \mapsto F(\mathbf{U}\mathbf{B}\mathbf{V}^T) + \lambda\mathrm{Trace}(\mathbf{B})$ and is introduced for notational convenience. An important observation for second-order algorithms (Absil et al., 2008) is that the local minima of the problem (5.9) are not isolated in the search space

$$\mathcal{M}_p := \mathrm{St}(p,n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p,m), \tag{5.10}$$

where $\mathrm{St}(p,n)$ is the set of $n \times p$ matrices with orthonormal columns and $\mathrm{S}_{++}(p)$ is the set of symmetric positive definite matrices. This is because the cost function f in (5.9) is *invariant* under rotations, i.e., $\mathbf{U}\mathbf{B}\mathbf{V}^T = (\mathbf{U}\mathbf{O})(\mathbf{O}^T\mathbf{B}\mathbf{O})(\mathbf{V}\mathbf{O})^T$ for any $p\times p$ rotation matrix $\mathbf{O} \in \mathcal{O}(p)$ such that $\mathbf{O}\mathbf{O}^T = \mathbf{O}^T\mathbf{O} = \mathbf{I}$. To remove the symmetry of the cost function in (5.9), we identify all the points of the search space that belong to the equivalence class defined by

$$[(\mathbf{U},\mathbf{B},\mathbf{V})] = \{(\mathbf{U}\mathbf{O}, \mathbf{O}^T\mathbf{B}\mathbf{O}, \mathbf{V}\mathbf{O}) : \mathbf{O} \in \mathcal{O}(p)\}. \tag{5.11}$$

The set of all such equivalence classes is denoted by $\mathcal{M}_p/\mathcal{O}(p)$ that has the structure of a smooth quotient manifold of $\mathcal{M}_p/\mathcal{O}(p)$ by $\mathcal{O}(p)$ (Lee, 2003, Theorem 9.16). It should be emphasized that the set $\mathcal{O}(p)$

takes away all the symmetry of the total space $\mathcal{M}_p$ (5.10) and consequently, the dimension of $\mathcal{M}_p$ is $(n+m-p)p$ which is equal to the dimension of the set of rank-p matrices. The dimension of the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ is obtained by subtracting the dimension of $\mathcal{O}(p)$ from the dimension of the product space $\mathcal{M}_p$ (5.10).

Problem (5.9) is, therefore, conceptually an *unconstrained* optimization problem on the *quotient manifold* $\mathcal{M}_p/\mathcal{O}(p)$ where the minima are *isolated*. Computations are performed in the total space $\mathcal{M}_p$, which is the product space of well-studied manifolds.

5.3.1 The Riemannian submersion of $\mathcal{M}_p/\mathcal{O}(p)$

An element of the total space $\mathcal{M}_p$ (5.10) is represented by x and its equivalence class is represented by $[x]$. The equivalence relationship (5.11) is represented the notation $\sim$. $[x] := \{y \in \mathcal{M}_p : y \sim x\}$. The matrix characterizations are provided in Table 5.1.

Following the theory of Riemannian submersion (Absil et al., 2008, Chapter 4), the running theme of the present thesis (presented earlier in Section 3.2.2), the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ has a Riemannian submersion structure, provided we endow the total space $\mathcal{M}_p$ with a Riemannian *metric* (a smooth inner product on the tangent space $T_x\mathcal{M}_p$). The flexibility of choosing a Riemannian metric has been exploited earlier in Chapters 3 and 4. Here we are particularly motivated by the natural geometry of the Stiefel manifold $\mathrm{St}(p,n)$ (Edelman et al., 1998) and the symmetric positive definite cone $\mathrm{S}_{++}(r)$ (Bhatia, 2007) which have a rich history of their own. To this end, we propose the metric $g_x : T_x\mathcal{M}_p \times T_x\mathcal{M}_p \to \mathbb{R}$ at $x \in \mathcal{M}_p$ is

$$g_x(\xi_x, \eta_x) = \langle \xi_{\mathbf{U}}, \eta_{\mathbf{U}} \rangle + \langle \xi_{\mathbf{B}}\mathbf{B}^{-1}, \mathbf{B}^{-1}\eta_{\mathbf{B}} \rangle + \langle \xi_{\mathbf{V}}, \eta_{\mathbf{V}} \rangle, \tag{5.12}$$

where $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product and ξ_x and η_x are any tangent vectors in $T_x\mathcal{M}_p$ with matrix characterizations $(\xi_{\mathbf{U}}, \xi_{\mathbf{B}}, \xi_{\mathbf{V}})$ and $(\eta_{\mathbf{U}}, \eta_{\mathbf{B}}, \eta_{\mathbf{V}})$, respectively and are shown in Table 5.1. The metric (5.12) is the summation of the normal metric of the Stiefel manifold (Edelman et al., 1998) and the natural (bi-invariant) metric of the positive definite cone (Bhatia, 2007; Smith, 2005). The metric so proposed also allows us to construct *special curves* on the set $\mathcal{M}_p/\mathcal{O}(p)$ that have particular properties. This is presented in Section 5.5.

A matrix representation of the tangent space $T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$ at $[x] \in \mathcal{M}_p/\mathcal{O}(p)$ relies on the decomposition of $T_x\mathcal{M}_p$ into complementary *vertical* and *horizontal* subspaces. Once the Riemannian metric is proposed, the characterization of the *horizontal space*, that is the matrix representation of the abstract tangent space of the quotient manifold $T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$ at $[x] \in \mathcal{M}_p/\mathcal{O}(p)$, follows through. We show the final expressions in Table 5.1. It should be stated that tangent vectors on the quotient manifold call for matrix representatives, called *horizontal lifts*, which are vectors in the horizontal space $\mathcal{H}_x$. Finally, the induced metric at $[x]$ on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ is

$$g_{[x]}(\xi_{[x]}, \eta_{[x]}) := g_x(\xi_x, \eta_x), \tag{5.13}$$

where $g_x(\cdot,\cdot)$ is the Riemannian metric proposed in (5.12) and $\xi_{[x]}$ and $\eta_{[x]}$ are tangent vectors in $T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$ with horizontal lifts ξ_x and η_x in the horizontal space $\mathcal{H}_x$ at x.

	Characterization of the set of the factorization $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ (5.2)
Matrix representation	$x = (\mathbf{U}, \mathbf{B}, \mathbf{V})$
Total space $\mathcal{M}_p$	$\mathrm{St}(p,n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p,m)$
Group action	$(\mathbf{U}\mathbf{O}, \mathbf{O}^T\mathbf{B}\mathbf{O}, \mathbf{V}\mathbf{O}), \mathbf{O} \in \mathcal{O}(p)$
Equivalence class $[x]$	$[(\mathbf{U}, \mathbf{B}, \mathbf{V})] = \{(\mathbf{U}\mathbf{O}, \mathbf{O}^T\mathbf{B}\mathbf{O}, \mathbf{V}\mathbf{O}) : \mathbf{O} \in \mathcal{O}(p)\}$
Quotient space $\mathcal{M}_p/\sim$	$\mathrm{St}(p,n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p,m)/\mathcal{O}(p)$
Tangent vectors in $T_x\mathcal{M}_p$	$\{(\xi_{\mathbf{U}}, \xi_{\mathbf{B}}, \xi_{\mathbf{V}}) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{m\times r} :$ $\mathbf{U}^T\xi_{\mathbf{U}} + \xi_{\mathbf{U}}^T\mathbf{U} = 0,$ $\xi_{\mathbf{B}}^T = \xi_{\mathbf{B}},$ $\mathbf{V}^T\xi_{\mathbf{V}} + \xi_{\mathbf{V}}^T\mathbf{V} = 0\}$
Metric $g_x(\xi_x, \eta_x)$ for any $\xi_x, \eta_x \in T_x\mathcal{M}_p$	$\langle \eta_{\mathbf{U}}, \xi_{\mathbf{U}} \rangle + \langle \eta_{\mathbf{B}}\mathbf{B}^{-1}, \mathbf{B}^{-1}\xi_{\mathbf{B}} \rangle + \langle \eta_{\mathbf{V}}, \xi_{\mathbf{V}} \rangle$
Vertical tangent vectors in $\mathcal{V}_x$	$\{(\mathbf{U}\boldsymbol{\Omega}, \mathbf{B}\boldsymbol{\Omega} - \boldsymbol{\Omega}\mathbf{B}, \mathbf{V}\boldsymbol{\Omega}) : \boldsymbol{\Omega} \in \mathbb{R}^{r\times r}, \boldsymbol{\Omega}^T = -\boldsymbol{\Omega}\}$
Horizontal tangent vectors in $\mathcal{H}_x$	$\{(\zeta_{\mathbf{U}}, \zeta_{\mathbf{B}}, \zeta_{\mathbf{V}}) \in T_x\mathcal{M}_p :$ $(\zeta_{\mathbf{U}}^T\mathbf{U} + \mathbf{B}^{-1}\zeta_{\mathbf{B}} - \zeta_{\mathbf{B}}\mathbf{B}^{-1} + \zeta_{\mathbf{V}}^T\mathbf{V})$ is symmetric$\}$
$\Psi(\cdot)$ projects an ambient vector $(\mathbf{Z}_{\mathbf{U}}, \mathbf{Z}_{\mathbf{B}}, \mathbf{Z}_{\mathbf{V}}) \in \mathbb{R}^{n\times r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{m\times r}$ onto $T_x\mathcal{M}_p$	$\Psi_x(\mathbf{Z}_{\mathbf{U}}, \mathbf{Z}_{\mathbf{B}}, \mathbf{Z}_{\mathbf{V}})$ $= (\mathbf{Z}_{\mathbf{U}} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T\mathbf{Z}_{\mathbf{U}}), \mathrm{Sym}(\mathbf{Z}_{\mathbf{B}}),$ $\mathbf{Z}_{\mathbf{V}} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T\mathbf{Z}_{\mathbf{V}}))$
$\Pi(\cdot)$ projects a tangent vector $\eta_x \in T_x\mathcal{M}_p$ onto $\mathcal{H}_x$	$\Pi_x(\eta_x) = (\eta_{\mathbf{U}} - \mathbf{U}\boldsymbol{\Omega}, \eta_{\mathbf{B}} - (\mathbf{B}\boldsymbol{\Omega} - \boldsymbol{\Omega}\mathbf{B}), \eta_{\mathbf{V}} - \mathbf{V}\boldsymbol{\Omega}),$ where $\boldsymbol{\Omega}$ is the unique solution to the Lyapunov equation $\boldsymbol{\Omega}\mathbf{B}^2 + \mathbf{B}^2\boldsymbol{\Omega} = \mathbf{B}(\mathrm{Skew}(\mathbf{U}^T\eta_{\mathbf{U}}) - 2\mathrm{Skew}(\mathbf{B}^{-1}\eta_{\mathbf{B}})$ $+\mathrm{Skew}(\mathbf{V}^T\eta_{\mathbf{V}}))\mathbf{B}$
Retraction $R_x(\xi_x)$ that maps a search direction $\xi_x \in \mathcal{H}_x$ onto $\mathcal{M}_p$	$(\mathrm{uf}(\mathbf{U} + \xi_{\mathbf{U}}), \mathbf{B}^{\frac{1}{2}}\mathrm{expm}(\mathbf{B}^{-\frac{1}{2}}\xi_{\mathbf{B}}\mathbf{B}^{-\frac{1}{2}})\mathbf{B}^{\frac{1}{2}}, \mathrm{uf}(\mathbf{V} + \xi_{\mathbf{V}}))$

TABLE 5.1: Matrix characterizations of various objects on the manifold $\mathcal{M}_p/\mathcal{O}(p)$. $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of the polar factorization of a full column-rank matrix, i.e., $\mathrm{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1/2}$, $\mathrm{expm}(\cdot)$ is the *matrix exponential* operator, $\mathrm{Skew}(\cdot)$ extracts the skew-symmetric of a square matrix, i.e., $\mathrm{Skew}(\boldsymbol{\Delta}) = (\boldsymbol{\Delta} - \boldsymbol{\Delta}^T)/2$, and $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a matrix, i.e., $\mathrm{Sym}(\boldsymbol{\Delta}) = (\boldsymbol{\Delta} + \boldsymbol{\Delta}^T)/2$.

Table 5.1 summarizes the concrete matrix operations involved in computing horizontal vectors. Starting from an arbitrary matrix (with appropriate dimensions), two linear projections are needed: the first projection with Ψ_x is onto the tangent space of the total space, while the second projection with Π_x is onto the horizontal subspace.

Finally, a retraction of interest that maps vectors from the horizontal space $\mathcal{H}_x$ onto $\mathcal{M}_p$ is (Absil et al., 2008; Meyer, 2011);

$$R_x(\xi_x) = (\mathrm{uf}(\mathbf{U}+\xi_{\mathbf{U}}), \mathbf{B}^{\frac{1}{2}}\mathrm{expm}(\mathbf{B}^{-\frac{1}{2}}\xi_{\mathbf{B}}\mathbf{B}^{-\frac{1}{2}})\mathbf{B}^{\frac{1}{2}}, \mathrm{uf}(\mathbf{V}+\xi_{\mathbf{V}})), \tag{5.14}$$

where $\mathrm{uf}(\cdot)$ extracts the orthogonal factor of the polar factorization of a full-column rank matrix, i.e., $\mathrm{uf}(\mathbf{A}) = \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1/2}$ and $\mathrm{expm}(\cdot)$ is the *matrix exponential* operator. The retraction on the positive definite cone is the natural exponential mapping for the metric (5.12) (Smith, 2005). The combination of these well-known retractions on the individual manifolds is also a valid retraction on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ by virtue of Absil et al. (2008, Proposition 4.1.3).

5.3.2 Gradient and Hessian computations in Riemannian submersion

The choice of the metric (5.12), which is invariant along the equivalence class $[x] := [\mathbf{U}, \mathbf{B}, \mathbf{V}]$ (5.11) turns the quotient manifold $\mathcal{M}_p$ into a Riemannian submersion of $(\mathcal{M}_p, g)$ (Lee, 2003, Theorem 9.16) and (Absil et al., 2008, Section 3.6.2). As shown by Absil et al. (2008), this special construction allows for a convenient matrix representation of the Riemannian gradient (Absil et al., 2008, Section 3.6.2) and the Riemannian Hessian (Absil et al., 2008, Proposition 5.3.3) on the abstract manifold $\mathcal{M}_p/\mathcal{O}(p)$ from the computation of their counterparts in the total space $\mathcal{M}_p$.

The Riemannian gradient $\mathrm{grad}_{[x]}f$ of $f : \mathcal{M}_p \to \mathbb{R} : x \mapsto f(x) = F(\mathbf{U}\mathbf{B}\mathbf{V}^T) + \lambda\mathrm{Trace}(\mathbf{B})$ on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ is uniquely represented by its horizontal lift in $\mathcal{H}_x$ which has the matrix representation

$$\text{horizontal lift of } \mathrm{grad}_{[x]}f = \mathrm{grad}_x f, \tag{5.15}$$

where $\mathrm{grad}_x f$ is the gradient of the function f on the total space $\mathcal{M}_p$. It should be emphasized that $\mathrm{grad}_x f$ is in the the tangent space $T_x\mathcal{M}_p$. However, due to invariance of the cost f along the equivalence class $[x]$, $\mathrm{grad}_x f$ also belongs to the horizontal space $\mathcal{H}_x$ and hence, the equality in (5.15) (Absil et al., 2008, Section 3.6.2). The matrix expression of $\mathrm{grad}_x f$ in the total space $\mathcal{M}_p$ at $x = (\mathbf{U}, \mathbf{B}, \mathbf{V})$ is obtained by solving the convex quadratic program

$$\begin{array}{rcll} \mathrm{grad}_x f & = & \underset{\zeta_x \in T_x\mathcal{M}_p}{\arg\min} & f(x) - \langle f_x(x), \zeta_x\rangle + \frac{1}{2}g_x(\zeta_x, \zeta_x) \\ & = & \underset{\substack{\zeta_{\mathbf{U}} \in \mathbb{R}^{n\times r},\\ \zeta_{\mathbf{B}} \in \mathbb{R}^{r\times r},\\ \zeta_{\mathbf{V}} \in \mathbb{R}^{m\times r}}}{\arg\min} & -\langle f_{\mathbf{U}}, \zeta_{\mathbf{U}}\rangle - \langle f_{\mathbf{B}}, \zeta_{\mathbf{B}}\rangle - \langle f_{\mathbf{V}}, \zeta_{\mathbf{V}}\rangle + \frac{1}{2}(\langle \zeta_{\mathbf{U}}, \zeta_{\mathbf{U}}\rangle + \langle \zeta_{\mathbf{B}}, \zeta_{\mathbf{B}}\rangle + \langle \zeta_{\mathbf{V}}, \zeta_{\mathbf{V}}\rangle) \\ & & \text{subject to} & \mathbf{U}^T\zeta_{\mathbf{U}} + \zeta_{\mathbf{U}}^T\mathbf{U} = 0, \zeta_{\mathbf{B}}^T = \zeta_{\mathbf{B}}, \mathbf{V}^T\zeta_{\mathbf{V}} + \zeta_{\mathbf{V}}^T\mathbf{V} = 0, \end{array} \tag{5.16}$$

where $f_x(x)$ is the first-order derivative of the function f in (5.9) with matrix representation $(f_{\mathbf{U}}, f_{\mathbf{B}}, f_{\mathbf{V}}) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$, $\langle\cdot,\cdot\rangle$ is the standard Euclidean inner product, and $g_x(\cdot,\cdot)$ is the Riemannian metric (5.12). This solution of (5.16) admits the solution (after few standard computations)

$$\mathrm{grad}_x f = (f_{\mathbf{U}} - \mathbf{U}\mathrm{Sym}(\mathbf{U}^T f_{\mathbf{U}}), \mathbf{B}\mathrm{Sym}(f_{\mathbf{B}})\mathbf{B}, f_{\mathbf{V}} - \mathbf{V}\mathrm{Sym}(\mathbf{V}^T f_{\mathbf{V}})), \tag{5.17}$$

where $(\mathbf{U}, \mathbf{B}, \mathbf{V})$ is the matrix representation of $x \in \mathcal{M}_p$, $(f_{\mathbf{U}}, f_{\mathbf{B}}, f_{\mathbf{V}}) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$ is matrix representation of $f_x(x)$ (the first-order derivative of the cost function f), and $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix.

In addition to the Riemannian gradient, any optimization algorithm that makes use of second-order information also requires the directional derivative of the Riemannian gradient along a search direction. This involves the choice of an *affine connection* ∇ on the manifold $\mathcal{M}_p$. The affine connection provides a definition for the *covariant derivative* of a tangent vector η_x with respect to a tangent vector ν_x, denoted by $\nabla_{\nu_x}\eta_x$. Imposing an additional compatibility condition with the Riemannian metric fixes the so-called *Riemannian* connection which is always unique (Absil et al., 2008, Theorem 5.3.1 and Section 5.2). The Riemannian connection $\nabla_{\nu_{[x]}}\eta_{[x]}$ on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ for tangent vectors $\nu_{[x]}, \eta_{[x]} \in T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$ is uniquely represented by its horizontal lift in the horizontal space $\mathcal{H}_x$ which is (Absil et al., 2008, Proposition 5.3.3)

$$\text{horizontal lift of } \nabla_{\nu_{[x]}}\eta_{[x]} = \Pi_x(\nabla_{\nu_x}\eta_x), \tag{5.18}$$

where $\nu_{[x]}$ and $\eta_{[x]}$ are tangent vectors on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ and ν_x and η_x are their horizontal lifts in $\mathcal{H}_x$, respectively. Here $\Pi_x(\cdot) : T_x\mathcal{M}_p \to \mathcal{H}_x$ extracts the horizontal component of a tangent vector in $T_x\mathcal{M}_p$ and is worked out in Table 5.1.

Once again, the Riemannian connection $\nabla_{\nu_x}\eta_x$ on the total space $\mathcal{M}_p$ has well-known expression as a result of the individual Riemannian connection characterizations on $\mathrm{St}(p, n)$ (Absil et al., 2008; Journée, 2009) and on $\mathrm{S}_{++}(p)$ (Bhatia, 2007; Smith, 2005). The Riemannian connection on the Stiefel manifold $\mathrm{St}(p, n)$ is derived by Journée (2009, Example 4.3.6) and on the positive definite cone $\mathrm{S}_{++}(p)$ is derived by Meyer (2011, Appendix B). Finally, the Riemannian connection on the total space is given by the product structure

$$\nabla_{\nu_x}\eta_x = \Psi_x(\mathrm{D}\eta_x[\nu_x]) - \Psi_x(\nu_{\mathbf{U}}\mathrm{Sym}(\mathbf{U}^T\eta_{\mathbf{U}}), \mathrm{Sym}(\nu_{\mathbf{B}}\mathbf{B}^{-1}\eta_{\mathbf{B}}), \nu_{\mathbf{V}}\mathrm{Sym}(\mathbf{V}^T\eta_{\mathbf{V}})), \tag{5.19}$$

where ν_x and η_x are tangent vectors in $T_x\mathcal{M}_p$ with matrix representatives $(\nu_{\mathbf{U}}, \nu_{\mathbf{B}}, \nu_{\mathbf{V}})$ and $(\eta_{\mathbf{U}}, \eta_{\mathbf{B}}, \eta_{\mathbf{V}})$, $\Psi_x(\cdot)$ is the projection operator that projects an ambient vector on the tangent space $T_x\mathcal{M}_p$ (worked out in Table 5.1), $\mathrm{Sym}(\cdot)$ extracts the symmetric part of a square matrix, and $\mathrm{D}\eta_x[\nu_x]$ is the standard Euclidean directional derivative of η_x in the direction ν_x, i.e. $\mathrm{D}\eta_x[\nu_x] = \lim_{t\to 0}(\eta_{x+t\nu_x} - \eta_x)/t$.

The Riemannian Hessian operation (along a tangent vector) $\mathrm{Hess}_{[x]}f[\xi_{[x]}]$ of the cost function f in (5.9) is defined as the covariant derivative of the Riemannian gradient $\mathrm{grad}_{[x]}f$ in the direction $\xi_{[x]} \in T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$. The horizontal lift of the Riemannian Hessian, from (5.18), in $\mathcal{H}_x$ has the matrix expression

$$\text{horizontal lift of } \mathrm{Hess}_{[x]}f[\xi_{[x]}] = \Pi_x(\nabla_{\xi_x}\mathrm{grad}_x f), \tag{5.20}$$

where $\Pi_x(\cdot)$ is the projection operator that extracts the horizontal component (defined in Table 5.1), $\nabla_{\xi_x}\mathrm{grad}_x f$ is the Riemannian connection on the total space $\mathcal{M}_p$ shown in (5.19), and $\xi_x \in \mathcal{H}_x$ is the horizontal lift of $\xi_{[x]} \in T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$.

5.3.3 Riemannian trust-region algorithm on $\mathcal{M}_p/\mathcal{O}(p)$

The optimization method that we consider is the Riemannian *trust-region method.* Analogous to trust-regions in the Euclidean space (Nocedal and Wright, 2006, Chapter 4), trust-region algorithms on a Riemannian quotient manifold with (locally in the neighborhood of the minimum) quadratic rate convergence have been proposed by Absil et al. (2008, Chapter 7). See Absil et al. (2008, Section 7.4.2) for assumptions. It should be stated that these assumptions hold in our case.

Similar to the Euclidean case, at each iteration the trust-region algorithm involves a step to compute a search direction and a subsequent retraction operation to compute the next iterate. For computing the search direction, we solve the *trust-region subproblem* on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$. Conceptually, the trust-region subproblem is formulated as the minimization of the locally-quadratic model of the cost function f in (5.9) at $[x] \in \mathcal{M}_p/\mathcal{O}(p)$

$$\begin{array}{ll} \underset{\xi_{[x]} \in T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))}{\arg\min} & g_{[x]}(\xi_{[x]}, \mathrm{grad}_{[x]} f) + \frac{1}{2} g_{[x]}(\xi_{[x]}, \mathrm{Hess}_{[x]} f[\xi_{[x]}]) \\ \text{subject to} & g_{[x]}(\xi_{[x]}, \xi_{[x]}) \leq \Delta^2, \end{array} \tag{5.21}$$

where $g_{[x]}(\cdot,\cdot)$ is the Riemannian metric on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$, $\mathrm{grad}_{[x]} f$ is the Riemannian gradient of the cost function f in (5.9), and $\mathrm{Hess}_{[x]} f[\xi_{[x]}]$ is the Riemannian Hessian applied along the tangent vector $\xi_{[x]} \in T_{[x]}(\mathcal{M}_p/\mathcal{O}(p))$ on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$, and Δ is the trust-region radius. Computationally, however, we *horizontally* lift the abstract subproblem (5.21) to the horizontal space $\mathcal{H}_x$ which boils down to the expression

$$\begin{array}{ll} \underset{\xi_x \in \mathcal{H}_x}{\arg\min} & g_x(\xi_x, \mathrm{grad}_x f) + \frac{1}{2} g_x(\xi_x, \Pi_x(\nabla_{\xi_x} \mathrm{grad}_x f)) \\ \text{subject to} & g_x(\xi_x, \xi_x) \leq \Delta^2, \end{array} \tag{5.22}$$

where $g_x(\cdot,\cdot)$ is the Riemannian metric on the total space $\mathcal{M}_p$ and Δ is the trust-region radius. Here $\mathrm{grad}_x f$ is the Riemannian gradient of the cost function f on the total space $\mathcal{M}_p$ and is equal to the horizontal lift of $\mathrm{grad}_{[x]} f$ (5.15). Similarly, $\Pi_x(\nabla_{\xi_x} \mathrm{grad}_x f)$ is the horizontal lift of $\mathrm{Hess}_{[x]} f[\xi_{[x]}]$ (5.18), where $\nabla_{\xi_x} \mathrm{grad}_x f$ is the covariant derivative of the Riemannian gradient $\mathrm{grad}_x f$ (5.19) along the horizontal vector $\xi_x \in \mathcal{H}_x$ and $\Pi_x(\cdot)$ extracts the horizontal component of a tangent vector and is worked out in Table 5.1.

Solving the above trust-region subproblem (5.22) leads to a direction $\xi_x \in \mathcal{H}_x$ that minimizes the quadratic model. Depending on whether the decrease of the cost function is sufficient or not, the potential iterate is accepted or rejected. In particular, we implement the Riemannian trust-region algorithm of Absil et al. (2008, Algorithm 10) using the generic solver GenRTR (Baker et al., 2007), where the trust-region subproblem (5.22) is solved using the *truncated conjugate-gradient* method of Absil et al. (2008, Algorithm 11) that does not require *inverting* the Hessian. The stopping criterion for the trust-region subproblem is

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

where r_t is the residual of the subproblem at t^{th} iteration of the truncated conjugate-gradient method (Absil et al., 2008, (7.10)). The parameters θ and κ are set to 1 and 0.1 as suggested by Absil et al.

(2008, Section 7.5). The parameter $\theta = 1$ ensures that we seek a quadratic rate of convergence of the trust-region algorithm in the neighborhood of a local minimum.

The convergence of the Riemannian trust-region algorithm to critical points follows from the analysis by Absil et al. (2008, Section 7.4), where the algorithm is shown to be *globally convergent* implying that the Riemannian trust-region algorithm converges to critical points from *all* initial conditions. It should be note that, theoretically, the trust-region method guarantee convergence *only* to critical points. Practically, however, convergence to local minima are observed.

5.3.4 Numerical complexity

The numerical complexity of manifold-based optimization methods depends on the computational cost of the components listed in Table 5.1 and the matrix computations of the Riemannian gradient and Hessian operations. In particular, the numerical complexity per iteration of the proposed trust-region algorithm for (5.9) depends on the computational cost of the following ingredients.

1. Cost function $f(x)$ in (5.9): The computational cost is problem dependent.
2. Metric g_x (5.12):
   The dominant computational cost comes from computing terms like $\xi_{\mathbf{U}}^T \eta_{\mathbf{U}}$ which requires a numerical cost of $O(np^2)$. Other matrix operations involve handling matrices of size $p \times p$ with total computational cost of $O(p^3)$.
3. Projecting an ambient vector onto the tangent space $T_x\mathcal{M}_p$ with $\Psi_x(\cdot)$ (Table 5.1):
   It involves multiplications between matrices of sizes $n \times p$ and $p \times p$ which costs $O(np^2)$. Other operations involve handling matrices of size $r \times r$.
4. Projecting a tangent vector onto the horizontal space $\mathcal{H}_x$ with $\Pi_x(\cdot)$ (Table 5.1):
   - Forming the Lyapunov equations: Dominant computational cost of $O(np^2 + mp^2)$ with matrix multiplications that cost $O(p^3)$.
   - Solving the Lyapunov equation costs $O(p^3)$ (Bartels and Stewart, 1972). An efficient solution approach is presented in Appendix A.
5. Retraction $R_x(\cdot)$ (5.14):
   - Computing the retraction on the $\mathrm{St}(p, n)$ (the set of matrices of size $n \times p$ with orthonormal columns) costs $O(np^2)$
   - Computing the retraction on the set of positive definite matrices $\mathrm{S}_{++}(p)$ involves matrix exponential operations which cost $O(p^3)$.
6. Riemannian gradient $\mathrm{grad}_x f$ (5.17):
   First, it involves computing the first-order derivative $f_x(x)$ of the cost function f which depends on the cost function f. Second, extra modifications to these derivatives involve matrix multiplications between matrices of sizes $n \times p$ and $p \times p$ which costs $O(np^2)$.
7. Riemannian connection $\nabla_{\xi_x} \mathrm{grad}_x f$ in the direction $\xi_x \in \mathcal{H}_x$ on the total space (5.19):
   The Riemannian connection on each of the two manifolds, $\mathrm{St}(p, n)$ and $\mathrm{S}_{++}(p)$, consists of two terms. The first term is the projection of the standard Euclidean directional derivative of the

Riemannian gradient in the direction ξ_x, i.e., $\mathrm{Dgrad}_x f[\xi_x]$. The second term is the *correction term* corresponds to the manifold structure and the metric.

- $\mathrm{Dgrad}_x f[\xi_{\bar{x}}]$: The computational cost depends on the cost function f and its first-order derivative.
- Correction term: It involves matrix multiplications with total cost of $O(np^2 + p^3)$.

It is clear that all the manifold related operations are of linear complexity in n and m, and cubic in p. For the case of interest, $p \ll \min(n,m)$, these operations are computationally very efficient. The ingredients that depend on the problem at hand are the evaluation of the cost function f and computation of its first-order derivative and its directional derivative along a search direction. In Section 5.6, these computations are worked out for two specific examples of low-rank matrix completion and multivariate regression, where we exploit the least-squares nature of the cost function.

5.4 An optimization scheme for the trace norm regularized convex problem (5.1)

Starting with a rank-1 problem, we alternate a second-order local optimization algorithm on fixed-rank manifold with a first-order rank-one update in order to propose an algorithm for the convex problem with trace norm penalty (5.1). The scheme is shown in Table 5.2. The rank-one update decreases the cost with the updated iterate in $\mathcal{M}_{p+1}$.

Proposition 5.4. *Assume that the function F in (5.1) has* Lipschitz *continuous derivative with the Lipschitz constant L_F such that $\|\mathrm{Grad}_{\mathbf{X}} F - \mathrm{Grad}_{\mathbf{Y}} F\|_F \leq L_F \|\mathbf{X} - \mathbf{Y}\|_F$ for all $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n\times m}$, where $\mathrm{Grad}_{\mathbf{X}} F$ is Euclidean gradient of the function F in $\mathbb{R}^{n\times m}$. If $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ is a stationary point of (5.3) with $(\mathbf{U},\mathbf{B},\mathbf{V}) \in \mathrm{St}(p_0,n) \times \mathrm{S}_{++}(p_0) \times \mathrm{St}(p_0,m)$, then the rank-one update*

$$\mathbf{X}_+ = \mathbf{X} - \beta u v^T \tag{5.23}$$

ensures a decrease in the cost function $F(\mathbf{X}) + \lambda\|\mathbf{X}\|_*$, provided that $\beta > 0$ is sufficiently small and the unit norm descent directions $u \in \mathbb{R}^n$ and $v \in \mathbb{R}^m$ are the dominant left and right singular vectors of the dual variable $\mathbf{S} = \mathrm{Grad}_{\mathbf{X}} F$.

Additionally, the maximum decrease in the cost function in (5.1) is obtained for $\beta = (\sigma_1 - \lambda)/L_F$ where σ_1 is the dominant singular value of $\mathbf{S}$.

Proof. This is in fact a descent step as shown by Cai et al. (2010); Ma et al. (2011); Mazumder et al. (2010) but now projected onto the rank-one dominant subspace. The proof follows.

Since $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ is a stationary point for the problem (5.3) and not the global optimum of (5.1), by virtue of Proposition 5.2 we have $\|\mathbf{S}\|_{op} > \lambda$ (strict inequality). We assume that F is smooth and hence, let the first derivative of F is Lipschitz continuous with the Lipschitz constant L_F, i.e.,

Algorithm to solve convex problem (5.1)

0.
   - Initialize p to p_0, a rank guess.
   - Initialize the threshold ϵ for convergence criterion, refer to Proposition 5.2.
   - Initialize the iterates $(\mathbf{U}_0, \mathbf{B}_0, \mathbf{V}_0) \in \mathrm{St}(p_0, n) \times \mathrm{S}_{++}(p_0) \times \mathrm{St}(p_0, m)$.
1. Solve the fixed-rank problem (5.3) with rank p to obtain a critical point $(\mathbf{U}, \mathbf{B}, \mathbf{V})$.
2. Compute σ_1 (the dominant singular value) of dual variable $\mathbf{S} = \mathrm{Grad}_{\mathbf{X}} F$, where $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$.
   - If $\sigma_1 - \lambda \leq \epsilon$ (or duality gap $\leq \epsilon$) due to Proposition 5.2, output $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ as the solution to problem (5.1) and stop.
   - Else, compute the update as shown in Proposition 5.4 and compute the new point $(\mathbf{U}_+, \mathbf{B}_+, \mathbf{V}_+)$ as described in (5.23). Set $p = p + 1$ and repeat step 1.

Table 5.2: Algorithm to solve the trace norm minimization problem (5.1).

$\|\mathrm{Grad}_{\mathbf{X}} F - \mathrm{Grad}_{\mathbf{Y}} F\|_F \leq L_F \|\mathbf{X} - \mathbf{Y}\|_F$ for any $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times m}$ (Nesterov, 2003, Chapter 2). Therefore, the update (5.23), $\mathbf{X}_+ = \mathbf{X} - \beta u v^T$, results in the inequalities

$$\begin{array}{rlll} & F(\mathbf{X}_+) & \leq & F(\mathbf{X}) + \langle \mathrm{Grad}_{\mathbf{X}} F, \mathbf{X}_+ - \mathbf{X} \rangle + \frac{L_F}{2} \|\mathbf{X}_+ - \mathbf{X}\|_F^2 \\ & & = & F(\mathbf{X}) - \beta \sigma_1 + \frac{L_F}{2} \beta^2 (\text{from Lipschitz continuity}). \\ & \text{Also} & & \\ & \|\mathbf{X}_+\|_* & \leq & \|\mathbf{X}\|_* + \beta \text{ (from triangle inequality of matrix norm in (5.23))} \\ \Rightarrow & F(\mathbf{X}_+) + \lambda \|\mathbf{X}_+\|_* & \leq & F(\mathbf{X}) + \lambda \|\mathbf{X}\|_* - \beta(\sigma_1 - \lambda - \frac{L_F}{2} \beta) \end{array} \tag{5.24}$$

for $\beta > 0$ and σ_1 is the largest singular value of $\mathbf{S}$ ($= \mathrm{Grad}_{\mathbf{X}} F$). The maximum decrease in the cost function is obtained by maximizing $\beta(\sigma_1 - \lambda - \frac{L_f}{2} \beta)$ with respect to β which gives $\beta_{\max} = \frac{\sigma_1 - \lambda}{L_F} > 0$. In addition, $\beta_{\max} = 0 \Leftrightarrow \sigma_1 - \lambda = 0$ which characterizes global optimality as shown in Proposition (5.2). This proves the proposition.

□

A representation of $\mathbf{X}_+ = \mathbf{X} - \beta u v^T$ on $\mathcal{M}_{p+1}$ is obtained by computing the singular value decomposition of $\mathbf{X}_+$. Since $\mathbf{X}_+$ is a rank-one update of $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$, the singular value decomposition of $\mathbf{X}_+$ only requires $O(np^2 + mp^2 + p^3)$ operations (Brand, 2006). Finally, we perform a *backtracking* linesearch along the rank-one descent direction to compute a good value of β starting from the value $\frac{\sigma_1 - \lambda}{L_F}$, where L_F is the Lipschitz constant for the first-order derivative of F (Nesterov, 2003). The justification for this value is given in Proposition 5.4. In many problem instances, it is easy to estimate L_F by randomly selecting two points, say $\mathbf{X}$ and $\mathbf{Y} \in \mathbb{R}^{n \times m}$, and computing $\|\mathrm{Grad}_{\mathbf{X}} F - \mathrm{Grad}_{\mathbf{Y}} F\|_F / \|\mathbf{X} - \mathbf{Y}\|_F$ (Nesterov, 2003).

There is no theoretical guarantee that the algorithm in Table 5.2 stops at $p = p^*$ (the optimal rank). However, convergence to the global solution is guaranteed from the fact that the algorithm alternates between fixed-rank optimization and rank updates (unconstrained projected rank-1 gradient step) and both are descent iterates. Disregarding the fixed-rank step, the algorithm reduces to a gradient algorithm for a convex problem with classical global convergence guarantees. This theoretical certificate however does not capture the convergence properties of an algorithm that empirically always converges at a rank

Computing the regularization path of solutions

0. Given $\{\lambda_i\}_{i=1,\ldots,N}$ in decreasing order. Also given are the solutions $\mathbf{X}^*(\lambda_1)$ and $\mathbf{X}^*(\lambda_2)$ at λ_1 and λ_2 respectively and their low-rank factorizations.
1. Predictor step:
    - If $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$ belong to the same fixed-rank manifold $\mathcal{M}_p$, then construct a curve approximating the solution path at λ_i and compute the estimate $\hat{\mathbf{X}}(\lambda_{i+1})$ as shown in (5.26).
    - Else $\hat{\mathbf{X}}(\lambda_{i+1}) = \mathbf{X}^*(\lambda_i)$.
2. Corrector step: Using the estimated solution of the $\lambda_{i+1}-$problem, initialize the algorithm described in Table 5.2 to compute the exact solution $\mathbf{X}^*(\lambda_{i+1})$.
3. Repeat steps 1 and 2 for all subsequent values of λ.

Table 5.3: A scheme for computing the regularization path of solutions $\{\mathbf{X}^*(\lambda_i)\}$ for $i = \{1,\ldots,N\}$. If N is the number of values of λ and r is the number of rank increments, then the scheme uses r warm-restarts and $N-r$ predictor steps to compute the full path.

$p \ll \min(m,n)$ and most often at the optimal rank. One advantage of the scheme, in contrast to trace norm minimization algorithms proposed by Cai et al. (2010); Ma et al. (2011); Mazumder et al. (2010); Toh and Yun (2010), is that it offers a tighter control over the rank at all intermediate iterates of the scheme. It should be also be emphasized that the stopping criterion threshold of the fixed-rank problem (5.3) and of the convex problem (5.1) are chosen separately. This means that rank-increments can be made after a fixed number of iterations of the manifold optimization without waiting for the trust-region algorithm to converge to a critical point. Though not discussed here, the rank-one updating scheme can be extended to rank-k updating in a straightforward way.

5.5 Regularization path

In most applications the optimal value of λ is unknown (Mazumder et al., 2010) which means that in fact problem (5.1) should be solved for a number of regularization parameters. In addition, even if the optimal λ is known a priori, a *path* of solutions corresponding to different values of λ provides interpretability to the intermediate iterates which are now global minima for different values of λ. This motivates to compute the complete regularization path of (5.1) for a number of values λ, i.e., $\{\mathbf{X}^*(\lambda_i)\}$ for $i = \{1,\ldots,N\}$, where $\mathbf{X}^*(\lambda_i) = \underset{\mathbf{X}\in\mathbb{R}^{n\times m}}{\arg\min}\, F(\mathbf{X}) + \lambda_i\|\mathbf{X}\|_*$. The problem $\underset{\mathbf{X}\in\mathbb{R}^{n\times m}}{\arg\min}\, F(\mathbf{X}) + \lambda_i\|\mathbf{X}\|_*$ is referred to as the λ_i*-problem* in the subsequent sections.

A common approach to compute $\{\mathbf{X}^*(\lambda_i)\}$ for different regularization parameters is the *warm-restart* approach where the algorithm (any algorithm) to solve the λ_{i+1}-problem is initialized from $\mathbf{X}^*(\lambda_i)$ and so on (Mazumder et al., 2010). However, the warm-restart approach does not use the fact that a regularization path is *smooth*. An argument towards this is given later in the paragraph.

In this section, we propose a *predictor-corrector* scheme to compute a regularization path of solutions $\{\mathbf{X}^*(\lambda_i)\}$ efficiently for $i = \{1,\ldots,N\}$. We first take a *predictor* (estimator) step to predict the solution and then rectify the prediction by a *corrector* step. This scheme has been widely used in many regularization problems, e.g., regression problems (Park and Hastie, 2006).

The prediction step exploits the available information about solutions. For example, if $\mathbf{X}^*(\lambda_i)$ is the is the solution to the λ_i-problem, then the solution of the λ_{i+1} optimization problem is *estimated* from the solutions $\mathbf{X}^*(\lambda_i)$ and $\mathbf{X}^*(\lambda_{i-1})$ of the λ_{i-1}- and λ_i-problems, respectively. In the present context, we have the following two cases.

- $\mathbf{X}^*(\lambda_i)$ and $\mathbf{X}^*(\lambda_{i-1})$ have same rank: We exploit the Riemannian geometry of fixed-rank matrices, presented in Section 5.3, to construct interpolating curves that connect $\mathbf{X}^*(\lambda_i)$ and $\mathbf{X}^*(\lambda_{i-1})$ and use this information to compute an estimate $\hat{\mathbf{X}}^*(\lambda_{i+1})$ for the λ_{i+1}-problem on the fixed-rank manifold.
- $\mathbf{X}^*(\lambda_i)$ and $\mathbf{X}^*(\lambda_{i-1})$ have different ranks: In this scenario we resort to the standard warm-restart approach by assuming $\mathbf{X}^*(\lambda_i)$ to be an estimate of the solution of the λ_{i+1}-problem, i.e., $\hat{\mathbf{X}}^*(\lambda_{i+1}) = \mathbf{X}^*(\lambda_i)$.

The corrector step is subsequently carried out by initializing the algorithm in Table 5.2 from the predicted solution. The complete scheme is shown in Table 5.3 and has the following advantages.

- With a few number of rank increments we traverse the entire path of solutions $\{\mathbf{X}^*(\lambda_i)\}$ for $i = \{1, \ldots, N\}$.
- Potentially every iterate of the optimization scheme in Table 5.3 is a global solution for a value of the parameter λ.
- The predictor-corrector approach outperforms the warm-restart approach in maximizing *prediction accuracy* with minimal extra computations.

We also assume that the optimization problem (5.1) has a unique solution for each value of the parameter λ. A sufficient condition is that F is *strictly* convex, which can be enforced by regularizing F with square Frobenius norm of $\mathbf{X}$.

In order to characterize smoothness of a regularization path we observe that the global solution $\mathbf{X}^*(\lambda) = \mathbf{U}\mathbf{B}\mathbf{V}^T$, where $(\mathbf{U}, \mathbf{B}, \mathbf{V}) \in \mathcal{M}_p$ (the characterization of the fixed-rank factorization is presented in Table 5.1), is uniquely characterized by the nonlinear system of equations

$$\mathbf{S}\mathbf{V} \quad = \quad \lambda\mathbf{U}, \ \ \mathbf{U}^T\mathbf{S}\mathbf{V} = \lambda\mathbf{I}, \ \text{and} \ \ \mathbf{S}^T\mathbf{U} = \lambda\mathbf{V}$$

which is obtained from the optimality conditions (5.5) and Proposition 5.2. The smoothness of $\mathbf{X}^*(\lambda)$ with respect to λ follows from the implicit function theorem (Krantz and Parks, 2002). A geometrical reasoning is by inspection of the dual formulation of (5.1). It should be noted that we employ the predictor-corrector step *only* when we are on the fixed-rank manifold which corresponds to a *face* of the dual operator norm set. From Proposition 5.3, the dual optimal solution is obtained by projection onto the dual set. Smoothness of the dual variable, say $\mathbf{M}^*(\lambda)$, with respect to λ follows from the smoothness of the projection operator (Hiriart-Urruty and Lemaréchal, 1993). Consequently, smoothness of the primal variable $\mathbf{X}^*(\lambda)$ follows from the smoothness assumption of F.

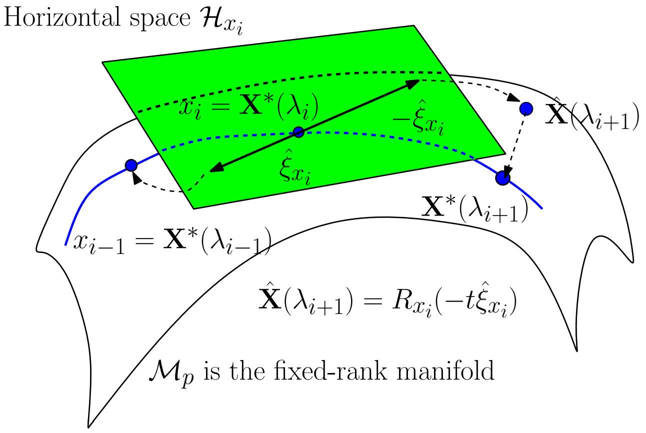

Horizontal space $\mathcal{H}_{x_i}$
$x_i = \mathbf{X}^*(\lambda_i)$
$-\hat{\xi}_{x_i}$
$\hat{\mathbf{X}}(\lambda_{i+1})$
$\hat{\xi}_{x_i}$
$\mathbf{X}^*(\lambda_{i+1})$
$x_{i-1} = \mathbf{X}^*(\lambda_{i-1})$
$\hat{\mathbf{X}}(\lambda_{i+1}) = R_{x_i}(-t\hat{\xi}_{x_i})$
$\mathcal{M}_p$ is the fixed-rank manifold

FIGURE 5.1: Tracing the path of solutions using a *predictor-corrector* approach on the fixed-rank manifold. The blue line denotes the curve of globally optimal solutions for problem (5.1) with different values for the parameter λ. The predicted solution $\hat{\mathbf{X}}^*(\lambda_{i+1})$ of the λ_{i+1}-problem is obtained by following an interpolating-curve connecting $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$. Finally, the solution $\mathbf{X}^*(\lambda_{i+1})$ is obtained from $\hat{\mathbf{X}}^*(\lambda_{i+1})$ with a corrector step that is the scheme in Table 5.2 with initialization from $\hat{\mathbf{X}}^*(\lambda_{i+1})$.

Predictor step on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ through interpolation

The basic idea of a predictor step for estimating the solutions for the λ_{i+1}-problem is to use information from the solutions for the λ_{i-1}- and λ_i-problems, where the λ_i-problem refers to $\operatorname*{arg\,min}_{\mathbf{X}\in\mathbb{R}^{n\times m}} F(\mathbf{X}) + \lambda_i\|\mathbf{X}\|_*$. We assume that both $x_i = (\mathbf{U}_i, \mathbf{B}_i, \mathbf{V}_i)$ and $x_{i-1} = (\mathbf{U}_{i-1}, \mathbf{B}_{i-1}, \mathbf{V}_{i-1})$ are in $\mathcal{M}_p$ are the matrix representatives of fixed-rank factorizations (the characterization is shown in Table 5.1) of solutions $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$ of the λ_{i-1}- and λ_i-problems, respectively. The standard approach in computing a predictor step is to follow the *geodesic* (the curve of the shortest length) connecting $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$ on fixed-rank manifold which corresponds to the abstract geodesic curve on the quotient manifold $\mathcal{M}_p/\mathcal{O}(p)$ connecting the equivalence classes $[x_{i-1}]$ and $[x_i]$ corresponding to $\mathbf{X}^*(\lambda_{i-1})$ and $\mathbf{X}^*(\lambda_i)$, respectively, where $[\cdot]$ is the equivalence class shown in Table 5.1. Computationally, the geodesic curve has a matrix representation in the total space $\mathcal{M}_p$. Equivalently, we identify a vector $\xi_{x_i} \in \mathcal{H}_{x_i}$ in the horizontal space $\mathcal{H}_{x_i}$ (the matrix characterization of the abstract tangent space at $[x_i]$) at x_i, defined as $\xi_{x_i} = \mathrm{Log}_{x_i}(x_{i-1})$ that maps the point $x_{i-1} \in \mathcal{M}_p$ to a vector in the horizontal space $\mathcal{H}_{x_i}$, where $\mathrm{Log}_{x_i}(\cdot) : \mathcal{M}_p \to \mathcal{H}_{x_i}$ is called the *logarithmic* mapping (inverse of the exponential map that constructs the geodesic curve) (Absil et al., 2008; Lee, 2003). As we deal with a retraction mapping (shown in Table 5.1) instead of the exponential mapping, the logarithmic mapping is relaxed to the *inverse* of the retraction mapping. However, it is not trivial to compute such a mapping as, first, it may not even be well-defined (except in the local neighborhood of x_i), and second, it may not be computationally tractable. For the case of interest there is no analytic expression for the logarithmic mapping. Instead, a numerically efficient way is to use an *approximate* inverse retraction $\hat{R}_{x_i}^{-1}(x_{i-1})$ (*locally* around x_i), where $\hat{R}_{x_i}^{-1} : \mathcal{M}_p \to \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$ to obtain a *direction* in the ambient space $\mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$. The direction so obtained is additionally *projected* onto the horizontal space $\mathcal{H}_{x_i}$ (using projection operators presented in Table 5.1). The *approximate* inverse retraction $\hat{R}_{x_i}^{-1}(x_{i-1})$ and an estimate on $\xi_{x_i} := \mathrm{Log}_{x_i}(x_{i-1})$ that we propose are

$$\begin{array}{l} \hat{R}_{x_i}^{-1}(x_{i-1}) = (\mathbf{U}_{i-1} - \mathbf{U}_i,\ \mathbf{B}_i^{\frac{1}{2}}\mathrm{logm}(\mathbf{B}_i^{-\frac{1}{2}}\mathbf{B}_{i-1}\mathbf{B}_i^{-\frac{1}{2}})\mathbf{B}_i^{\frac{1}{2}},\ \mathbf{V}_{i-1} - \mathbf{V}_i) \\ \text{and} \\ \hat{\xi}_{x_i} = \Pi_{x_i}(\Psi_{x_i}(\hat{R}_{x_i}^{-1}(x_{i-1}))), \end{array} \tag{5.25}$$

where the projection operators $\Psi_{x_i}(\cdot)$ and $\Pi_{x_i}(\cdot)$ are the projection operators defined in Table 5.1, $(\mathbf{U}_i, \mathbf{B}_i, \mathbf{V}_i)$ and $(\mathbf{U}_{i-1}, \mathbf{B}_{i-1}, \mathbf{V}_{i-1})$ are the matrix representations of x_i and x_{i-1}, respectively, and $\mathrm{logm}(\cdot)$ is the *matrix logarithm* operator. The computations of the estimate search direction $\hat{\xi}_{x_i}$ cost $O(np^2+mp^2+p^3)$. We do not comment about the *goodness* of the estimate $\hat{\xi}_{x_i}$ with respect to the $\mathrm{Log}_{x_i}(x_{i-1})$ except to point out that the error between the two converges to zero as x_{i-1} tends to x_i.

Subsequently, the predicted solution for the λ_{i+1}-problem, i.e., $\arg\min_{\mathbf{X}\in\mathbb{R}^{n\times m}} F(\mathbf{X}) + \lambda_{i+1}\|\mathbf{X}\|_*$, is obtained by taking a step $t > 0$ and performing a backtracking linesearch along the direction $-\hat{\xi}_{x_i}$, i.e.,

$$\hat{\mathbf{X}}(\lambda_{i+1}) = R_{x_i}(-t\hat{\xi}_{x_i}), \tag{5.26}$$

where $\hat{\xi}_{x_i}$ is the horizontal vector computed in (5.25) and $R_{x_i}(\cdot)$ is the retraction operator (to compute an iterate on the manifold $\mathcal{M}_p$) is presented in Table 5.1. It should be emphasized that we move along the *negative* of the search direction $\hat{\xi}_{x_i}$ obtained in (5.25). A good choice of the initial step-size t is $(\lambda_{i+1} - \lambda_i)/(\lambda_i - \lambda_{i-1})$. The motivation for this choice comes the observation that it is optimal when the solution path is a straight line in the Euclidean space. The numerical complexity to perform the prediction step in the manifold $\mathcal{M}_p$ is $O(np^2 + mp^2 + p^3)$.

5.6 Numerical Experiments

The overall optimization scheme with *descent-restart* and trust-region algorithm for the fixed-rank optimization problem is denoted as “Descent-restart + TR” (TR). We test the proposed optimization framework on the problems of low-rank matrix completion and multivariate linear regression where trace norm penalization has shown efficient recovery. Full regularization paths are constructed with optimality certificates. All simulations in this section have been performed in MATLAB on a 2.53 GHz Intel Core i5 machine with 4 GB of RAM. Our matrix completion implementation may be downloaded from `http://www.montefiore.ulg.ac.be/~mishra/codes/traceNorm.html`.

5.6.1 Diagonal versus matrix scaling

Before entering a detailed numerical experiment we illustrate here the empirical evidence that constraining $\mathbf{B}$ in the factorization (5.2) to be diagonal (as is the case with SVD) is detrimental to optimization. To this end, we consider the simplest implementation of a gradient descent algorithm for matrix completion problem (see below). The plots shown Figure 5.2 compare the behavior of the same algorithm in the search space $\mathrm{St}(p,n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p,m)$ and $\mathrm{St}(p,n) \times \mathrm{Diag}_{++}(p) \times \mathrm{St}(p,m)$ (SVD). $\mathrm{Diag}_{++}(p)$ is the set of diagonal matrices with positive entries. The empirical observation that convergence suffers

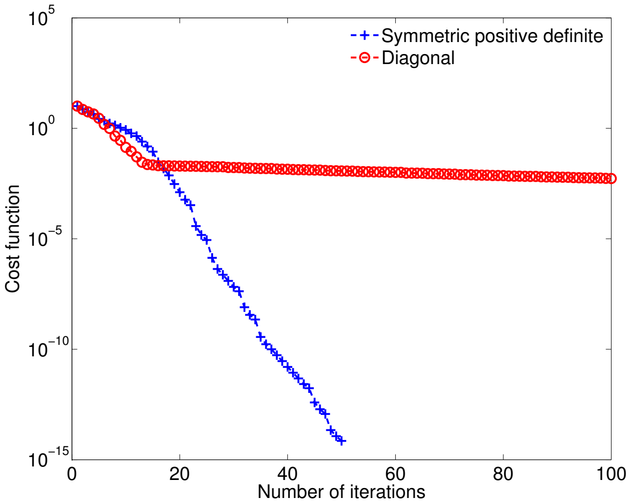


Figure 5.2: Convergence of a gradient descent algorithm is affected by making $\mathbf{B}$ in the factorization (5.2) diagonal.

from imposing diagonal structure on $\mathbf{B}$ is a generic observation that does not depend on the particular problem at hand. The problem here involves completing a 200×200 of rank 5 from 40% of observed entries. λ is fixed at 10^{-10}.

5.6.2 Low-rank matrix completion

The problem of matrix completion involves completing an $n \times m$ matrix when only a few entries of the matrix entries are known. Given an incomplete low-rank (but unknown rank) $n \times m$ real matrix $\widetilde{\mathbf{X}}$, a *convex relaxation* of the matrix completion problem is

$$\min_{\mathbf{X} \in \mathbb{R}^{n \times m}} \quad \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2 + \lambda \|\mathbf{X}\|_* \tag{5.27}$$

for $\mathbf{X} \in \mathbb{R}^{n \times m}$ and a regularization parameter $\lambda \in \mathbb{R}_+$. Here $\|\cdot\|_F$ denotes the Frobenius norm, matrix $\mathbf{W}$ is an $n \times m$ *weight* matrix with binary entries and the operator $\odot$ denotes element-wise multiplication. If $\mathcal{W}$ is the set of known entries in $\widetilde{\mathbf{X}}$ then, $\mathbf{W}_{ij} = 1$ if $(i, j) \in \mathcal{W}$ and $\mathbf{W}_{ij} = 0$ otherwise. The problem of matrix completion is known to be combinatorially hard. However, by solving the convex relaxation (5.27) a low-rank reconstruction is possible with a very high probability under Gaussian distribution of the observed entries (Candès and Plan, 2009; Keshavan et al., 2010). For an exact reconstruction, the lower bound on the number of known entries $|\mathcal{W}|$ is typically of the order $O(nr + mr)$ where r is the optimal rank, $|\mathcal{W}| > \max(n, m) \gg r$. Consequently, it leads to a very *sparse weight* matrix $\mathbf{W}$, which plays a very crucial role (along with the least-squares of the cost function) for efficient algorithmic implementations including the computation of the duality gap expression. For our case, we assume that the lower bound on the number of entries is met and we seek a solution to the optimization problem (5.27). Customizing the terminology for the present problem, the convex function $F : \mathbb{R}^{n \times m} \to \mathbb{R}$ is $F(\mathbf{X}) = \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2$. Using the factorization $\mathbf{X} = \mathbf{U}\mathbf{B}\mathbf{V}^T$ of (5.2), the rank-p cost function

$f : \mathcal{M}_p \rightarrow \mathbb{R}$ is $f(\mathbf{U}, \mathbf{B}, \mathbf{V}) = \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{U}\mathbf{B}\mathbf{V}^T)\|_F^2 + \lambda \mathrm{Trace}(\mathbf{B})$, where $(\mathbf{U}, \mathbf{B}, \mathbf{V}) \in \mathcal{M}_p$. The dual variable for the problem (5.27) is $\mathbf{S} = 2(\mathbf{W} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T - \widetilde{\mathbf{X}}))$.

For the fixed-rank problem, the Riemannian gradient and the Riemannian Hessian (applied along a search direction) are computed directly using formulas (5.17) and (5.20). Specifically, we require matrix representation of the first-order derivative $f_x(x)$ of f in $\mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$ which is

$$f_x(x) = (f_{\mathbf{U}}, f_{\mathbf{B}}, f_{\mathbf{V}}) = (\mathbf{S}\mathbf{V}\mathbf{B}, \mathbf{U}^T\mathbf{S}\mathbf{V} + \lambda\mathbf{I}, \mathbf{S}^T\mathbf{U}\mathbf{B}),$$

where $\mathbf{S} = 2(\mathbf{W} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T - \widetilde{\mathbf{X}}))$ and $(\mathbf{U}, \mathbf{B}, \mathbf{V})$ is the matrix representation of $x \in \mathcal{M}_p$. Apart from the matrix representation of $f_x(x)$, we also require a matrix representation of the Euclidean directional derivative of $f_x(x)$ along $(\mathbf{Z}_{\mathbf{U}}, \mathbf{Z}_{\mathbf{B}}, \mathbf{Z}_{\mathbf{V}}) \in \mathbb{R}^{n\times p}\times\mathbb{R}^{p\times p}\times\mathbb{R}^{m\times p}$, that is, $\mathrm{D}f_x(x)[(\mathbf{Z}_{\mathbf{U}}, \mathbf{Z}_{\mathbf{B}}, \mathbf{Z}_{\mathbf{V}})] = (\mathbf{S}\mathbf{V}\mathbf{Z}_{\mathbf{B}} + \mathbf{S}\mathbf{Z}_{\mathbf{V}}\mathbf{B} + \mathbf{S}_*\mathbf{V}\mathbf{B}, \mathbf{Z}_{\mathbf{U}}^T\mathbf{S}\mathbf{V} + \mathbf{U}\mathbf{S}\mathbf{Z}_{\mathbf{V}} + \mathbf{U}^T\mathbf{S}_*\mathbf{V}, \mathbf{S}^T\mathbf{U}\mathbf{Z}_{\mathbf{B}} + \mathbf{S}^T\mathbf{Z}_{\mathbf{U}}\mathbf{B} + \mathbf{S}_*^T\mathbf{U}\mathbf{B})$, where the auxiliary variable $\mathbf{S}_* = \mathrm{D}\mathbf{S}[(\mathbf{Z}_{\mathbf{U}}, \mathbf{Z}_{\mathbf{B}}, \mathbf{Z}_{\mathbf{V}})] = 2(\mathbf{W} \odot (\mathbf{Z}_{\mathbf{U}}\mathbf{B}\mathbf{V}^T + \mathbf{U}\mathbf{Z}_{\mathbf{B}}\mathbf{V}^T + \mathbf{U}\mathbf{B}\mathbf{Z}_{\mathbf{V}}^T))$ is the directional derivative of the dual variable $\mathbf{S}$ along $(\mathbf{Z}_{\mathbf{U}}, \mathbf{Z}_{\mathbf{B}}, \mathbf{Z}_{\mathbf{V}}) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p} \times \mathbb{R}^{m\times p}$.

It should be noted that as $\mathbf{W}$ is sparse, the matrices $\mathbf{S}$ and $\mathbf{S}_*$ are sparse too.Consequently, the computational complexity per iteration for the trust-region algorithm is of order $O(|\mathcal{W}|p + np^2 + mp^2 + p^3)$, where $|\mathcal{W}|$ is the number of known entries. In addition, computation of the dominant singular value and vectors of $\mathbf{S}$ for the rank-one updating step (5.23) for the algorithm in Table 5.2 is done is using few iterations of the *power iteration* algorithm (Golub and Van Loan, 1996, Chapter 8) with a cost $O(|\mathcal{W}|)$ (Larsen, 2004), thereby potentially allowing to handle large datasets.

5.6.2.1 Fenchel dual and duality gap computation for matrix completion

From Proposition 5.3, the Fenchel conjugate F^* of F admits the expression $F^*(\mathbf{M}) = \mathrm{Trace}(\mathbf{M}^T\mathbf{M})/4 + \mathrm{Trace}(\mathbf{M}^T(\mathbf{W} \odot \widetilde{\mathbf{X}}))$, where the domain of F^* is the non-zero support of $\mathbf{W}$. The Fenchel conjugate computation exploits the least-squares nature of the function F. The duality gap expression for a dual candidate $\mathbf{M} = \min(1, \frac{\lambda}{\sigma_1})\mathbf{S}$ is

$$F(\mathbf{X}) + \lambda\|\mathbf{X}\|_* + \mathrm{Trace}(\mathbf{M}^T\mathbf{M})/4 + \mathrm{Trace}(\mathbf{M}^T(\mathbf{W} \odot \widetilde{\mathbf{X}})), \tag{5.28}$$

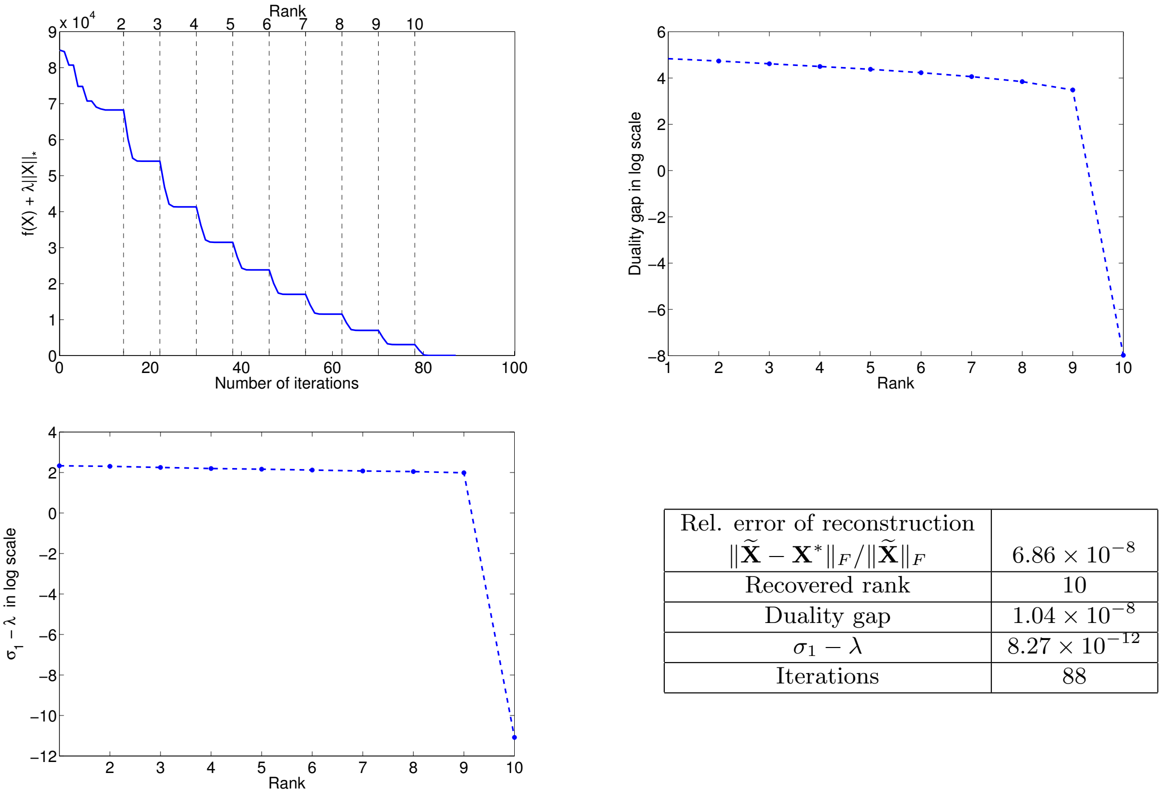
where σ_1 is the dominant singular value of $\mathbf{S} = 2(\mathbf{W} \odot (\mathbf{U}\mathbf{B}\mathbf{V}^T - \widetilde{\mathbf{X}}))$ and $\mathbf{X}$ admits the factorization $\mathbf{U}\mathbf{B}\mathbf{V}^T$ (5.27). It should be stressed that, thanks to the fixed-rank matrix factorization, the duality gap computation (5.28) can be accomplished efficiently.

5.6.2.2 Simulations

Next we provide some benchmark simulations for the low-rank matrix completion problem. For each example, an $n \times m$ random matrix of rank p is generated as proposed by Cai et al. (2010). Two matrices $\mathbf{A} \in \mathbb{R}^{n\times p}$ and $\mathbf{B} \in \mathbb{R}^{m\times p}$ are generated according to a Gaussian distribution with zero mean and unit standard deviation. The matrix product $\mathbf{A}\mathbf{B}^T$ gives a *random* matrix of rank p. A fraction of the entries are randomly removed with uniform probability. The dimension of rank-p matrices of size $n \times m$ is $(n + m - p)p$ and the over-sampling (OS) ratio determines the number of entries that are known as a multiple of the dimension. A OS $= 6$ implies that $6(n+m-p)p$ number of randomly and uniformly selected entries are known a priori out of nm entries.

Example 1: for fixed λ

A 100×100 random matrix of rank 10 is generated as mentioned above. 20% (OS $= 4.2$) of the entries are randomly removed with uniform probability. To reconstruct the original matrix we run the optimization scheme proposed in the Table 5.2 along with the trust-region algorithm to solve the fixed-rank problem. For illustration purposes λ is fixed at 10^{-5}. We also assume that we do not have any a priori knowledge of the optimal rank and, thus, start from rank 1. The trust-region algorithm stops when the relative or absolute variation of the cost function is below 10^{-10}. The rank-incrementing strategy stops when relative duality gap is less than 10^{-5}, i.e., $\frac{F(\mathbf{X})+\lambda\|\mathbf{X}\|_* + F^*(\mathbf{M})}{|F^*(\mathbf{M})|} \leq 10^{-5}$. Convergence plots of the scheme are shown in Figure 5.3. A good way to characterize matrix reconstruction at $\mathbf{X}$ is to look at the relative error of reconstruction, defined as,



Rel. error of reconstruction $\|\widetilde{\mathbf{X}} - \mathbf{X}^*\|_F / \|\widetilde{\mathbf{X}}\|_F$	6.86×10^{-8}
Recovered rank	10
Duality gap	1.04×10^{-8}
$\sigma_1 - \lambda$	8.27×10^{-12}
Iterations	88

FIGURE 5.3: Matrix completion by trace norm minimization algorithm with $\lambda = 10^{-5}$. Upper left: rank incremental strategy with descent directions. Upper right: optimality certificate of the solution with duality gap. Lower left: convergence to the global solution according to Proposition 5.2 . Lower right: recovery of the original low-rank matrix.

$$\text{relative error of reconstruction} = \|\widetilde{\mathbf{X}} - \mathbf{X}\|_F / \|\widetilde{\mathbf{X}}\|_F.$$

Next, to understand low-rank matrix reconstruction by trace norm minimization we repeat the experiment for a number of values of λ all initialized from the same starting point and report the relative

λ	10	10^{-2}	10^{-5}	10^{-8}
Rel. reconstruction error	6.33×10^{-2}	7.42×10^{-5}	7.11×10^{-8}	6.89×10^{-11}
Recovered rank	10	10	10	10
Iterations	113	120	119	123
Time in seconds	2.7	2.8	2.9	2.9

Table 5.4: Efficacy of trace norm penalization to reconstruct low-rank matrices by solving (5.27).

reconstruction error in Table 5.4 averaged over five runs. This, indeed, confirms that matrix reconstruction is possible by solving the trace norm minimization problem (5.27).

Example 2: regularization path for matrix completion

In order to compute a regularization path of solutions corresponding to different values of λ, we employ the predictor-corrector approach described in Table 5.3 to find solutions for a grid of λ values. For the purpose of illustration, a geometric sequence of λ values is created with the maximum value fixed at $\lambda_1 = 10^3$, the minimum value is set at $\lambda_N = 10^{-3}$ and a reduction factor $\gamma = 0.95$ such that $\lambda_{i+1} = \gamma\lambda_i$. We consider the example that has been proposed previously. The algorithm for a $\lambda_i \in \{\lambda_1, \ldots, \lambda_N\}$ stops when the relative duality gap is less than 10^{-5}. Various plots are shown in Figure 5.4. Figure 5.4 also demonstrates the advantage of the scheme in Table 5.3 with respect to a warm-restart approach. We compare both approaches on the basis of

$$\text{Inaccuracy in prediction} = f(\hat{\mathbf{X}}(\lambda_i)) - f(\mathbf{X}^*(\lambda_i)) \tag{5.29}$$

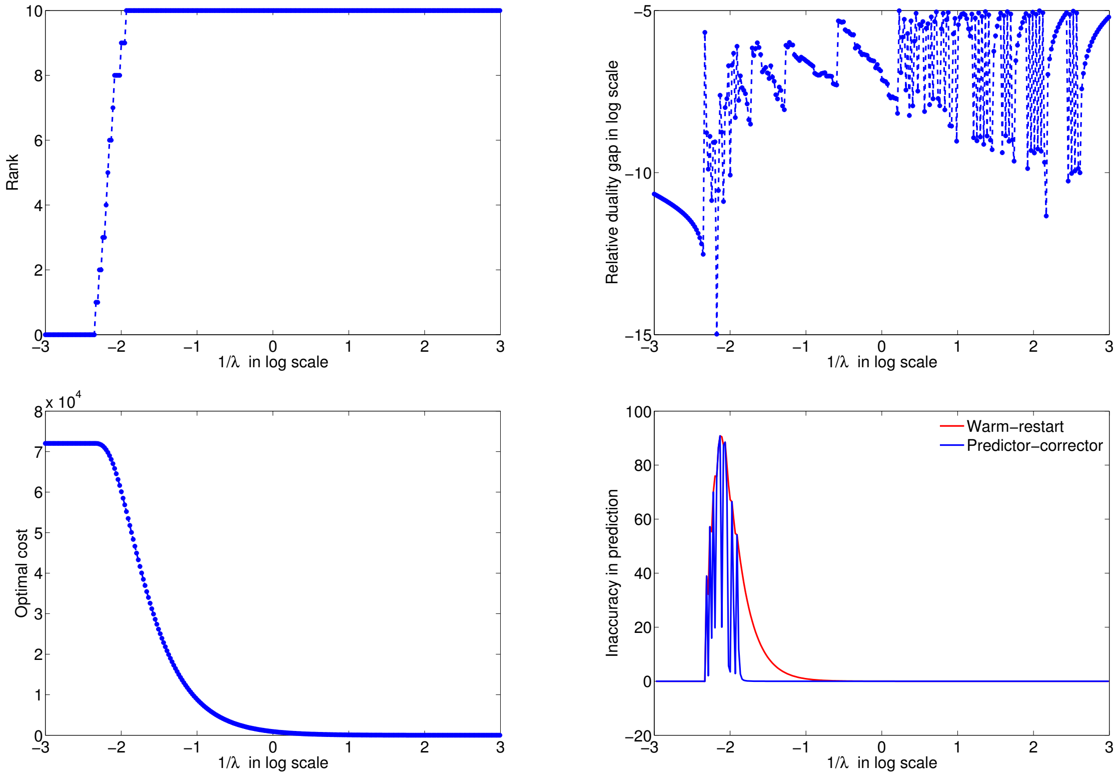
where $\mathbf{X}^*(\lambda_i)$ is the global minimum at λ_i and $\hat{\mathbf{X}}(\lambda_i)$ is the prediction. A lower inaccuracy means better prediction. It should be emphasized that in Figure 5.3 most of the points on the curve of the cost function have no other utility than being intermediate iterates towards the global solution of the algorithm. In contrast all points of the curve of optimal cost values in Figure 5.4 are now global minima for different values of λ.

Example 3: competing methods for matrix completion

In this section, we analyze the following state-of-the-art algorithms for trace norm regularized low-rank matrix completion, namely,

1. SVT algorithm by Cai et al. (2010),
2. FPCA algorithm by Ma et al. (2011),
3. SOFT-IMPUTE (Soft-I) algorithm by Mazumder et al. (2010), and
4. APG and APGL algorithms by Toh and Yun (2010).

For our simulation studies we use the MATLAB codes supplied on the authors' webpages for SVT, FPCA, and APGL. Due to simplicity of the SOFT-IMPUTE algorithm we use our own MATLAB implementation. The numerically expensive step in all these algorithms is the computation of the *singular value thresholding* operation that forms the core of these algorithms. To reduce the computational burden



# λ values	270
# iterations	766
Time	38.60 seconds

Figure 5.4: Computation of a regularization path using Descent-restart + TR with a predictor-corrector approach. Upper left: recovery of solutions of all ranks. Upper right: optimality certificate for each solution on the regularization path. Lower left: path traced by the algorithm. Lower right: better prediction by the algorithm in Table 5.3 than a pure warm-restart approach. Table: number of iterations per value of λ is < 3.

FPCA uses a linear time approximate singular value decomposition (SVD). Likewise, implementations of SVT, SOFT-IMPUTE and APGL exploit the *low-rank + sparse* structure of the iterates to optimize the thresholding operation (Larsen, 2004).

The basic algorithm FPCA by Ma et al. (2011) is a fixed-point algorithm with a proven bound on the iterations for convergence to the ϵ-accuracy ball, i.e., the error with respect to the actual solution is bounded by ϵ. To accelerate the convergence they use the technique of *continuation* that involves approximately solving a decreasing sequence of values of λ leading to the target value of λ. The singular value thresholding burden step is carried out by a *linear time* approximate singular value decomposition that has shown superior performance.

The basic algorithm APG of Toh and Yun (2010) is a *proximal method* (Nesterov, 2003) and gives a much stronger bound, precisely $O(1/\sqrt{\epsilon})$, on the number of iterations to converge with ϵ-accuracy. To accelerate the scheme, the authors propose three additional heuristics: *continuation*, *truncation* (hard-thresholding of ranks by projecting onto the set of fixed-rank matrices), and *linesearch technique* for estimating the

Lipschitz constant (of the first-order derivative of the function F). The accelerated version is called APGL.

The basic algorithm SOFT-IMPUTE of Mazumder et al. (2010) iteratively replaces the missing elements with those given by an approximate SVD thresholding (of a sparse + low-rank matrix) at each iteration. Accelerated versions involve post processing like continuation and truncation of singular values to obtain a low-rank solution. It should be emphasized that the performance of SOFT-IMPUTE greatly varies with the singular values computation at each iteration. For our simulations we compute 20 dominant singular values at each iteration of SOFT-IMPUTE.

While FPCA, SOFT-IMPUTE, and APGL solve the problem formulation (5.27), the iterates of the SVT algorithm converge towards a solution of the optimization problem that minimizes $\tau\|\mathbf{X}\|_* + \|\mathbf{X}\|_F^2/2$ subject to the constraint that the entries of $\mathbf{X}$ belonging to the set agree with the known entries of the incomplete matrix $\widetilde{\mathbf{X}}$, i.e., $\mathbf{W} \odot \mathbf{X} = \mathbf{W} \odot \widetilde{\mathbf{X}}$, $\tau > 0$ is the regularization parameter for SVT.

Convergence behavior of different algorithms with varying λ. In the current section we analyze the algorithms FPCA, SOFT-IMPUTE, APG, and Descent-restart + TR regarding their ability to solve (5.27) for a fixed value of λ. For this simulation, we use FPCA, SOFT-IMPUTE, and APG *without* any acceleration techniques like continuation and truncation. SVT is not used for this test since it deals with a different cost function. We plot the cost function $F(\mathbf{X}) + \lambda\|\mathbf{X}\|_*$ against the number of iterations for a number of values of the parameter λ. A 100×100 random matrix of rank 5 is generated under standard assumptions with over-sampling ratio OS $= 4$ (61% of entries are removed uniformly). The algorithms Descent-restart + TR, FPCA, SOFT-IMPUTE, and APG are initialized similarly. The algorithms are stopped when either the absolute variation or relative variation of the cost function $F(\mathbf{X}) + \lambda\|\mathbf{X}\|_*$ is less than 10^{-10}. The maximum number of iterations is set at 500. The rank-one updating procedure of our algorithm is stopped when the relative duality gap is less than 10^{-5}.

The plots in Figure 5.5 show convergence behavior of the considered algorithm for four different values of λ. The convergence behavior of FPCA is greatly affected by the value of the parameter λ. It has *slower* convergence for a smaller λ. For a larger value of λ, FPCA shows a fluctuating behavior. SOFT-IMPUTE shows a better convergence in all the cases. However, its convergence suffers when a higher accuracy is sought. The performance of APG is robust to the change in values of λ. For a moderate accuracy, it outperforms all other algorithms. However, when a higher accuracy is sought it requires a significantly higher number of iterations. Descent-restart + TR, on the other hand, outperforms others in all the cases here with minimal number of iterations.

Convergence of data fitting error for different algorithms. To understand the convergence behavior of different algorithms involving different optimization problems, we look at the evolution of the error in data fitting or training error (Cai et al., 2010; Mazumder et al., 2010) defined as

$$\text{training error} = \|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2 \tag{5.30}$$

with iterations. Here $\widetilde{\mathbf{X}}$ is the incomplete matrix that we seek to complete, $\mathbf{W}$ the binary weight matrix of zeros and ones, and $\odot$ denotes element-wise multiplication of matrices. We generate a 150×300 random matrix of rank 10 under standard assumptions with OS $= 5$. The algorithms Descent-restart + TR, FPCA, SOFT-IMPUTE (Soft-I), and APG are initialized similarly. λ is fixed to 10^{-5} as it results

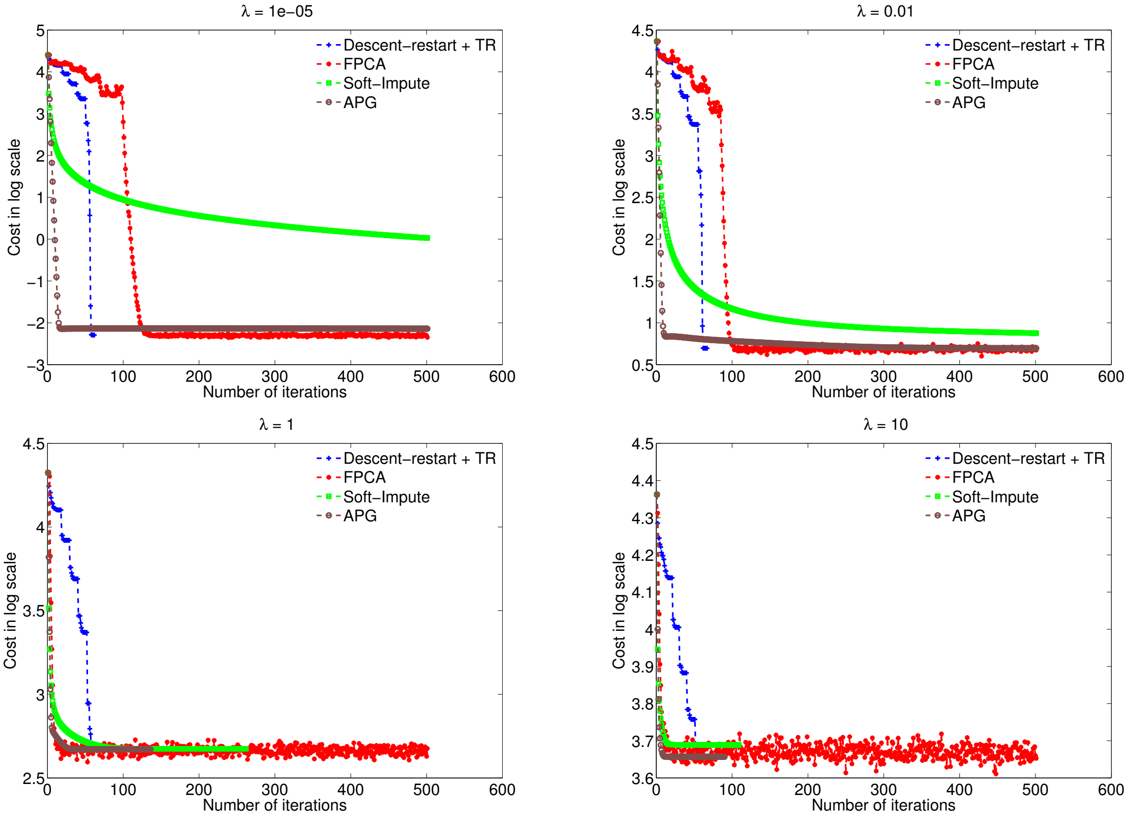


FIGURE 5.5: Convergence behavior of different algorithms for different values of λ. The algorithms compared here do not use any acceleration heuristics.

in a good reconstruction of the incomplete matrix. For SVT we use the default values of τ and step-size as suggested by Cai et al. (2010). The algorithms are stopped when the variation or relative variation of training error (5.30) is less than 10^{-10}. The maximum number of iterations is set at 500. The rank-one updating procedure of our algorithm is stopped when the relative duality gap is below 10^{-5}.

In Figure 5.6 APG has a fast convergence but the performance slows down later. Consequently, it exceeds the maximum limit of iterations. Similarly, SOFT-IMPUTE converges to a different solution but has a faster convergence in the initial phase (for iterations less than 60). FPCA and Descent-restart + TR converge faster at a later stage of their iterations. Descent-restart + TR initially sweeps through ranks until arriving at the optimal rank where the convergence is accelerated owing to the trust-region algorithm.

Scaling test. To analyze the scalability of these algorithms to larger problems we perform a test where we vary the problem size n from 200 to 2200. The reason for choosing a moderate value of n is that large-scale implementations of SVT, FPCA, and Soft-Impute are unavailable from their respective authors' webpages. For each n, we generate a random matrix of size $n \times n$ of rank 10 under standard assumptions with different over-sampling ratios (OS). The initializations are chosen as in the earlier example i.e., $\lambda = 10^{-5}$. We note the time and number of iterations taken by the algorithms until the stopping criterion is satisfied or when the number of iterations exceed 500. The stopping criterion is same as the one used before for comparison, when the absolute variation or relative variation of training error (5.30) is less than 10^{-10}. Results averaged over five runs are shown in Figure 5.7. We have not

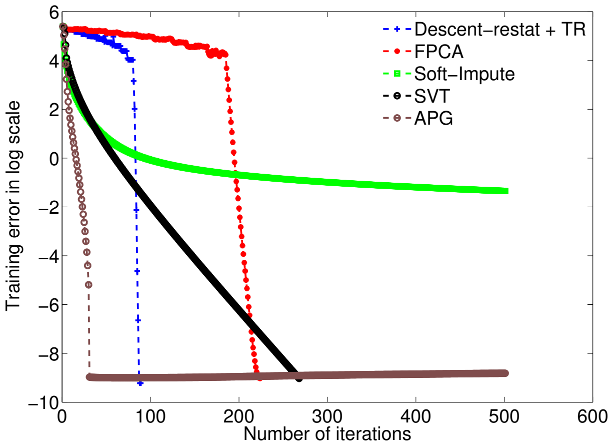
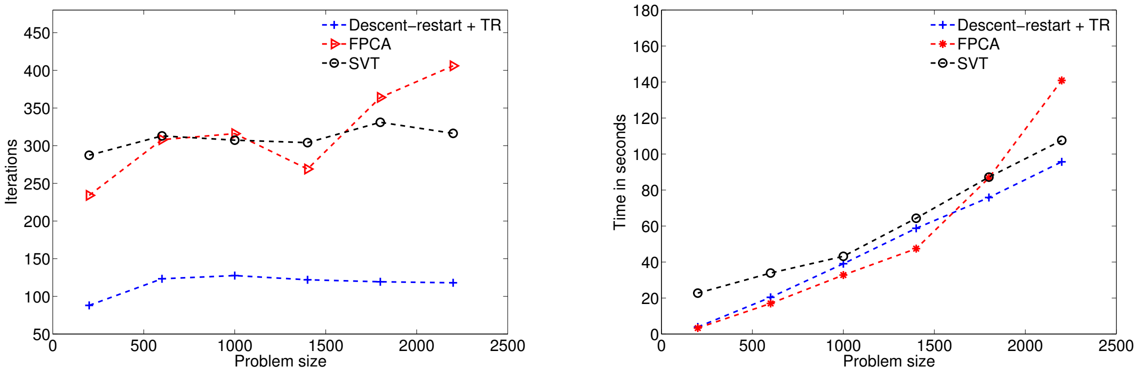


FIGURE 5.6: Convergence behavior of different algorithms for minimizing the training error (5.30).



n	200	600	1000	1400	1800	2200
$\vert\mathcal{W}\vert$	18409	66676	120000	176184	234380	294134
F	0.46	0.19	0.12	0.09	0.07	0.06
OS	4.7	5.6	6.0	6.3	6.5	6.7

FIGURE 5.7: Analysis of the algorithms on randomly generated datasets of rank 10 with varying fractions of missing entries. SVT, FPCA and Descent-restart + TR have similar performances but Descent-restart + TR usually outperforms others.

shown the plots for SOFT-IMPUTE and APG as in all the cases either they did not converge in 500 iterations or took much more time than the nearest competitor.

Below we have shown two more case studies where we intend to show numerical scalability of our algorithm to large-scale instances. The first one involves comparisons with fixed-rank optimization algorithms. The second case is a large-scale comparison with APGL (the accelerated version of APG). We consider the problem of completing a 50000×50000 matrix $\widetilde{\mathbf{X}}$ of rank 5. The over-sampling ratio OS is 8 implying that 0.16% (3.99×10^6) of entries are randomly and uniformly revealed. The maximum number of iterations is fixed at 500.

Fixed-rank comparison. Because our algorithm uses a fixed-rank approach, it is meaningful to compare its performance with other fixed-rank optimization algorithms. However, a rigorous comparison with other algorithms is beyond the scope of the present chapter. Here we compare with two set-of-the-art

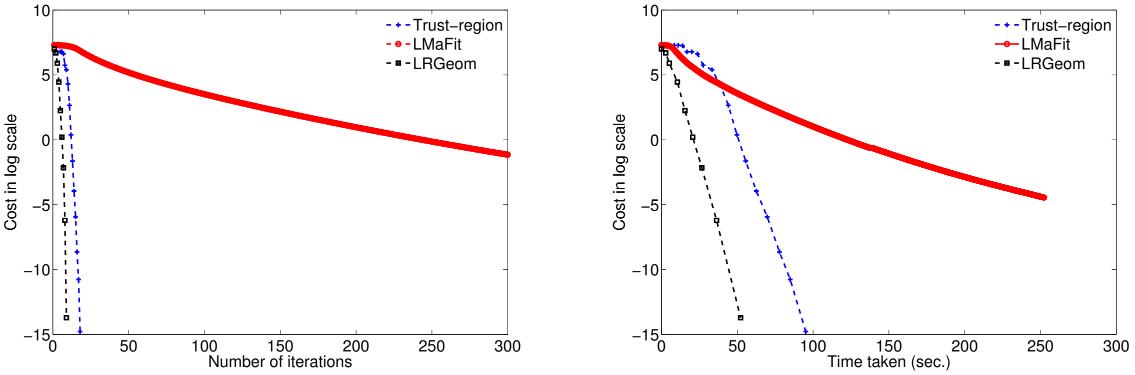


FIGURE 5.8: Rank 5 completion of 50000×50000 matrix with OS = 8. All the algorithms are initialized by taking 5 dominant SVD of sparse $\widetilde{\mathbf{X}}$ as proposed by Keshavan et al. (2010). Algorithms are stopped when the cost function below a threshold, $\|\mathbf{W} \odot (\widetilde{\mathbf{X}} - \mathbf{X})\|_F^2 \leq 10^{-10}$. The proposed trust-region scheme is competitive with LMaFit for large-scale problems. Although LMaFit has a smaller time complexity per iteration but its convergence seems to suffer for large-scale problems. With respect to LRGeom, the performance is poorer although both have a similar asymptotic rate of convergence.

algorithms that are LMaFit (Wen et al., 2012) and LRGeom (trust-region implementation) (Vandereycken, 2013). LMaFit is an alternating minimization scheme with a different factorization for a fixed-rank matrix. We use the fixed-rank implementation of LMaFit. It is a tuned-version of the Gauss-Seidel non-linear scheme and has a smaller time complexity per iteration. LRGeom is based on the embedded geometry of fixed-rank matrices. This viewpoint allows to simplify notions of moving on the search space. We use their trust-region implementation. The geometry leads to efficient guess of the optimal step-size in a search direction. Figure 5.8 shows a competitive performance of our trust-region scheme with respect to LMaFit. Asymptotically, both our trust-region scheme and LRGeom perform similarly with LRGeom performing better in the initial phase.

Comparison with APGL. APG has a better iteration complexity than other optimization algorithms. However, scalability of APG by itself to larger dimensional problems is an issue. The principal bottleneck is that the ranks of the intermediate iterates seem to be uncontrolled and only asymptotically, a low-rank solution is expected. To circumvent this issue, an accelerated version of APG called APGL is also proposed (Toh and Yun, 2010). APGL is APG with three additional heuristics: *continuation* (a sequence of parameters leading to the target λ), *truncation* (hard-thresholding of ranks by projecting onto fixed-rank matrices) and linesearch technique for estimating the Lipschitz constant L_F for the first derivative of the cost function. We compare our algorithm with APGL. The algorithms are stopped when either absolute variation or relative variation of the cost function is less than 10^{-10}. For our algorithm, the trust-region algorithm is also terminated with the same criterion. In addition, the rank-one updating is stopped when the relative duality gap is below 10^{-5}.

For a fixed $\lambda = \lambda$, APGL proceeds through a sequence of values for λ such that $\lambda_k = \max\{0.7\lambda_{k-1}, \lambda\}$ where k is the iteration count of the algorithm. Initial λ_0 is set to $2\|\mathbf{W} \odot \widetilde{\mathbf{X}}\|_{op}$. We also follow a similar approach and create a sequence of values. A decreasing sequence is generated leading to λ is by using the recursive rule, $\lambda_i = \lambda_{i-1}/2$ when $\lambda_{i-1} > 1$ and $\lambda_i = \lambda_{i-1}/100$ otherwise until $\lambda_{i-1} < \lambda$. Initial λ_0 is set to $\|\mathbf{W} \odot \widetilde{\mathbf{X}}\|_{op}$. For $\lambda_i \neq \lambda$ we also relax the stopping criterion for the trust-region algorithm (for the

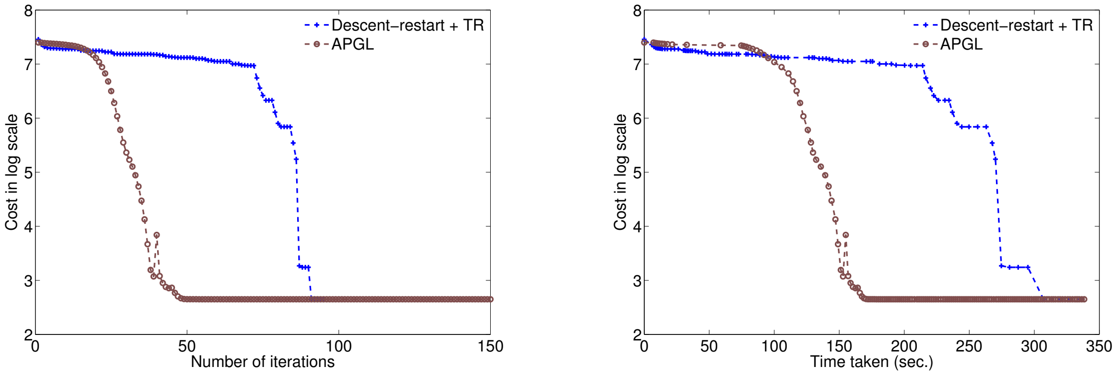


FIGURE 5.9: A large-scale instance of rank 5 completion of 50000 × 50000 matrix with OS = 8. $\lambda = 2\|\mathbf{W} \odot \widetilde{\mathbf{X}}\|_{op}/10^5$ as suggested by Toh and Yun (2010). The proposed framework is competent for very low-ranks and when a high accuracy is sought. However, we spend a considerable time in just traversing through ranks before arriving at the optimal rank.

fixed-rank subproblem) to 10^{-5} as well as stopping the rank-one increment when relative duality gap is below 1 as we are only interested for an accurate solution for $\lambda = \lambda$.

In Figure 5.9 we compete favorably with APGL in large-scale problems for very low-ranks and when a higher accuracy is required. However, as the rank increases, APGL performs better. This is not surprising as our algorithm traverses all ranks, one by one before arriving at the optimal rank. In the process we spend a considerable effort in traversing ranks. This approach is most effective only when computing in the entire regularization path. Also for moderate accuracy, APGL performs extremely well. However, the better performance of APGL significantly relies on heuristics like continuation and truncation. The truncation heuristic allows the APGL algorithm to approximate an iterate by low and fixed-rank iterate. On the other hand, we strictly move in the low-rank space. Exploiting this leads to an efficient way for computing the entire regularization path using a predictor-corrector strategy of Section 5.5.

Comments on competing matrix completion algorithms. We summarize our observations in the following points.

- The convergence rate of SOFT-IMPUTE is greatly dependent on the computation of singular values. For large-scale problems this is a bottleneck and the performance is greatly affected. However, in our experiments, it performs quite well within a reasonable accuracy as seen in Figure 5.5 and Figure 5.6.
- SVT, in general, performs well on random examples. However, the choice of step-size and regularization parameter τ affect the convergence speed of the algorithm (Ma et al., 2011; Mazumder et al., 2010).
- FPCA has a superior numerical complexity per iteration owing to an approximate singular value decomposition (Ma et al., 2011). But the performance suffers as the regularization parameter λ is increased as shown in Figure 5.5.
- APG has a better iteration complexity than the others and is well-suited when a moderate accuracy is required (Figure 5.5 and Figure 5.6). As the ranks of the intermediate iterates are not necessarily low, scalability to large dimensions is an issue. Its accelerated version APGL does not suffer from this problem and performs very well for large dimensions.

- In all our simulation studies on random examples, Descent-restart+TR has shown a favorable performance on different benchmarks. In particular our framework is well suited when the optimal solution is low-rank and when one needs to compute the entire regularization path. The Riemannian geometry of the set of fixed-rank matrices allows us to make a local prediction of the regularization path, thereby employing an efficient predictor-corrector strategy.

5.6.3 Multivariate linear regression

Given matrices $\mathbf{Y} \in \mathbb{R}^{n \times k}$ (response space) and $\mathbf{X} \in \mathbb{R}^{n \times q}$ (input data space), we seek to find a weight/-coefficient matrix $\mathbf{W} \in \mathbb{R}^{q \times k}$ that minimizes the *loss* between $\mathbf{Y}$ and $\mathbf{XW}$ (Yuan et al., 2007). Here n is the number of observations, q is the number of predictors and k is the number of responses. One popular approach to multivariate linear regression problem is by minimizing a *quadratic loss* function. It should be noted that in various applications, *responses* are related and may therefore, be represented with much fewer coefficients. Consequently, this corresponds to finding a low-rank coefficient matrix $\mathbf{W}$ that best fits the data. The papers by Amit et al. (2007); Yuan et al. (2007) motivate the use of the trace norm regularization in the optimization problem formulation

$$\min_{\mathbf{W} \in \mathbb{R}^{q \times k}} \quad \|\mathbf{Y} - \mathbf{XW}\|_F^2 + \lambda \|\mathbf{W}\|_*,$$

where $\lambda > 0$ is the regularization parameter and the optimization variable is $\mathbf{W} \in \mathbb{R}^{q \times k}$. Although the focus here is on the quadratic loss function, our proposed optimization scheme, that alternates between fixed-rank optimization and rank-one updates, can be directly applied to other smooth loss functions.

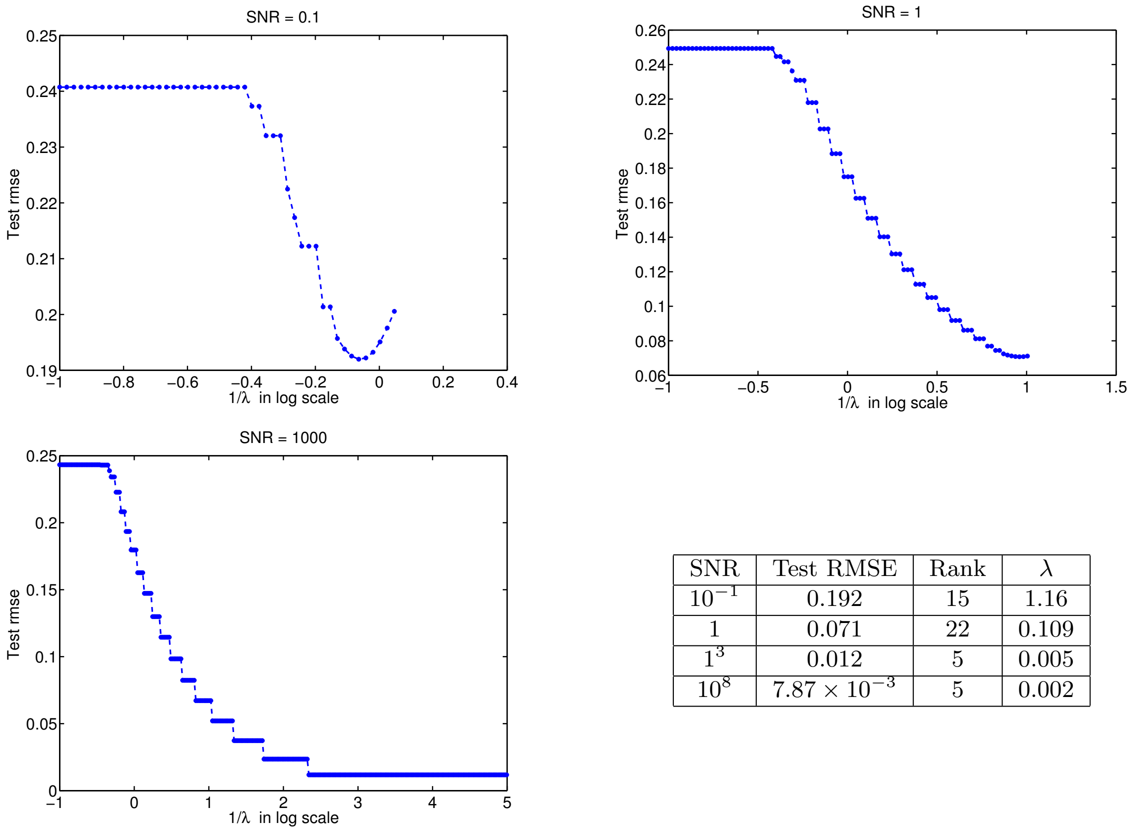
Customizing the terminology for the present problem, the convex function $F : \mathbb{R}^{q \times k} \rightarrow \mathbb{R} : \mathbf{W} \mapsto F(\mathbf{W})$ is $F(\mathbf{W}) = \|\mathbf{Y} - \mathbf{XW}\|_F^2$. Using the factorization $\mathbf{W} = \mathbf{UBV}^T$ of (5.2), the rank-p cost function $f : \mathcal{M}_p \rightarrow \mathbb{R} : (\mathbf{U}, \mathbf{B}, \mathbf{V}) \mapsto f(\mathbf{U}, \mathbf{B}, \mathbf{V})$ is $f(\mathbf{U}, \mathbf{B}, \mathbf{V}) = \|\mathbf{Y} - \mathbf{XUBV}^T\|_F^2 + \lambda \text{Trace}(\mathbf{B})$, where $(\mathbf{U}, \mathbf{B}, \mathbf{V}) \in \mathcal{M}_p$. Other than the difference in the dual variable $\mathbf{S}$, computation of the Riemannian gradient and the Riemannian Hessian for the fixed-rank problem follows directly from the developments in Table 5.1 and in Section 5.6.2. The matrix representations of the dual variable $\mathbf{S}$

$$\mathbf{S} = 2(\mathbf{X}^T\mathbf{XW} - \mathbf{X}^T\mathbf{Y}),$$

where the rank of $\mathbf{W}$ is p and it admits the matrix factorization $\mathbf{W} = \mathbf{UBV}^T$ (5.2). Building upon this, computation of the Riemannian Hessian applied along a search direction is straightforward to derive. A careful study of numerical cost of matrix operations shows that computations cost $O(q^2p + qkp)$ assuming dense matrix operations. However, additional structures of $\mathbf{X}$ and $\mathbf{Y}$ can be exploited to decrease the cost. It should be noted that the numerical complexity per iteration is *linear* in n.

5.6.3.1 Fenchel dual and duality gap computation

As an extension for some functions F of type $F(\mathbf{W}) = \psi(\mathcal{A}(\mathbf{W}))$ where $\mathcal{A}$ is a linear operator (appropriate domains of functions are assumed), computing the Fenchel conjugate of the function ψ may be easier than that of F. When $\|\mathcal{A}^*(\mathbf{M})\|_{op} \leq \lambda$ the duality gap, using similar calculations as in Proposition 5.3,



SNR	Test RMSE	Rank	λ
10^{-1}	0.192	15	1.16
1	0.071	22	0.109
1^3	0.012	5	0.005
10^8	7.87×10^{-3}	5	0.002

Figure 5.10: Regularization path for multivariate linear regression with various SNR values. Results are averaged over 5 random 70/30 splits.

is $F(\mathbf{W}) + \lambda\|\mathbf{W}\|_* + \psi^*(\mathbf{M})$, where $\mathcal{A}^*$ is the adjoint operator of $\mathcal{A}$ and ψ^* is the Fenchel conjugate of the transformed function ψ. A good choice of $\mathbf{M}$ is again $\min\{1, \frac{\lambda}{\sigma_\psi}\}\mathrm{Grad}_{\mathbf{XS}}\psi$ where σ_ψ is the dominant singular value of $\mathcal{A}^*(\mathrm{Grad}_{\mathbf{XW}}\psi)$ (Bach et al., 2011).

For the multivariate linear regression problem we have $\mathcal{A}(\mathbf{W}) = \mathbf{XW}$ which suggests the choice $F(\mathbf{W}) = \psi(\mathbf{XW})$. Note that the domains of F and ψ are different. Finally, the duality gap is $F(\mathbf{W}) + \lambda\|\mathbf{W}\|_* + \psi^*(\mathbf{M})$, where the dual candidate $\mathbf{M} = 2\min(1, \frac{\lambda}{\sigma_\psi})(\mathbf{XW} - \mathbf{Y})$ and σ_ψ is the dominant singular value of $\mathcal{A}^*(\mathrm{Grad}_{\mathbf{XW}}\psi) = \mathbf{X}^T\mathrm{Grad}_{\mathbf{XW}}\psi = 2\mathbf{X}^T(\mathbf{XW} - \mathbf{Y})$. The Fenchel dual function $\psi^* : \mathbb{R}^{n\times k} \rightarrow \mathbb{R}$ admits the expression $\psi^*(\mathbf{M}) = \mathrm{Trace}(\mathbf{M}^T\mathbf{M})/4 + \mathrm{Trace}(\mathbf{M}^T\mathbf{Y})$. Exploiting the fixed-rank factorization of $\mathbf{W}$, i.e., $\mathbf{W} = \mathbf{UBV}^T$ the numerical complexity of finding the duality gap is dominated by the numerical cost of computing $\psi^*(\mathbf{M})$ which is also of the order of the cost of computing $f(\mathbf{U}, \mathbf{B}, \mathbf{V})$. Numerical complexity of computing $\mathbf{M}$ is $O(nqp + nkp + kp^2)$ and of $\psi^*(\mathbf{M})$ is $O(nk)$.

5.6.3.2 Regularization path for multivariate linear regression

An input data matrix $\mathbf{X}$ of size 5000×120 is randomly generated according to a Gaussian distribution with zero mean and unit standard deviation. The response matrix $\mathbf{Y}$ is computed as $\mathbf{XW}_*$ where $\mathbf{W}_*$ is a randomly generated coefficient matrix of rank 5 matrix and size 120×100. We randomly split the observations as well as responses into *training* and *testing* datasets in the ratio 70/30 resulting in

$\mathbf{Y}_{\text{train}}/\mathbf{Y}_{\text{test}}$ and $\mathbf{X}_{\text{train}}/\mathbf{X}_{\text{test}}$. A Gaussian *white noise* of zero mean and variance σ^2_{noise} is added to the training response matrix $\mathbf{Y}_{\text{train}}$ resulting in $\mathbf{Y}_{\text{noise}}$. We seek to find the coefficient matrix $\mathbf{W}$ by minimizing the cost function

$$\min_{\mathbf{W}\in\mathbb{R}^{q\times k}} \quad \frac{1}{nk}\|\mathbf{Y}_{\text{noise}} - \mathbf{X}_{\text{train}}\mathbf{W}\|_F^2 + \lambda\|\mathbf{W}\|_*,$$

where λ is the trace norm regularization parameter. We validate the learning by computing the root mean square error (RMSE) defined as

$$\text{Test RMSE} = \sqrt{\frac{1}{n_{\text{test}}k}\|\mathbf{Y}_{\text{test}} - \mathbf{X}_{\text{test}}\mathbf{W}\|_F^2},$$

where n_{test} is the number of test observations. Similarly, the signal to noise ratio (SNR) is defined as $\sqrt{\frac{\|\mathbf{Y}_{\text{train}}\|_F^2}{\sigma^2_{\text{noise}}}}$.

We compute the entire regularization path for four different SNR values. The maximum value of λ is fixed at 10 and the minimum value is set at 10^{-5} with the reduction factor $\gamma = 0.95$ (270 values of λ in total). Apart from this we also put the restriction that we only fit ranks less than 30. The solution to an optimization problem for a value of λ is claimed to have been obtained when either the duality gap is less than 10^{-2} or the relative duality gap is below 10^{-2} or $\sigma_1 - \lambda$ is less than 10^{-2}. Similarly, the trust-region algorithm stops when relative or absolute variation of the cost function is less than 10^{-10}. The results are summarized in Figure 5.10.

5.7 Chapter summary

Three main ideas have been presented in this chapter. First, we have given a framework to solve the general trace norm minimization problem (5.1) with a sequence of increasing but fixed-rank problems (5.3). We have analyzed the convergence criterion and the duality gap expression which are used to monitor convergence to the solution of the original problem. The duality gap expression was shown numerically tractable even for large problems thanks to the specific choice of the low-rank parameterization. We have also presented a way of updating the rank while simultaneously ensuring a decrease of the cost function. This may be termed as a *descent-restart* approach. The second contribution of the chapter is to present a second-order trust-region algorithm for a general rank-p optimization problem on the search space $\mathrm{St}(p,n) \times \mathrm{S}_{++}(p) \times \mathrm{St}(p,m)/\mathcal{O}(p)$ that is equipped with the *natural* metric g (5.12). The search space with the proposed metric has the structure of a Riemannian submersion. We have used manifold-optimization techniques, as advocated by Absil et al. (2008), to derive the matrix expressions for proposing a trust-region algorithm. The third contribution of the chapter is to develop a *predictor-corrector* algorithm on the fixed-rank manifold to compute a grid of solutions, called a *regularization path*, corresponding to different values of the parameter λ. The resulting performance is superior to the conventional warm-restart approach. These ideas have been applied to the problems of low-rank matrix completion and multivariate linear regression leading to encouraging numerical results.

The results of the chapter have been published in the *SIAM Journal on Optimization, 2013* (Mishra et al., 2013).

Chapter 6

Conclusion and research perspectives

In this thesis, we have proposed novel algorithms that exploit the particular geometry of optimization problems that combine least-squares cost and constraints with symmetries. Our main emphasis has been on developing computationally efficient algorithms that are scalable to large-scale problems. Many current applications call for such developments, among which the low-rank matrix completion problem was a primary benchmark throughout the thesis. The main challenge is to exploit the geometry of constraint while extending the computational setup to large-scale data, which often comes with ill-conditioning issues. The thesis addresses these challenges and the novel contributions are summarized below.

Two optimization problems that combine least-squares and constraints with symmetries, rank and orthogonality constraints, were presented in the first part of Chapter 2. The second part of Chapter 2 dealt with the geometry of constraints with symmetries. In particular, the relevant constraints were identified as *quotient manifolds* resulting from structured group actions on matrix manifolds. Furthermore, the symmetries result from the interplay of few well-studied manifolds shown in Figure 2.2. We further motivate the *Riemannian optimization* framework that deals with symmetries in search space effectively. The Riemannian optimization framework on quotient manifolds forms the core foundation on which this thesis rests. Our original contributions are listed in Chapters 3, 4, and 5, where we propose efficient Riemannian algorithms for least-squares problems with rank and orthogonality constraints.

The connection between the Riemannian optimization framework and sequential quadratic programming (SQP) in *manifold-constrained* optimization has been explored in Chapter 3. Building upon their equivalence for *submanifolds* (manifolds embedded in a vector space), we have explored the relationship for quotient manifolds. We have shown that this connection allows us to construct Riemannian metrics (smooth inner products) on manifolds that are tailored to least-squares costs. Such metrics can be thought of as an effective way to perform *preconditioning* on Riemannian manifolds, particularly crucial when dealing with ill-conditioned data. The notion of metric tuning was linked to the ability of capturing relevant second-order information of the problem at hand. Two particular case studies were considered: the *generalized eigenvalue problem*, a least-squares problem with orthogonality constraints, and the *matrix Lyapunov equations*, a least-squares problem with rank constraints. We have established novel connections with *power*, *inverse*, and *Rayleigh quotient* iterations for the generalized eigenvalue problem, all described as steepest-descent algorithms with specific choices of metrics. For the matrix Lyapunov equations, we have proposed novel metrics that show good performance as compared to more conventional metric choices.

Chapter 4 specifically focused on the low-rank matrix completion problem by fixing the rank a priori. The concept of metric tuning of Chapter 3 provided a basis to construct computationally *efficient metrics* on the Riemannian manifold of fixed-rank matrices for the low-rank matrix completion problem. Two novel conjugate-gradient algorithms were proposed and tested in significant problems against state-of-the-art algorithms. All the resulting matrix formulas have been summarized in tables. The novel algorithms suggest superior performance on various benchmarks.

Finally in Chapter 5, we have dealt with large-scale convex programs where the expected solution is low-rank. Here the low-rank constraint is enforced "softly" via a *trace norm regularization* term in the cost function. The trace norm regularization term is convex but non-smooth. Conventional convex optimization algorithms tackle this non-smoothness by a *soft-thresholding* operation on singular values of iterates. The basic approach taken in this thesis was to use a particular fixed-rank factorization that made the trace norm *differentiable* on the fixed-rank manifold. This allowed us to exploit the Riemannian geometry of fixed-rank matrices. Our approach alternates between fixed-rank optimization and rank-one updates and ensures a monotonic decrease of the cost function. Such an approach provides a tighter control over the rank of iterates and provides a trade-off between computational efficiency and minimizing data-fitting error. This scheme was used to propose an efficient *predictor-corrector* scheme to compute *regularization path* of solutions by tuning the regularization parameter. Two examples of low-rank matrix completion and multivariate regression were solved with the proposed setup.

Research perspectives

Many rank constrained optimization problems in engineering have additional *structure* beyond what was considered in the present thesis. In particular, coupling low-rank constraint to *affine* constraints seems particularly challenging. The survey paper by Markovsky (2008) provides a rich insight into such problems and their numerous applications. Even identifying a *feasible critical point* of the optimization problem might be difficult in such applications. However, the recent progress, notably by Ishteva et al. (2013); Markovsky (2014); Markovsky and Usevich (2013); Schost and Spaenlehauer (2013), has shown promising directions. As a future research direction, it would be interesting to explore the role of Riemannian optimization in these problems. More generally, we would like to explore optimization problems where the constraints have an *affine + manifold* structure.

The large-scale problems considered in this thesis have been tackled by exploiting the low-rank and/or sparse structures. This is, for example, the case in the matrix completion problem. However, such structures are of little help when considering the class of even bigger dimensional problems that are motivated by the ever increasing size of internet based applications. In order to circumvent the scaling issue, the *coordinate-descent* approach has become a popular choice in many *unconstrained* and *separable-constrained* (constraints which can be decoupled) optimization problems. The coordinate-descent method in its basic form partitions the variables into a large number of blocks of smaller number of variables

(Nocedal and Wright, 2006, Chapter 3). The optimization method proceeds by updating these blocks of variables alternatively, e.g., either *randomly* or *sequentially*. The recent article by Nesterov (2012) presents a concrete overview of the *random* coordinate-descent method (where partitions of the variable are updated randomly) with a detailed analysis. This motivates a notion of a coordinate-descent-type approach on manifolds for dealing with huge-scale problems. The recent paper by Shalit and Chechik (2014) proposes such an approach on the set of orthogonal matrices with state-of-the-art performance in many applications. It would be interesting to pursue this research direction and look at possible generalizations and applications.

Finally, many cost functions are a summation of smooth functions which are revealed at different time instances. Conventional optimization approaches are of little use since the full cost function is not known at any time instance. In such problems (that most notably arise in machine learning applications), the *stochastic gradient descent* method exploits the summation structure to converge to a minimum of the *expectation of the cost* (Bottou, 1998). At each step, that is at each time instance, the variables are updated along the gradient descent direction computed with respect to the currently revealed smooth function. Recently, the stochastic gradient descent method has been extended to Riemannian manifolds with a convergence analysis by Bonnabel (2013). As a third research perspective, we would like to explore the research direction further.

Appendix A

Solution to smaller dimensional Lyapunov equations

We present the numerical approach that we adopt in this thesis to solve the two specific forms of the Lyapunov equations of *smaller dimension*. These appear in Table 4.3 (Chapter 4) and Table 5.1 (Chapter 5).

Standard form

We are interested in computing the solution $\mathbf{\Omega} \in \mathbb{R}^{r\times r}$ to the Lyapunov equation of the form

$$\mathbf{D}\mathbf{\Omega} + \mathbf{\Omega}\mathbf{D} = \mathbf{C}, \tag{A.1}$$

where $\mathbf{D}$ is a symmetric *positive definite* matrix of size $r \times r$ and $\mathbf{C}$ is a square matrix of size $r \times r$. For the case in Table 4.3, $\mathbf{D} = \mathbf{R}^T\mathbf{R}$, where $\mathbf{R} \in \mathrm{GL}(r)$ is a non-singular (non-zero determinant) matrix of size $r \times r$. For the case Table 5.1, $\mathbf{D} = \mathbf{B}$, where $\mathbf{B}$ is a symmetric positive definite matrix of size $r \times r$.

We compute the *eigenvalue decomposition* $\mathbf{D} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with positive entries and $\mathbf{Q} \in \mathcal{O}(r)$ (the set of orthogonal matrices) is the orthogonal matrix such that $\mathbf{Q}^T\mathbf{Q} = \mathbf{Q}\mathbf{Q}^T = \mathbf{I}$. We follow the approach shown below to solve (A.1), where we replace $\mathbf{D}$ by $\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$.

$$\begin{array}{rlcl}
 & \mathbf{Q}\mathbf{\Lambda}^2\mathbf{Q}^T\mathbf{\Omega} + \mathbf{\Omega}\mathbf{Q}\mathbf{\Lambda}^2\mathbf{Q}^T & = & \mathbf{C} \\
\Rightarrow & \underbrace{\mathbf{\Lambda}^2}_{\widetilde{\mathbf{\Lambda}}}\mathbf{Q}^T\mathbf{\Omega}\mathbf{Q} + \underbrace{\mathbf{Q}^T\mathbf{\Omega}\mathbf{Q}}_{\widetilde{\mathbf{\Omega}}}\mathbf{\Lambda}^2 & = & \underbrace{\mathbf{Q}^T\mathbf{C}\mathbf{Q}}_{\widetilde{\mathbf{C}}} \\
\Rightarrow & \widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{\Omega}} + \widetilde{\mathbf{\Omega}}\widetilde{\mathbf{\Lambda}} & = & \widetilde{\mathbf{C}} \\
\Rightarrow & (\widetilde{\sigma}\mathrm{e}^T) \odot \widetilde{\mathbf{\Omega}} + \widetilde{\mathbf{\Omega}} \odot (\mathrm{e}\widetilde{\sigma}^T) & = & \widetilde{\mathbf{C}} \\
\Rightarrow & (\widetilde{\sigma}\mathrm{e}^T + \mathrm{e}\widetilde{\sigma}^T) \odot \widetilde{\mathbf{\Omega}} & = & \widetilde{\mathbf{C}} \\
\Rightarrow & \widetilde{\mathbf{\Omega}} & = & \widetilde{\mathbf{C}} \oslash (\widetilde{\sigma}\mathrm{e}^T + \mathrm{e}\widetilde{\sigma}^T),
\end{array} \tag{A.2}$$

where $\widetilde{\sigma}$ is a column vector of length r that contains all the diagonal entries of $\widetilde{\mathbf{\Lambda}}$ and e is the column vector of all ones, of length r. Here $\odot$ and $\oslash$ denote element-wise multiplication and division of matrices, respectively. Finally, we compute the sought solution $\mathbf{\Omega}$ by using the equality $\mathbf{Q}^T\mathbf{\Omega}\mathbf{Q} = \widetilde{\mathbf{\Omega}}$, i.e.,

$$\mathbf{\Omega} = \mathbf{Q}(\widetilde{\mathbf{C}} \oslash (\widetilde{\sigma}\mathrm{e}^T + \mathrm{e}\widetilde{\sigma}^T))\mathbf{Q}^T, \tag{A.3}$$

where $\widetilde{\mathbf{C}} = \mathbf{Q}^T\mathbf{C}\mathbf{Q}$. The operations in (A.2) and (A.3) cost $O(r^3)$.

Coupled form

We are interested in computing the solutions $\mathbf{\Omega}_1, \mathbf{\Omega}_2 \in \mathbb{R}^{r\times r}$ that satisfy the *coupled Lyapunov equations* of the form

$$\begin{aligned} \mathbf{R}\mathbf{\Omega}_2\mathbf{R}^T - \mathbf{R}\mathbf{R}^T\mathbf{\Omega}_1 - \mathbf{\Omega}_1\mathbf{R}\mathbf{R}^T &= \mathbf{C}_1 \\ \mathbf{R}^T\mathbf{\Omega}_1\mathbf{R} - \mathbf{R}^T\mathbf{R}\mathbf{\Omega}_2 - \mathbf{\Omega}_2\mathbf{R}^T\mathbf{R} &= \mathbf{C}_2, \end{aligned} \tag{A.4}$$

where $\mathbf{C}_1$ and $\mathbf{C}_2$ be are square matrices of size $r \times r$ and $\mathbf{R} \in \mathrm{GL}(r)$ is a non-singular matrix of size $r \times r$. This equation shows up in Table 4.3.

We first compute the singular value decomposition $\mathbf{R} = \mathbf{P}\mathbf{\Lambda}\mathbf{Q}^T$, where $\mathbf{\Lambda}$ is a diagonal matrix with positive entries and $\mathbf{P}, \mathbf{Q} \in \mathcal{O}(r)$ (the set of orthogonal matrices of size $r \times r$). We invoke similarity transformations on the variables $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ to define new variables $\widetilde{\mathbf{\Omega}}_1 = \mathbf{P}^T\mathbf{\Omega}_1\mathbf{P}$ and $\widetilde{\mathbf{\Omega}}_2 = \mathbf{Q}^T\mathbf{\Omega}_2\mathbf{Q}$. Similarly, define $\widetilde{\mathbf{C}}_1 = \mathbf{P}^T\mathbf{C}_1\mathbf{P}$, $\widetilde{\mathbf{C}}_2 = \mathbf{Q}^T\mathbf{C}_2\mathbf{Q}$, and $\widetilde{\mathbf{\Lambda}} = \mathbf{\Lambda}^2$. The equations in (A.4) are equivalently written in the new defined variables as

$$\begin{aligned} \Rightarrow \quad & \begin{array}{rcl} \mathbf{\Lambda}\widetilde{\mathbf{\Omega}}_2\mathbf{\Lambda} - \widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{\Omega}}_1 - \widetilde{\mathbf{\Omega}}_1\widetilde{\mathbf{\Lambda}} & = & \widetilde{\mathbf{C}}_1 \\ \mathbf{\Lambda}\widetilde{\mathbf{\Omega}}_1\mathbf{\Lambda} - \widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{\Omega}}_2 - \widetilde{\mathbf{\Omega}}_2\widetilde{\mathbf{\Lambda}} & = & \widetilde{\mathbf{C}}_2, \end{array} \\ \Rightarrow \quad & \begin{array}{rcl} (\sigma\mathrm{e}^T) \odot (\mathrm{e}\sigma^T) \odot \widetilde{\mathbf{\Omega}}_2 - (\widetilde{\sigma}\mathrm{e}^T + \mathrm{e}\widetilde{\sigma}^T) \odot \widetilde{\mathbf{\Omega}}_1 & = & \widetilde{\mathbf{C}}_1 \\ (\sigma\mathrm{e}^T) \odot (\mathrm{e}\sigma^T) \odot \widetilde{\mathbf{\Omega}}_1 - (\widetilde{\sigma}\mathrm{e}^T + \mathrm{e}\widetilde{\sigma}^T) \odot \widetilde{\mathbf{\Omega}}_2 & = & \widetilde{\mathbf{C}}_2, \end{array} \end{aligned} \tag{A.5}$$

where $\widetilde{\sigma}$ is a column vector of length r that contains all the diagonal entries of $\widetilde{\mathbf{\Lambda}}$, σ is a column vector containing all the diagonal entries of $\mathbf{\Lambda}$, and e is the column vector of length r containing ones. (A.5) can now be solved *efficiently* for $\widetilde{\mathbf{\Omega}}_1$ and $\widetilde{\mathbf{\Omega}}_2$. Subsequently, $\mathbf{\Omega}_1$ and $\mathbf{\Omega}_2$ are obtained by the *inverse* similarity transforms $\mathbf{P}\widetilde{\mathbf{\Omega}}_1\mathbf{P}^T$ and $\mathbf{Q}\widetilde{\mathbf{\Omega}}_2\mathbf{Q}^T$. Here $\odot$ denotes element-wise multiplication between matrices of same size. The operations in (A.5) cost $O(r^3)$.

Bibliography

Abernethy J, Bach F, Evgeniou T, Vert JP (2009) A new approach to collaborative filtering: Operator estimation with spectral regularization. Journal of Machine Learning Research 10(Mar):803–826

Absil PA (2003) Invariant subspace computation: A geometric approach. PhD thesis, Faculty of Applied Science, Univesity of Liège

Absil PA, Van Dooren P (2010) Two-sided Grassmann-Rayleigh quotient iteration. Numerische Mathematik 114(4):549–571

Absil PA, Mahony R, Sepulchre R, Van Dooren P (2002) A Grassmann-Rayleigh quotient iteration for computing invariant subspaces. SIAM Review 44(1):57–73

Absil PA, Mahony R, Sepulchre R (2004a) Riemannian geometry of Grassmann manifolds with a view on algorithmic computation. Acta Applicandae Mathematicae 80(2):199–220

Absil PA, Sepulchre R, Van Dooren P, Mahony R (2004b) Cubically convergent iterations for invariant subspace computation. SIAM Journal on Matrix Analysis and Applications 26(1):70–96

Absil PA, Mahony R, Sepulchre R (2008) Optimization Algorithms on Matrix Manifolds. Princeton University Press, Princeton, NJ

Absil PA, Trumpf J, Mahony R, Andrews B (2009) All roads lead to Newton: Feasible second-order methods for equality-constrained optimization. Tech. rep., UCL-INMA-2009.024

Absil PA, Amodei L, Meyer G (2014) Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. Computational Statistics 29(3–4):569–590

Amatriain X, Basilico J (2012) Netflix recommendations: Beyond the 5 stars (part 1). http://techblog.netflix.com/2012/04/netflix-recommendations-beyond-5-stars.html

Amit Y, Fink M, Srebro N, Ullman S (2007) Uncovering shared structures in multiclass classification. In: Proceedings of the 24th International Conference on Machine Learning, pp 17–24

Bach F (2008) Consistency of trace norm minimization. Journal of Machine Learning Research 9(Jun):1019 – 1048

Bach F, Jenatton R, Mairal J, Obozinsky G (2011) Convex optimization with sparsity-inducing norms. In: S Sra SN, Wright SJ (eds) Optimization for Machine Learning, MIT Press

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

Balzano L, Nowak R, Recht B (2010) Online identification and tracking of subspaces from highly incomplete information. In: The 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp 704–711

Bartels RH, Stewart GW (1972) Solution of the matrix equation ax+xb=c [f4] (algorithm 432). Commun ACM 15(9):820–826

Benner P, Saak J (2013) Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: A state of the art survey. Tech. rep., MPI Magdeburg

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

Bonnabel S (2013) Stochastic gradient descent on Riemannian manifolds. IEEE Transactions on Automatic Control 58(9):2217–2229

Bonnabel S, Sepulchre R (2009) Riemannian metric and geometric mean for positive semidefinite matrices of fixed rank. SIAM Journal on Matrix Analysis and Applications 31(3):1055–1070

Bottou L (1998) Online algorithms and stochastic approximations. In: Saad D (ed) Online Learning in Neural Networks, Cambridge University Press, Cambridge, UK, pp 9–42

Boumal N (2014) Optimization and estimation on manifolds. PhD thesis, Université catholique de Louvain

Boumal N, Absil PA (2011) RTRMC: A Riemannian trust-region method for low-rank matrix completion. In: Advances in Neural Information Processing Systems 24 (NIPS), pp 406–414

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

Boumal N, Singer A, Absil PA, Blondel V (2013) Cramér-Rao bounds for synchronization of rotations. Information and Inference URL Doi:10.1093/imaiai/iat006

Boumal N, Mishra B, Absil PA, Sepulchre R (2014) Manopt: a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research 15(Apr):1455–1459

Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press

Brand M (2006) Fast low-rank modifications of the thin singular value decomposition. Linear Algebra and its Applications 415(1):20–30

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

Burer S, Monteiro R (2003) A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. Mathematical Programming 95(2):329–357

Cai D, He X, Han J (2007) Efficient kernel discriminant analysis via spectral regression. In: IEEE International Conference on Data Mining (ICDM), pp 427–432

Cai JF, Candès EJ, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization 20(4):1956–1982

Candès EJ, Plan Y (2009) Matrix completion with noise. Proceedings of the IEEE 98(6):925–936

Candès EJ, Recht B (2009) Exact matrix completion via convex optimization. Foundations of Computational Mathematics 9(6):717–772

Carron I (2014) The advanced matrix factorization jungle. URL https://sites.google.com/site/igorcarron2/matrixfactorizations

Da Silva C, Herrmann FJ (2014) Optimization on the Hierarchical Tucker manifold - applications to tensor completion. Tech. rep., arXiv:1405.2096

Dai W, Milenkovic O, Kerman E (2011) Subspace evolution and transfer (SET) for low-rank matrix completion. IEEE Transactions on Signal Processing 59(7):3120–3132

Dai W, Kerman E, Milenkovic O (2012) A geometric approach to low-rank matrix completion. IEEE Transactions on Information Theory 58(1):237–247

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

Eldén L, Park H (1999) A Procrustes problem on the Stiefel manifold. Numerische Mathematik 82(4):599–619

Fazel M (2002) Matrix rank minimization with applications. PhD thesis, Department of Electrical Engineering, Stanford University

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

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

Hiriart-Urruty JB, Lemaréchal C (1993) Convex analysis and minimization algorithms. Springer-Verlag, Berlin, Germany

Ishteva M, Usevich K, Markovsky I (2013) Factorization approach to structured low-rank approximation with applications. Tech. rep., arXiv:1308.1827

Jain P, Meka R, Dhillon I (2010) Guaranteed rank minimization via singular value projection. In: Advances in Neural Information Processing Systems 23 (NIPS), pp 937–945

Jeffrey DJ (2010) LU factoring of non-invertible matrices. ACM Communications in Computer Algebra 44(171):1–8

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

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

Journée M, Bach F, Absil PA, Sepulchre R (2010) Low-rank optimization on the cone of positive semidefinite matrices. SIAM Journal on Optimization 20(5):2327–2351

Journée M, Nesterov Y, Richtárik P, Sepulchre R (2010) Generalized Power method for sparse principal component analysis. Journal of Machine Learning Research 11(Feb):517–553

Keshavan RH, Montanari A, Oh S (2010) Matrix completion from a few entries. IEEE Transactions on Information Theory 56(6):2980–2998

Krantz SG, Parks HR (2002) The implicit function theorem: history, theory, and applications. Birkhäuser, Boston, MA

Kressner D, Steinlechner M, Vandereycken B (2013) Low-rank tensor completion by Riemannian optimization. BIT Numerical Mathematics Doi: 10.1007/s10543-013-0455-z

Kulis B, Sustik M, Dhillon IS (2009) Low-rank kernel learning with Bregman matrix divergences. Journal of Machine Learning Research 10(Feb):341–376

Larsen RM (2004) PROPACK - software for large and sparse SVD calculations, version 1.1. URL http://soi.stanford.edu/~rmunk/PROPACK/

Lee JM (2003) Introduction to smooth manifolds, Graduate Texts in Mathematics, vol 218, 2nd edn. Springer-Verlag, New York

Lee K, Bresler Y (2010) Admira: Atomic decomposition for minimum rank approximation. IEEE Transactions on Information Theory 56(9):4402–4416

Li JR, White J (2004) Low-rank solution of Lyapunov equations. SIAM Review 46(4):693–713

Ma S, Goldfarb D, Chen L (2011) Fixed point and bregman iterative methods for matrix rank minimization. Mathematical Programming 128(1–2):321–353

Manton J (2002) Optimization algorithms exploiting unitary constraints. IEEE Transactions on Signal Processing 50(3):635–650

Markovsky I (2008) Structured low-rank approximation and its applications. Automatica 44(4):891–909

Markovsky I (2014) Recent progress on variable projection methods for structured low-rank approximation. Signal Processing 96(Part B):406–419

Markovsky I, Usevich K (2013) Structured low-rank approximation with missing data. SIAM Journal on Matrix Analysis and Applications 34(2):814–830

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

Meka R, Jain P, Dhillon IS (2009) Matrix completion from power-law distributed samples. In: Advances in Neural Information Processing Systems 22 (NIPS), pp 1258–1266

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

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

Meyer G, Bonnabel S, Sepulchre R (2011b) Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. Journal of Machine Learning Research 11(Feb):593–625

Mishra B, Sepulchre R (2014a) R3MC: A Riemannian three-factor algorithm for low-rank matrix completion. In: Accepted for publication in the proceedings of the 53rd IEEE Conference on Decision and Control (CDC)

Mishra B, Sepulchre R (2014b) Riemannian preconditioning. Tech. rep., arXiv:1405.6055

Mishra B, Vandereycken B (2014) A Riemannian approach to low-rank algebraic Riccati equations. In: Proceedings of the 21st International Symposium on Mathematical Theory of Networks and Systems (MTNS), pp 965–968

Mishra B, Meyer G, Sepulchre R (2011) Low-rank optimization for distance matrix completion. In: Proceedings of the 50th IEEE Conference on Decision and Control (CDC-ECC), pp 4455–4460

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

Mishra B, Meyer G, Bach F, Sepulchre R (2013) Low-rank optimization with trace norm penalty. SIAM Journal on Optimization 23(4):2124–2149

Mishra B, Meyer G, Bonnabel S, Sepulchre R (2014) Fixed-rank matrix factorizations and Riemannian low-rank optimization. Computational Statistics 29(3–4):591–621

MovieLens (1997) MovieLens. URL http://grouplens.org/datasets/movielens/

Nesterov Y (2003) Introductory lectures on convex optimization: a basic course. No. 87 in Applied Optimization, Kluwer Academic Publishers

Nesterov Y (2012) Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM Journal on Optimization 22(2):341–362

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

Ngo TT, Saad Y (2012) Scaled gradients on Grassmann manifolds for matrix completion. In: Advances in Neural Information Processing Systems 25 (NIPS), pp 1421–1429

Nocedal J, Wright SJ (2006) Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering, Springer, New York, USA

Park MY, Hastie T (2006) Regularization path algorithms for detecting gene interactions. Tech. rep., Department of Statistics, Stanford University, URL http://www.stanford.edu/~hastie/Papers/glasso.pdf

Penzl T (1999) A cyclic low-rank Smith method for large sparse Lyapunov equations. SIAM Journal on Scientific Computing 21(4):1401–1418

Piziak R, Odell PL (1999) Full rank factorization of matrices. Mathematics Magazine 72(3):193–201

Recht B, Ré C (2013) Parallel stochastic gradient algorithms for large-scale matrix completion. Mathematical Programming Computation 5(2):201–226

Recht B, Fazel M, Parrilo PA (2010) Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Review 53(3):471–501

Rennie J, Srebro N (2005) Fast maximum margin matrix factorization for collaborative prediction. In: International Conference on Machine learning (ICML), pp 713–719

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

Sato H, Iwai T (2013) A new, globally convergent Riemannian conjugate gradient method. Optimization: A Journal of Mathematical Programming and Operations Research Doi: 10.1080/02331934.2013.836650

Schost E, Spaenlehauer PJ (2013) A quadratically convergent algorithm for structured low-rank approximation. Tech. rep., arXiv:1312.7279

Shalit U, Chechik G (2014) Coordinate-descent for learning orthogonal matrices through Givens rotations. In: Proceedings of the 31st International Conference on Machine Learning (ICML), pp 548–556

Shalit U, Weinshall D, Chechik G (2010) Online learning in the manifold of low-rank matrices. In: Neural Information Processing Systems conference (NIPS), pp 2128–2136

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

Smith S (1994) Optimization techniques on Riemannian manifold. In: Bloch A (ed) Hamiltonian and Gradient Flows, Algorithms and Control, vol 3, American Mathematical Society, Providence, RI, pp 113–136

Smith S (2005) Covariance, subspace, and intrinsic Cramér-Rao bounds. IEEE Transactions on Signal Processing 53:1610–1630

Srebro N, Jaakkola T (2003) Weighted low-rank approximations. In: Proceedings of the 20th International Conference on Machine Learning (ICML), pp 720–727

Teschendorff AE, Journée M, Absil PA, Sepulchre R, Caldas C (2007) Elucidating the altered transcriptional programs in breast cancer using independent component analysis. PLoS Computational Biology 3(8):e161

Theis FJ, Cason TP, Absil PA (2009) Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. In: Independent Component Analysis and Signal Separation, Springer Berlin Heidelberg, Berlin, Germany, pp 354–361

Toh KC, Yun S (2010) An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pacific Journal of Optimization 6(3):615–640

Usevich K, Markovsky I (2014) Optimization on a Grassmann manifold with application to system identification. Automatica 50(6):1656–1662

Vandereycken B (2010) Riemannian and multilevel optimization for rank-constrained matrix problems. PhD thesis, Department of Computer Science, KU Leuven

Vandereycken B (2013) Low-rank matrix completion by Riemannian optimization. SIAM Journal on Optimization 23(2):1214–1236

Vandereycken B, Vandewalle S (2010) A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. SIAM Journal on Matrix Analysis and Applications 31(5):2553–2579

Vandereycken B, Absil P, Vandewalle S (2013) A Riemannian geometry with complete geodesics for the set of positive semidefinite matrices of fixed rank. IMA Journal of Numerical Analysis 33(2):481–514

Viklands T (2006) Algorithms for the weighted orthogonal Procrustes problem and other least squares problems. PhD thesis, Department of Computing Science, Umeå University

Vounou M, Nichols TE, Montana G, Alzheimer's Disease Neuroimaging Initiative (2010) Discovering genetic associations with high-dimensional neuroimaging phenotypes: A sparse reduced-rank regression approach. Neuroimage 53(3):1147–1159

Wen Z, Yin W (2013) A feasible method for optimization with orthogonality constraints. Mathematical Programming 142(1–2):397–434

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

Yuan M, Ekici A, Lu Z, Monteiro R (2007) Dimension reduction and coefficient estimation in multivariate linear regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69(3):329–346