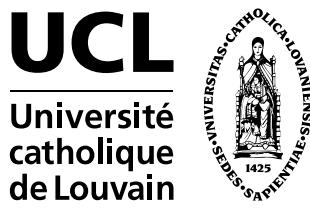


Optimization and estimation on manifolds

Nicolas Boumal



Thesis submitted in partial fulfillment of the requirements for the degree of
Docteur en sciences de l'ingénieur

Dissertation committee:

Prof. Pierre-Antoine Absil (UCL, advisor)

Prof. Vincent D. Blondel (UCL, advisor)

Prof. Alexandre d'Aspremont (École Normale Supérieure de Paris, France)

Prof. Rodolphe Sepulchre (University of Cambridge, UK)

Prof. Amit Singer (Princeton University, NJ, USA)

Prof. Paul Van Dooren (UCL)

Prof. Michel Verleysen (UCL, chair)

February 2014

Abstract

How to make the best decision? This general concern, pervasive in both research and industry, is what *optimization* is all about. Optimization is a field of applied mathematics concerned with making the best use—according to some quantitative criterion called the *cost function*—of our degrees of freedom called the *variables*, possibly under some *constraints*.

Optimization problems come in various forms. We consider continuous variables with differentiable cost functions. Furthermore, and this is central to our investigation, we assume that the variables are constrained to belong to a *Riemannian manifold*, that is, to a smooth space.

Building upon prior theory, we develop *Manopt*, a toolbox which considerably simplifies the use of Riemannian optimization. We apply this tool to two applications. First, we study *low-rank matrix completion*, which appears in recommender systems. Such systems aim at predicting which movies, books, etc. different users might appreciate, based on partial knowledge of their preferences. Second, we study *synchronization of rotations*. This is a central player in the reconstruction of 3D computer models of physical objects based on scans of their surface. In both cases, Riemannian optimization provides competitive, scalable and accurate algorithms.

Both applications constitute *estimation* problems. In estimation, one wishes to determine the value of unknown parameters based on noisy measurements. We address the following fundamental question: given a noise level on the measurements, how accurately can one hope to estimate the parameters? This prompts us to further develop Cramér-Rao bounds when the parameter space is a manifold. Applied to synchronization, these bounds bring about practical implications. First, they suggest that in many non-trivial scenarios, our estimation algorithm could be optimal. Second, they reveal the defining features that make a synchronization task more or less difficult, hinting at which measurements should be acquired.

Acknowledgments

As a master's student approaching graduation, academia ever more clearly held the promise of a rich environment offering intellectual freedom and colleagues to be proud of. And boy did it live up to those expectations. I have been surrounded by people I respect, admire and enjoy. This and their support has been a primary resource in concluding my thesis.

Pierre-Antoine, over the years you have remained available and enthusiastic, always crystal clear in addressing my questions with your unique style of hunting for answers:

One drawing to enlight it all, and with prowess, find them.

Your signature as a mentor, be it technical, in writing or in my interactions with collaborators, is noticeable all over my ways as a researcher.

Vincent, for me as for all of your students, you have strived to open international doors and you have encouraged me to interact with peers as much as enjoyable, to great effect. As a student, I was once warned that research is a lonely ride. It most certainly does not have to be. Thank you also for your precious and unique guidance in the various aspects of the academic life.

Amit, thank you for making me feel so welcome at PACM. On my different stays in your lab, I have felt like an integral part of your research group. Interactions with you and your students have a tremendous impact on my work and my perspective on research.

Alex, Michel, Paul and Rodolphe, together with Amit and my advisors, you are my dream team of a jury. I am touched by your interest in my work and personally grateful for your shared experience.

For the People of Euler—the wise ones who welcomed me in a joyful environment, the promising ones who continue to cultivate it and the everlasting ones who steer it—let there be pie. You are a formidable bunch to leave. I'll dare special thanks to my officemate Romain, but there are so many of you who deserve special thanks: how many more pages would my readers need to turn before they finally get to see a theorem statement?

Mihai, Afonso, Lanhui, Xiuyuan, Jane, Onur and Kevin, special thanks to you for welcoming me warmly on my visits to Fine Hall: you kept me flying back for more. Pierre and Bamdev, thank you for starting and pursuing the Manopt adventure with me. Bart, Gilles and Quentin, it's been great to have you in the manifold crowd to hang out with. Maxime, Antoine, your lively entrepreneurship and fresh perspectives are precious sources of motivation.

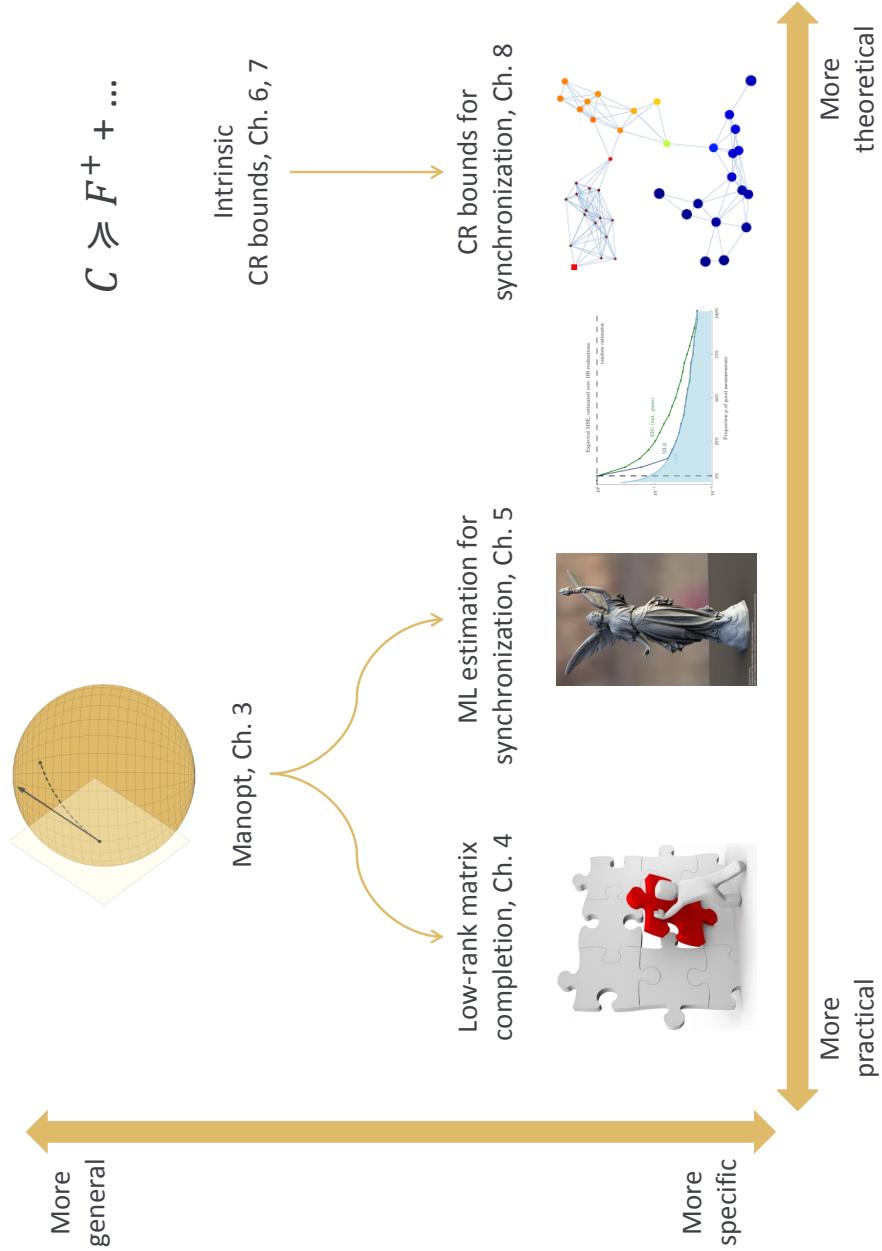
To my loved ones, friends and family—you know who you are—a very big thank you. The PhD life has been a blast, and that I owe to all of you even more than to the sheer fun it is to do math. It has also been a demanding life on more than one occasion. Perhaps some of you don't do math too often themselves, but your support proved theorems and no one can deny that.

Another resource, more down to earth but well necessary, has been the rare research comfort the professors at the INMA lab have constructed and maintained over the years. Vincent, Paul, Pierre-Antoine, Jojo, Jean-Charles, Denis, Michel, François, Julien, Raphaël, Philippe, Yurii, Vincent, in upholding the ARC and PAI programs (among others), you give your graduate students a good look at the international scene and the productively appeasing certainty they will be supported in their projects. Thank you. Likewise, I gratefully acknowledge the FNRS for funding my research and many of my US visits.

Contents

1	Introduction	15
2	Elements of Riemannian geometry	21
2.1	Charts and manifolds	22
2.2	Tangent spaces and tangent vectors	24
2.3	Riemannian structure	28
2.4	Connections and Hessians	32
2.5	Distances and geodesic curves	36
2.6	Exponential and logarithmic maps	38
2.7	Parallel translation	40
2.8	Curvature	42
I	Optimization	45
3	Optimization on manifolds	47
3.1	Riemannian conjugate gradients	48
3.2	Riemannian trust-regions	53
3.3	Manopt, a Matlab toolbox for optimization on manifolds	55
4	Low-rank matrix completion	65
4.1	Geometry of the Grassmann manifold	69
4.2	The cost function and its derivatives	73
4.3	Riemannian optimization setup	85
4.4	Numerical experiments	87
4.5	Application: the Netflix prize	97
4.6	Conclusions	100
5	Synchronization of rotations	101
5.1	Robust synchronization of rotations	103
5.2	Geometry of the parameter space, with anchors	106

5.3	The eigenvector method and its phase transition point	108
5.4	An algorithm to compute the maximum likelihood estimator	114
5.5	Numerical experiments	123
5.6	Application: 3D scan registration	138
5.7	Conclusions	142
II	Estimation bounds	145
6	Estimation on manifolds	147
6.1	Fisher information, bias and covariance	148
6.2	Intrinsic Cramér-Rao bounds	151
7	CRB's on sub- and quotient manifolds	159
7.1	Riemannian submanifolds	161
7.2	Riemannian quotient manifolds	165
7.3	Including curvature terms	170
7.4	Example	174
7.5	Conclusions	179
8	CRB's for synchronization of rotations	181
8.1	A family of noise models	185
8.2	Geometry of the parameter space, without anchors	188
8.3	Measures, integrals and distributions on $\text{SO}(n)$	191
8.4	The Fisher information matrix	193
8.5	The Cramér-Rao bounds	200
8.6	Curvature terms	206
8.7	Comments on, and consequences of the CRB	210
8.8	Conclusions	216
9	Conclusions	217
A	Integration over $\text{SO}(n)$	223
A.1	Langevin density normalization	224
A.2	Mixture of Langevin information weight	226
B	CRB's for synchronization of rotations: proof details	229
B.1	Proof of two properties of G_{ij}	229
B.2	Proof of Lemma 8.3	231



Notation

Vectors and matrices

\mathbb{R}	Set of real numbers
\mathbb{R}^+	Set of positive real numbers
\mathbb{R}^n	Set of real column vectors of size n
$\mathbb{R}^{m \times n}$	Set of real matrices with m rows and n columns
$\mathbb{R}_*^{m \times n}$	Set of full-rank $m \times n$ real matrices
I, I_n	Identity matrix of size n (or of size indicated by context)
$\mathbf{1}_n, \mathbf{1}_{m \times n}$	Column vector or matrix of all ones
e_1, \dots, e_n	Canonical basis vectors of \mathbb{R}^n : the columns of I_n
A^\top	Transpose of the matrix A
$\text{trace}(A)$	Trace of the square matrix A (sum of the diagonal entries)
$\text{diag}(A)$	Extracts the diagonal entries of A , in a column vector
$\text{col}(A)$	Subspace spanned by the columns of A
$\ A\ , \ A\ _{\text{F}}$	Frobenius norm of the matrix A , $\ A\ _{\text{F}} = \sqrt{\text{trace}(A^\top A)}$
$\lambda_{\max}(A)$	Largest eigenvalue of A , in magnitude
A^\dagger	Moore-Penrose pseudo-inverse of the matrix A
$\text{sym}(A)$	Symmetric part of the square matrix A : $(A + A^\top)/2$
$\text{skew}(A)$	Skew-symmetric part of the square matrix A : $(A - A^\top)/2$
$A \odot B$	Hadamard (entry-wise) product of matrices A and B
$A \otimes B$	Kronecker product of matrices A and B
$[X, Y]$	Lie bracket or commutator: $[X, Y] = XY - YX$
$A \succeq 0$	Positive semidefinite matrix
$\exp(A), \log(A)$	Matrix exponential and logarithm
\mathbf{X}	Tuple of matrices: $\mathbf{X} = (X_1, \dots, X_N)$
\mathbf{XY}	Product of tuples, entry-wise: $\mathbf{XY} = (X_1 Y_1, \dots, X_N Y_N)$

Sets and manifolds

$\mathcal{M}, \mathcal{N}, \mathcal{P}$	Smooth, finite-dimensional (usually Riemannian) manifolds
\mathcal{M}	A probability space (in the second part of this thesis)
\mathbb{S}^{n-1}	The unit sphere $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : x^\top x = 1\}$
$\text{St}(m, r)$	The (compact) Stiefel manifold $\text{St}(m, r) = \{U \in \mathbb{R}^{m \times r} : U^\top U = I_r\}$
$\text{Gr}(m, r)$	The Grassmann manifold of linear subspaces of \mathbb{R}^m of dimension r
$O(n)$	The orthogonal group $O(n) = \{R \in \mathbb{R}^{n \times n} : R^\top R = I_n\}$
$\text{SO}(n)$	The special orthogonal group $\text{SO}(n) = \{R \in O(n) : \det(R) = 1\}$
$\mathfrak{so}(n)$	Lie algebra of $\text{SO}(n)$, i.e., the set of skew-symmetric matrices

Tools on manifolds

$T_x \mathcal{M}$	Tangent space at x to the manifold \mathcal{M}
$\langle u, v \rangle_x, \langle u, v \rangle$	Inner product between tangent vectors $u, v \in T_x \mathcal{M}$
$\ u\ _x, \ u\ $	Norm of the tangent vector u at x , $\ u\ _x = \sqrt{\langle u, u \rangle_x}$
Proj_x	For a Riemannian submanifold, orthogonal projector from the ambient space to the tangent space at x
H_x, V_x	Horizontal and vertical spaces at x to a quotient manifold
$\text{Proj}^h, \text{Proj}^v$	For a Riemannian quotient manifold, orthogonal projectors from the structure space to the horizontal and vertical spaces
R_x	Retraction at x , Definition 2.25
$\text{dist}(x, y)$	Riemannian (or geodesic) distance, Definition 2.22
$\nabla_X Y$	Affine connection on a manifold, typically the Riemannian connection, Definition 2.16
Exp_x	Exponential map at x , Definition 2.23
Log_x	Logarithmic map at x , Definition 2.26
$\text{Transp}_{y \leftarrow x}$	Vector transport from x to y , Definition 2.27
$\mathcal{R}(U, V)$	Riemannian curvature tensor, Definition 2.28

Functions

Id	Identity map
$f \circ g$	Function composition: $(f \circ g)(x) = f(g(x))$
$Df(x)[u]$	Directional derivative of f at x along u , also $D(x \mapsto f(x))(x)[u]$
$\nabla f(x)$	Classical gradient of f , seen as a function in a Euclidean space
$\text{grad}f(x)$	Riemannian gradient of f , w.r.t. the manifold f is defined on
$\nabla^2 f(x)[u]$	Classical Hessian of f at x along u
$\text{Hess}f(x)[u]$	Riemannian Hessian of f at x along the tangent vector u at x
$I_\nu(x)$	Modified Bessel function of the first kind (A.4)
$\mathbb{E}\{Y\}$	Expectation of a random variable Y

Miscellaneous

$x \sim y$	Equivalence relation evaluated for two objects x and y
$[x]$	Equivalence class of x for the equivalence relation \sim
$i \sim j$	For i, j two nodes in a graph, evaluates to true if i and j are connected by an edge
$\sum_{i \sim j}$	Sum over the edges of a graph
$\mathcal{O}(f)$	Complexity class of f (Landau or big-O notation)

Acronyms and abbreviations

CG	Conjugate gradients
SD	Steepest-descent
CRB	Cramér-Rao bound
FIM	Fisher information matrix
ICP	Iterative closest point
i.i.d.	Independent, identically distributed (random variables)
MLE	Maximum likelihood estimator
MSE	Mean squared error
PCA	Principal component analysis
pdf	Probability density function
RCG	Riemannian conjugate gradients
RTR	Riemannian trust-regions
SDP	Semidefinite programming
SDR	Semidefinite relaxation
SNL	Sensor network localization
SNR	Signal to noise ratio
SVD	Singular value decomposition
BLUE	Best linear unbiased estimator
ECTD	Euclidean commute-time distance
LRMC	Low-rank matrix completion
QCQP	Quadratically constrained quadratic program
RMSE	Root mean square error

Chapter 1

Introduction

This thesis is concerned with optimization and estimation on manifolds, that is, on smooth nonlinear spaces. It originates in the study of two estimation problems.

The first problem, low-rank matrix completion (LRMC), is suggested by the study of recommender systems. In such systems, a collection of items are available to users. For example, as popularized by the Netflix prize (Bennett & Lanning, 2007), the items could be movies you rent out and the users could be your customers. Each of your customers rented some of the movies you offer and rated them based on how much they liked them. Your task is to estimate (or predict) how much each of your customers would like each of the movies they did not rate, so as to make a personalized recommendation. If there are m movies and n customers, the ratings may be arranged in an $m \times n$ matrix X . Most entries of X are unknown and the task is to complete it. Of course, unless additional knowledge is brought in, this is an ill-posed task. One popular regularization is to assume X is approximately low-rank. This amounts to saying that there exist a small rank $r \ll m, n$ and matrices $U \in \mathbb{R}^{m \times r}$ and $W \in \mathbb{R}^{r \times n}$ such that $X \approx UW$. One possible interpretation is that there exist a small number r of genres (action, comedy, romance...) such that if for each movie a vector u (a row of U) quantifies how much it belongs to each genre and for each customer their appreciation of these genres is quantified in a vector w (a column of W), then the rating that customer would give to that movie is the inner product $u \cdot w$. This particular formulation of recommender systems thus results in the mathematical problem of finding a matrix of low rank which agrees as well as possible (according to some criterion) with measured entries.

The second problem, synchronization of rotations, follows from the study of 3D scan registration. The goal is to construct a numerical representation of the shape of a physical object, such as a statue for example. To this

end, a 3D scanner can be used. It is a device which, pointed at the statue under some orientation, measures its topography. Naturally, the scanner can only image the visible side of the statue, so that the latter needs to be presented to the scanner under many different orientations. To then obtain a unified representation of the complete object, the different scans must be accurately pieced together, that is, each scan must be rotated and translated appropriately. Known algorithms can detect whether or not two given scans overlap and, if so, output an estimate of their relative alignment. The so-called synchronization task consists in using the collected pairwise relative measurements to estimate the position and orientation of the N individual scans. The nonlinear part of this problem is the estimation of the rotation matrices R_1, \dots, R_N from the measurements of $R_i R_j^{-1}$ for some pairs (i, j) .

Both are nonlinear estimation problems in the sense that the sought parameters belong to a nonlinear space. Furthermore, in both cases the search spaces are smooth: the set of fixed-rank matrices as well as the set of rotations form differentiable manifolds. Many such problems are currently active research topics, see for example metric learning (Bellet *et al.*, 2013), global registration (Chaudhury *et al.*, 2013), structure from motion (Arie-Nachimson *et al.*, 2012), distance matrix completion (Mishra *et al.*, 2011a), cryo-em imaging (Wang *et al.*, 2013), interferometry (Demanet & Jugnon, 2013), phase-less reconstruction (Candès *et al.*, 2012; Waldspurger *et al.*, 2012), subspace tracking (Balzano *et al.*, 2010), independent component analysis (Absil & Gallivan, 2006; Theis *et al.*, 2009), estimation of correlation matrices (Grubišić & Pietersz, 2007), etc. See also the numerous signal processing applications listed in (Smith, 2005).

When facing an estimation problem, two principal questions are of interest. First, how can one design efficient algorithms to perform the estimation? Efficiency can be assessed both in terms of required computational resources and in terms of estimation quality (bias, variance...). Second, what are the fundamental limits on the estimation quality one can hope for? Certainly, in general, when data are corrupted by noise, perfect recovery of the parameters is impossible. Establishing a link between noise level and attainable accuracy therefore provides a meaningful benchmark to compare estimators and, at the same time, is informative with respect to the nature of the problem.

To address the question of building estimators, optimization is often the tool of choice. In optimization, we distinguish between two sorts of solutions: a *global* optimizer is an absolute best, whereas a *local* optimizer is only the best in a neighborhood around itself. Of course, global optimizers are always the target, but in general they are overwhelmingly difficult to find.

Some optimization problems can be solved globally in polynomial time

(up to some precision). We refer to such problems as *tractable*. Among them, spectral formulations (which only call for an eigenvector decomposition or a singular value decomposition (SVD)) are often effective. For example, applying the SVD to the data matrix in LRMC will get you a long way (Chatterjee, 2012). Similarly, computing a few dominant eigenvectors of a well-crafted matrix works wonders on synchronization of rotations (Carmona *et al.*, 2011; Singer, 2011). Another large class of tractable problems includes (well-behaved) convex programs (Nesterov, 2004), among which semidefinite programs (SDP’s) are very popular. In particular, semidefinite relaxations (SDR’s)—whereby one solves a tractable SDP related to a difficult problem in the hope that it will yield valuable information about the latter—play a major role in finding approximate solutions to typically hard problems from the class of quadratically constrained quadratic programs (QCQP’s) (Luo *et al.*, 2010). Synchronization of rotations with a least-squares loss is an example of a nonconvex QCQP. On top of the availability of global solvers for tractable problems, strong theoretical tools have also been developed which can often be used to guarantee the performance of the solutions found with respect to the original estimation task, for example via dual certificates or randomized analysis.

On the downside, spectral and convex relaxations are limited in the classes of loss functions they can accommodate, which may preclude full use of prior knowledge about the noise distribution for example. Additionally, although convex formulations boast a polynomial time complexity, they may not be that efficient. Typical SDP solvers run in no-better than cubic time in the number of variables or constraints, which in a big data world is becoming less affordable despite the increase in available computing power. Another source of inefficiency is the fact that SDR’s for QCQP’s ultimately rely on lifting the problem to a high dimensional search space, where the problem becomes convex. Similarly for LRMC, convex approaches drop the rank constraint (Candès & Recht, 2009) and consequently operate in a much higher-dimensional search space than the target parameter warrants. As a result, both time and space requirements increase significantly.

It is hence a natural undertaking to try and combine cheap tractable relaxations of an estimation task at hand (to overcome locally optimizing traps) with an efficient, more flexible refinement strategy. In this thesis, we resort to, respectively, spectral relaxations and Riemannian optimization.

Riemannian optimization, or optimization on manifolds, is a natural candidate for the design of nonlinear estimation algorithms. By operating directly on the low-dimensional search space, nonlinear as it may be, it is able to keep the computational costs proportionate to the complexity of the sought object. Riemannian optimization generalizes well-known tools from continuous, unconstrained optimization such as gradient descent, Newton

methods, trust-region methods etc. In transitioning from the classical Euclidean case to the realm of Riemannian search spaces, little is lost in the convergence guarantees for these methods. Under essentially the same regularity conditions, global and local convergence results are established, in a mature theory laid out by Absil *et al.* (2008). Obviously, little is gained with respect to the convergence guarantees too: nonconvex optimization problems are still hard to solve and the relevance of the reached optimizers often depends on the quality of the initial guess of the solution.

A downside to the aforementioned blends of cheap relaxations and Riemannian optimization refinements is the absence of generic tools for their theoretical analysis. The OptSpace algorithm for LRMC (Keshavan *et al.*, 2010) is a notable exception. Its authors indeed succeeded in establishing exact and stable recovery guarantees for a method based on a (tweaked) truncated SVD followed by optimization over Grassmann manifolds. In general though, conducting such analyses still proves difficult, in part for lack of dedicated proof techniques. Even assessing numerically whether a global optimizer was reached on a particular problem instance is usually difficult.

To address the question of fundamental accuracy limits, one classical tool is that of Cramér-Rao bounds (CRB's). While well-established for linear estimation problems (Rao, 1945), it is only recently that useful generalizations to the Riemannian setting have been developed, notably through the work of Smith (2005). The resulting lower bounds on the variance of any estimator for an estimation problem are one way to alleviate the lack of theoretical guarantees for an estimation algorithm. Indeed, numerical demonstration that an estimator has the smallest variance possible, while not a formal guarantee of success, delivers some peace of mind. Furthermore, specifically when the bounds are derived in closed-form, they may reveal important information about the structure of the estimation problem at hand.

Both Riemannian optimization and Riemannian estimation, as laid out in (Absil *et al.*, 2008) and in (Smith, 2005), are recent endeavors. As such, their use is not widespread. This is in part due to an entry barrier in the form of differential geometry prerequisites. In this thesis, we contribute to both topics in an effort to lower the barriers.

On the optimization side, we develop Manopt, a toolbox for optimization on manifolds. The toolbox is open-source and can be operated by a user familiar with classical unconstrained optimization even without specific differential geometry background. This software is put to use on the model problems (LRMC and synchronization of rotations) in combination with spectral relaxations, with appreciable results.

On the estimation side, we contribute a specialization of the Riemannian

CRB's to the important cases of Riemannian submanifolds and Riemannian quotient manifolds. This notably elucidates how CRB's may be derived and interpreted when indeterminacies (ambiguities) remain in the estimation task. We apply these bounds to synchronization of rotations, which reveals striking insight into the structure of this problem and into the structure of estimation on graphs in general.

Much of this manuscript reports on collaborative efforts. Throughout, this work is written in “we” even though the group of persons concerned may vary. We identify who intervened in which parts in the outline below.

Outline of the thesis and related publications

The present introduction is followed by an overview of the fundamental tools of differential geometry we use throughout the thesis, see Chapter 2. Then, the thesis is divided in two parts, reflecting its twofold aim.

The first part is concerned with optimization on manifolds. Chapter 3 reviews the general topic of optimization on manifolds and describes two well-established algorithms, namely the Riemannian conjugate gradients method and the Riemannian trust-region method (Absil *et al.*, 2008). It finally introduces Manopt, the Matlab toolbox for optimization on manifolds we developed as part of this thesis. The toolbox is available with documentation at www.manopt.org and is described in a paper accepted for publication (Boumal *et al.*, 2014). This is a collaboration with Pierre-Antoine Absil from UCLouvain and with Bamdev Mishra and Rodolphe Sepulchre from the Université de Liège. The two other chapters in this first part of the thesis present applications of Riemannian optimization we describe momentarily. A third application investigated during this thesis is discrete curve fitting on manifolds. That work is reported in conference proceedings but left out of the present manuscript (Boumal, 2013a; Boumal & Absil, 2011a,b).

Chapter 4 reports on LRMC. We propose an algorithm to tackle LRMC as an optimization problem on the Grassmann manifold, leveraging the generic tools from Chapter 3. The algorithm compares favorably with a number of modern competitors on synthetic data and performs adequately on the Netflix dataset. It is also an original investigation of the broad and modern topic of optimization under low-rank constraints. The original version of this algorithm was presented at NIPS 2011 (Boumal & Absil, 2011c) and is further detailed in an extended technical report (Boumal & Absil, 2012).

Chapter 5 reports on synchronization of rotations. We propose a noise model which allows for outliers and use second-order Riemannian trust-regions for the estimation, following a maximum likelihood principle. A

known spectral relaxation of the problem with performance guarantees (generalized here) is exploited as initial guess. We further explore the method numerically on synthetic data and find that it appears to be efficient, as compared to CRB's developed in the second part. The method is also found to perform well on a 3D scan registration task. The proposed estimator, developed with Pierre-Antoine Absil and Amit Singer from Princeton University, was first presented in a CDC 2013 paper (Boumal *et al.*, 2013b).

The second part is concerned with bounds for estimation on manifolds. Chapter 6 reviews a derivation of the CRB's in the generalized setting of estimation on manifolds, due to Smith (2005). It defines the estimation theoretic tools required to discuss estimation tasks on manifolds and further establishes lower bounds on the variance of unbiased estimators for such tasks. This is the cornerstone to support the other chapters in this part of the thesis.

Chapter 7 derives a version of the CRB's introduced in Chapter 6 specifically aimed at estimation problems on Riemannian submanifolds and Riemannian quotient manifolds. This technical work is essentially necessary to prepare the following chapter, but has the added benefit of shedding light (not for the first time) on the relationship between indeterminacies in estimation problems and rank-deficiency in the Fisher information matrix (FIM). This work appears in the IEEE Transactions on Signal Processing (Boumal, 2013b).

Chapter 8 develops and analyzes CRB's for synchronization of rotations. As such, it rests upon the estimation theoretic formulation of that problem introduced in Chapter 5 and on the adapted bounds developed in Chapter 7. The accuracy one can hope to reach in synchronization tasks is seen to rely on the spectrum of the Laplacian of the measurement graph. This leads to revealing interpretations of the level of difficulty of such problems in terms of random walks. These findings appear in *Information and Inference: a Journal of the IMA*, in collaboration with Pierre-Antoine Absil, Amit Singer and Vincent Blondel from UCLouvain (Boumal *et al.*, 2013a). Given the reliance of the bound on the trace of the pseudoinverse of the Laplacian of the measurement graph, we further investigate what the average bound is if the available measurements are selected at random. This is discussed in collaboration with Xiuyuan Cheng from Princeton University in a technical report, but omitted from the present manuscript (Boumal & Cheng, 2013).

Matlab code for chapters 3, 4, 5 and 8 is available on my personal page, currently hosted at <http://perso.uclouvain.be/nicolas.boumal>.

Chapter 2

Elements of Riemannian geometry

This preliminary chapter gives an overview of essential differential geometric tools we use throughout the thesis. Our work is focused on Riemannian manifolds, for the optimization part as well as for the estimation part. Riemannian manifolds have a rich structure which can often be described in a direct and natural way. However, a proper definition of Riemannian manifolds in general requires a definition of smooth manifolds. In turn, defining smooth manifolds requires notions like charts, atlases and tangent vectors seen as equivalence classes of curves. These definitions seldom (if ever) come up in the main chapters of this work because they tend to be tedious to manipulate. Nevertheless, they constitute the solid ground on which rests our intuitive understanding of the Riemannian geometry of many familiar objects and for which more comfortable tools are described.

Besides covering the fundamental definitions leading to the notion of Riemannian manifold, this chapter deals with useful tools such as notions of calculus on manifolds (gradients, connections, Hessians) as well as the exponential and logarithmic maps and vectors transports which make up for some of the structure lost in transitioning from the realm of linear to nonlinear spaces. Curvature, as a means to quantify departure from flatness, is also addressed. Combined, these tools are instrumental in building generic optimization algorithms and estimation bounds on manifolds, as leveraged in this thesis.

The exposition adopted in this chapter is based on a similar chapter in my master's thesis and is mainly inspired from (Absil *et al.*, 2008). The figures are courtesy of Absil *et al.* (2008). All concepts are well-established, see also (Chavel, 1993; do Carmo, 1992; Lee, 1997; O'Neill, 1983).

2.1 Charts and manifolds

Manifolds are sets that can be locally identified with patches of \mathbb{R}^n . These identifications are called charts. A set of compatible charts that covers the whole set is called an atlas for that set. The set and the atlas together constitute a manifold. More formally:

Definition 2.1 (chart). *Let M be a set. A chart of M is a pair (U, φ) where $U \subset M$ and φ is a bijection between U and an open set of \mathbb{R}^n . U is the chart's domain and n is the chart's dimension. Given $p \in U$, the elements of $\varphi(p) = (x_1, \dots, x_n)$ are called the coordinates of p in the chart (U, φ) .*

Definition 2.2 (compatible charts). *Two charts (U, φ) and (V, ψ) of M , of dimensions n and m respectively, are smoothly compatible (\mathcal{C}^∞ -compatible) if either $U \cap V = \emptyset$ or $U \cap V \neq \emptyset$ and*

- $\varphi(U \cap V)$ is an open set of \mathbb{R}^n ,
- $\psi(U \cap V)$ is an open set of \mathbb{R}^m ,
- $\psi \circ \varphi^{-1} : \varphi(U \cap V) \rightarrow \psi(U \cap V)$ is a smooth diffeomorphism (i.e., a smooth invertible function with smooth inverse).

When $U \cap V \neq \emptyset$, the latter implies $n = m$.

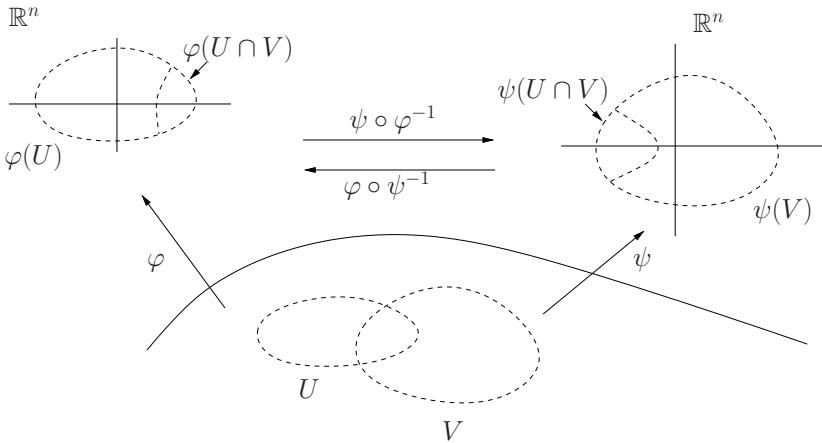


Figure 2.1: Charts. Figure courtesy of Absil *et al.* (2008).

Definition 2.3 (atlas). A set $\mathcal{A} = \{(U_i, \varphi_i), i \in I\}$ of pairwise smoothly compatible charts such that $\cup_{i \in I} U_i = M$ is a smooth atlas of M .

Two atlases \mathcal{A}_1 and \mathcal{A}_2 are compatible if $\mathcal{A}_1 \cup \mathcal{A}_2$ is an atlas. Given an atlas \mathcal{A} , one can generate a unique maximal atlas \mathcal{A}^+ . Such an atlas contains \mathcal{A} as well as all the charts compatible with \mathcal{A} . Classically, we define:

Definition 2.4 (manifold). A smooth manifold is a pair $\mathcal{M} = (M, \mathcal{A}^+)$, where M is a set and \mathcal{A}^+ is a maximal atlas of M .

All the manifolds considered in this work are smooth. Even though charts are a necessary ingredient to define manifolds, they are seldom used in practice when working on a specific manifold. The reason for this is that differential geometric tools are often *coordinate-free*. Coordinate-free means the choice of charts is irrelevant: only the ensuing manifold structure matters. As a result, it is often possible to bypass the explicit definition of charts to describe a manifold.

Example 2.1. The vector space \mathbb{R}^n can be endowed with an obvious manifold structure. Simply consider $\mathcal{M} = (\mathbb{R}^n, \mathcal{A}^+)$ where the atlas \mathcal{A}^+ contains the identity map (\mathbb{R}^n, φ) , $\varphi : U = \mathbb{R}^n \rightarrow \mathbb{R}^n : x \mapsto \varphi(x) = x$.

Often times, we will refer to M when we really mean \mathcal{M} and vice versa. Once the differential manifold structure is clearly stated, no confusion is possible. For example, the notation $\mathcal{M} \subset \mathbb{R}^n$ means $M \subset \mathbb{R}^n$.

Definition 2.5 (dimension). Given a manifold $\mathcal{M} = (M, \mathcal{A}^+)$, if all the charts of \mathcal{A}^+ have the same dimension n , then $\dim \mathcal{M} := n$ is the dimension of the manifold.

All the manifolds considered in this work have a finite dimension. We need one last definition to assess smoothness of curves, functions and maps defined on manifolds.

Definition 2.6 (smooth mapping). Let \mathcal{M} and \mathcal{N} be two smooth manifolds. A mapping $f : \mathcal{M} \rightarrow \mathcal{N}$ is of class \mathcal{C}^k if, for all p in \mathcal{M} , there is a chart (U, φ) of \mathcal{M} and a chart (V, ψ) of \mathcal{N} such that $p \in U$, $f(U) \subset V$ and

$$\psi \circ f \circ \varphi^{-1} : \varphi(U) \rightarrow \psi(V)$$

is of class \mathcal{C}^k , that is, if $\psi \circ f \circ \varphi^{-1}$ is k times continuously differentiable. The latter is called the local expression of f in the charts (U, φ) and (V, ψ) . A smooth map is of class \mathcal{C}^∞ .

This definition does not depend on the choice of charts.

2.2 Tangent spaces and tangent vectors

As is customary in differential geometry, we will define tangent vectors as equivalence classes of curves. This surprising detour from the very simple idea underlying tangent vectors (namely that they *point* in directions one can follow at a given point on a manifold) stems from the lack of a vector space structure. We first construct a simpler definition.

Consider a smooth curve $c: \mathbb{R} \rightarrow \mathcal{M}$. If $\mathcal{M} = \mathbb{R}^n$, one classically defines the derivative of c at $t = 0$ as:

$$c'(0) := \lim_{t \rightarrow 0} \frac{c(t) - c(0)}{t}.$$

Unfortunately, if \mathcal{M} is allowed to be any manifold, the difference appearing in the numerator does not, in general, make sense. For manifolds embedded in \mathbb{R}^n however (such as, e.g., the sphere), we can still make sense of the definition with the appropriate space identifications. A simple definition of tangent spaces in this (limited) setting follows.

Definition 2.7 (tangent spaces for manifolds embedded in \mathbb{R}^n). *Let $\mathcal{M} \subset \mathbb{R}^n$ be a smooth manifold. The tangent space at $x \in \mathcal{M}$, noted $T_x \mathcal{M}$, is the linear subspace of \mathbb{R}^n defined by:*

$$T_x \mathcal{M} = \{v \in \mathbb{R}^n : v = c'(0) \text{ for a smooth } c: \mathbb{R} \rightarrow \mathcal{M} \text{ such that } c(0) = x\}.$$

The dimension of $T_x \mathcal{M}$ is the dimension of a chart of \mathcal{M} containing x .

In general, \mathcal{M} is not embedded in \mathbb{R}^n so that a more general definition of tangent vectors is needed. The following definition does not require the manifold to be embedded in any space. Let \mathcal{M} be a smooth manifold and p be a point on \mathcal{M} . Then,

$$C_p = \{c: I \rightarrow \mathcal{M} : c \in \mathcal{C}^1, 0 \in I \text{ an open interval in } \mathbb{R}, c(0) = p\}$$

is the set of differentiable curves on \mathcal{M} passing through p at $t = 0$. Here, $c \in \mathcal{C}^1$ is to be understood with Definition 2.6, with the obvious manifold structure on open intervals of \mathbb{R} derived from Example 2.1. We define an equivalence relation on C_p , noted \sim . Let (U, φ) be a chart of \mathcal{M} such that $p \in U$ and let $c_1, c_2 \in C_p$. Then, $c_1 \sim c_2$ if and only if $\varphi \circ c_1$ and $\varphi \circ c_2$ have the same derivative at $t = 0$, that is:

$$c_1 \sim c_2 \Leftrightarrow \left. \frac{d}{dt} \varphi(c_1(t)) \right|_{t=0} = \left. \frac{d}{dt} \varphi(c_2(t)) \right|_{t=0}.$$

It is easy to prove that this is independent of the choice of chart.

Definition 2.8 (tangent space, tangent vector). *The tangent space to \mathcal{M} at p , noted $T_p\mathcal{M}$, is the quotient space*

$$T_p\mathcal{M} = C_p / \sim = \{[c] : c \in C_p\}.$$

Given $c \in C_p$, the equivalence class $[c]$ is an element of $T_p\mathcal{M}$ called a tangent vector to \mathcal{M} at p .

The mapping

$$\theta_p^\varphi : T_p\mathcal{M} \rightarrow \mathbb{R}^n : [c] \mapsto \theta_p^\varphi([c]) = \left. \frac{d}{dt} \varphi(c(t)) \right|_{t=0} = (\varphi \circ c)'(0)$$

is bijective and naturally defines a vector space structure over $T_p\mathcal{M}$ as follows:

$$a[c_1] + b[c_2] := (\theta_p^\varphi)^{-1} (a\theta_p^\varphi([c_1]) + b\theta_p^\varphi([c_2])).$$

This structure, again, is independent of the choice of chart. When $\mathcal{M} \subset \mathbb{R}^n$, it is possible to build a vector space isomorphism (i.e., an invertible linear map) proving that the two definitions 2.7 and 2.8 are, essentially, equivalent.

The notion of tangent vector induces a notion of directional derivatives. Let \mathcal{M} be a smooth manifold. A *scalar field* on \mathcal{M} is a smooth function $f : \mathcal{M} \rightarrow \mathbb{R}$. The set of scalar fields on \mathcal{M} is denoted $\mathfrak{F}(\mathcal{M})$.

Definition 2.9 (directional derivative). *The directional derivative of a scalar field f on \mathcal{M} at $p \in \mathcal{M}$ in the direction $\xi = [c] \in T_p\mathcal{M}$ is the scalar:*

$$Df(p)[\xi] := \left. \frac{d}{dt} f(c(t)) \right|_{t=0} = (f \circ c)'(0).$$

The equivalence relation over C_p is specifically crafted so that this definition does not depend on the choice of c , the representative of the equivalence class ξ . In the above notation, the brackets around ξ are a convenient way of denoting that ξ is the direction. They do not mean that we are considering some sort of equivalence class of ξ .

Just like scalar fields associate a scalar to each point of a manifold, it will often times be useful to associate a tangent vector to each point of a manifold. This leads to the definition of vector field.

Definition 2.10 (tangent bundle). *Let \mathcal{M} be a smooth manifold. The tangent bundle, noted $T\mathcal{M}$, is the set:*

$$T\mathcal{M} = \coprod_{p \in \mathcal{M}} T_p\mathcal{M},$$

where \coprod stands for disjoint union. The projection π extracts the root of a vector, that is, $\pi(\xi) = p$ if and only if $\xi \in T_p\mathcal{M}$.

The tangent bundle inherits a smooth manifold structure from \mathcal{M} (Absil *et al.*, 2008, § 3.5.3). This makes it possible to define vector fields on manifolds as smooth mappings from \mathcal{M} to $T\mathcal{M}$, where smoothness is once more understood according to Definition 2.6.

Definition 2.11 (vector field). *A vector field X is a smooth mapping from \mathcal{M} to $T\mathcal{M}$ such that $\pi \circ X = \text{Id}$, the identity map. The vector at p is written X_p or $X(p)$ and lies in $T_p\mathcal{M}$. The set of vector fields on \mathcal{M} is denoted as $\mathfrak{X}(\mathcal{M})$.*

An important example of a vector field is the gradient of a scalar field on a manifold, which we define in the next section and use extensively in optimization algorithms. This will require an additional structure on \mathcal{M} , namely, a Riemannian metric.

2.2.1 Embedded submanifolds

A set N may admit several manifold structures \mathcal{N} . Given a subset $M \subset N$, there may similarly exist several manifold structures for M , but only one of these is such that M is a d -dimensional *embedded submanifold* of \mathcal{N} , as defined in (Absil *et al.*, 2008, Prop. 3.3.2): for each point $x \in M$, there exists a chart (U, φ) of \mathcal{N} such that $M \cap U = \{x \in U : \varphi(x) \in \mathbb{R}^d \times \{0\}\}$. This inherited structure is a strong tie between M and \mathcal{N} . In particular, smooth functions on \mathcal{N} , when restricted to M , become smooth functions on M .

The special case of a smooth manifold \mathcal{M} which is embedded (contained) in a Euclidean space (say, \mathbb{R}^n) is of particular interest in applications. The following theorem shows how to define such manifolds by means of equality constraints on the Cartesian coordinates. This will be one of our favorite tools to describe smooth manifolds without resorting to charts explicitly.

Theorem 2.1. *Let \mathcal{M} be a subset of \mathbb{R}^n . Statements (1) and (2) below are equivalent:*

- (1) *\mathcal{M} is a smooth embedded submanifold of \mathbb{R}^n of dimension $n - m$;*
- (2) *For all $x \in \mathcal{M}$, there is an open set V of \mathbb{R}^n containing x and a smooth function $f : V \rightarrow \mathbb{R}^m$ such that the differential $Df(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ has rank m and $V \cap \mathcal{M} = f^{-1}(0)$.*

Furthermore, the tangent space at x is given by $T_x\mathcal{M} = \ker Df(x)$.

Example 2.2. *An example of a smooth, two-dimensional submanifold of \mathbb{R}^3 is the sphere $\mathbb{S}^2 = \{x \in \mathbb{R}^3 : x^\top x = 1\}$. Use $f : \mathbb{R}^3 \rightarrow \mathbb{R} : x \mapsto f(x) = x^\top x - 1$ in Theorem 2.1. The tangent spaces are then $T_x\mathbb{S}^2 = \{v \in \mathbb{R}^3 : v^\top x = 0\}$ —see Figure 2.2.*

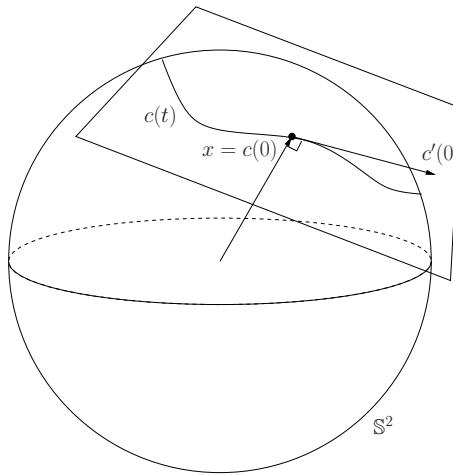


Figure 2.2: Tangent space on the sphere. Since \mathbb{S}^2 is an embedded submanifold of \mathbb{R}^3 , the tangent space $T_x \mathbb{S}^2$ can be pictured as the plane tangent to the sphere at x , with origin at x . Figure courtesy of Absil *et al.* (2008).

2.2.2 Quotient manifolds

Embedded submanifolds can be easily described by means of equality constraints on a structure space. Another convenient way of defining smooth manifolds is by means of equivalence relations.

Let $\overline{\mathcal{M}}$ be a smooth manifold and let \sim define an equivalence relation over $\overline{\mathcal{M}}$. Every point $\bar{x} \in \overline{\mathcal{M}}$ belongs to an equivalence class $[\bar{x}] = \{\bar{y} \in \overline{\mathcal{M}} : \bar{x} \sim \bar{y}\}$. Now consider the quotient space $\mathcal{M} = \overline{\mathcal{M}} / \sim := \{[\bar{x}] : \bar{x} \in \overline{\mathcal{M}}\}$, that is, the set of equivalence classes. That space may in general admit several smooth manifold structures. Let us assume $\dim(\mathcal{M}) < \dim(\overline{\mathcal{M}})$. Under certain conditions, \mathcal{M} admits a unique smooth manifold structure that turns it into a *quotient manifold* of the total or structure space $\overline{\mathcal{M}}$. We leave a proper definition of quotient manifolds to (Absil *et al.*, 2008, § 3.4) and instead focus on one of their instrumental properties.

The natural projection $\pi: \overline{\mathcal{M}} \rightarrow \mathcal{M}$ defined by $\pi(\bar{x}) = [\bar{x}]$ will be useful. If \mathcal{M} is made a quotient manifold of $\overline{\mathcal{M}}$, then the equivalence classes $[\bar{x}] \subset \overline{\mathcal{M}}$ are embedded submanifolds of $\overline{\mathcal{M}}$. This property is depicted in Figure 2.3. This excludes for example discrete symmetries, which declare isolated points of $\overline{\mathcal{M}}$ to be equivalent.

Objects on the quotient manifold such as points and tangent vectors, although well defined, are rather abstract to work with and do not lend themselves to an obvious numerical representation. This is a practically

important point we address now through the definition of *horizontal distributions*.

One way to represent an equivalence class $x \in \mathcal{M}$ in a computer is to store a representation of an arbitrary $\bar{x} \in x$. Then, considering an (abstract) tangent vector $\xi \in T_x \mathcal{M}$, one may represent ξ as a tangent vector $\bar{\xi} \in T_{\bar{x}} \overline{\mathcal{M}}$ which has the same “effect” as ξ in terms of derivations. More precisely, choose any $\bar{\xi}$ such that for all scalar fields f on \mathcal{M} and considering the scalar field $\bar{f} = f \circ \pi: \overline{\mathcal{M}} \rightarrow \mathbb{R}$, the following identity holds: $D\bar{f}(\bar{x})[\bar{\xi}] = Df(x)[\xi]$. Unfortunately, this representation is not unique, notably because the dimension of $T_{\bar{x}} \overline{\mathcal{M}}$ is larger than that of $T_x \mathcal{M}$.

The quotient manifold structure is now leveraged to identify a unique, privileged vector $\bar{\xi}$ as described above, to represent ξ . Since the equivalence class x is an embedded submanifold of $\overline{\mathcal{M}}$, for each $\bar{x} \in x$ it admits a tangent space which is a subspace of the (total) tangent space at \bar{x} in $\overline{\mathcal{M}}$. This special tangent space is called the *vertical space* at \bar{x} :

$$V_{\bar{x}} := T_{\bar{x}}(\pi(\bar{x})) \subset T_{\bar{x}} \overline{\mathcal{M}}.$$

Thus, for each \bar{x} , we may choose a complementary space $H_{\bar{x}} \subset T_{\bar{x}} \overline{\mathcal{M}}$, called the *horizontal space* at \bar{x} , such that

$$T_{\bar{x}} \overline{\mathcal{M}} = V_{\bar{x}} \oplus H_{\bar{x}},$$

where \oplus denotes the direct sum of two subspaces—see Figure 2.3. Notice that this choice is not unique; the chosen mapping H is called a *horizontal distribution* on $\overline{\mathcal{M}}$. There exists a unique horizontal vector $\bar{\xi} \in H_{\bar{x}}$ such that $D\bar{f}(\bar{x})[\bar{\xi}] = Df(x)[\xi]$ for all scalar fields $f \in \mathfrak{F}(\mathcal{M})$. Equivalently, $\bar{\xi}$ is the unique horizontal vector such that $D\pi(\bar{x})[\bar{\xi}] = \xi$ and is called the *horizontal lift* of ξ at \bar{x} .

2.3 Riemannian structure and gradients

Tangent spaces are linear subspaces. Endowing them with inner products provides notions of length and angles on these spaces.

Definition 2.12 (inner product). *Let \mathcal{M} be a smooth manifold and fix $p \in \mathcal{M}$. An inner product $\langle \cdot, \cdot \rangle_p$ on $T_p \mathcal{M}$ is a bilinear, symmetric positive-definite form on $T_p \mathcal{M}$, i.e., $\forall \xi, \zeta, \eta \in T_p \mathcal{M}, a, b \in \mathbb{R}$:*

- $\langle a\xi + b\zeta, \eta \rangle_p = a \langle \xi, \eta \rangle_p + b \langle \zeta, \eta \rangle_p$,
- $\langle \xi, \zeta \rangle_p = \langle \zeta, \xi \rangle_p$, and
- $\langle \xi, \xi \rangle_p \geq 0$, with $\langle \xi, \xi \rangle_p = 0 \Leftrightarrow \xi = 0$.

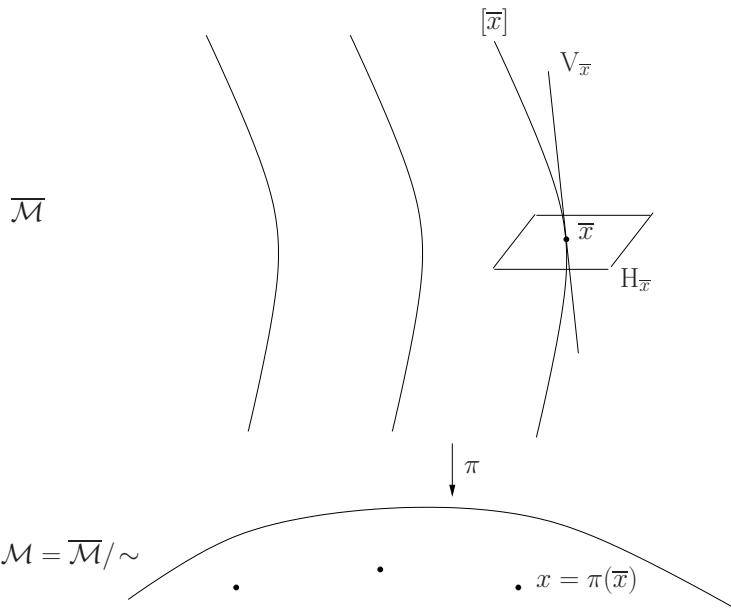


Figure 2.3: Schematic illustration of a quotient manifold. Figure courtesy of Absil *et al.* (2008).

The norm of a tangent vector $\xi \in T_p \mathcal{M}$ is $\|\xi\|_p = \sqrt{\langle \xi, \xi \rangle_p}$.

Often, when it is clear from the context that ξ and η are rooted at p , i.e., $\xi, \eta \in T_p \mathcal{M}$, we write $\langle \xi, \eta \rangle$ instead of $\langle \xi, \eta \rangle_p$. Defining an inner product on all tangent spaces of a smooth manifold in a smooth way defines a Riemannian metric on that manifold.

Definition 2.13 (Riemannian manifold). *A Riemannian manifold is a pair (\mathcal{M}, g) , where \mathcal{M} is a smooth manifold and g is a Riemannian metric. A Riemannian metric is a smoothly varying inner product defined on the tangent spaces of \mathcal{M} , that is, for each $p \in \mathcal{M}$, $g_p(\cdot, \cdot) = \langle \cdot, \cdot \rangle_p$ is an inner product on $T_p \mathcal{M}$.*

In this definition, *smoothly varying* can be understood in the following sense: for all vector fields $X, Y \in \mathfrak{X}(\mathcal{M})$ on \mathcal{M} , the function $p \mapsto g_p(X_p, Y_p)$ is a smooth function from \mathcal{M} to \mathbb{R} . A vector space equipped with an inner product is a special kind of Riemannian manifold called a *Euclidean space*.

As is customary, we will often refer to a Riemannian manifold (\mathcal{M}, g) simply as \mathcal{M} when the metric is clear from the context.

The following definition is of major importance for our purpose. It introduces the notion of gradient of a scalar field on a Riemannian manifold. This constitutes a main reason to require a Riemannian structure in the context of optimization on \mathcal{M} .

Definition 2.14 (gradient). *Let f be a scalar field on a Riemannian manifold \mathcal{M} . The gradient of f at p , denoted by $\text{grad}f(p)$, is defined as the unique element of $T_p\mathcal{M}$ satisfying:*

$$Df(p)[\xi] = \langle \text{grad}f(p), \xi \rangle_p, \quad \forall \xi \in T_p\mathcal{M}.$$

Thus, $\text{grad}f : \mathcal{M} \rightarrow T\mathcal{M}$ is a vector field on \mathcal{M} .

The gradient depends on the Riemannian metric but directional derivatives do not. For a scalar field f on a Euclidean space, $\text{grad}f$ is the usual gradient, which we note ∇f . Remarkably, and similarly to the Euclidean case, the gradient defined above is the steepest-ascent vector field and the norm $\|\text{grad}f(p)\|_p$ is the steepest slope of f at p . More precisely,

$$\|\text{grad}f(p)\|_p = \max_{\xi \in T_p\mathcal{M}, \|\xi\|_p=1} Df(p)[\xi]$$

and $\xi = \text{grad}f(p)/\|\text{grad}f(p)\|_p$ achieves the maximum.

Based on this definition, one privileged way to derive an expression for the gradient of a scalar field f is to work out an expression for the directional derivatives of f , according to Definition 2.9, then to write it as an inner product suitable for direct identification in Definition 2.14. For Riemannian submanifolds and Riemannian quotient manifolds, shortcuts are available. These involve computing classical directional derivatives of matrix functions. Two excellent surveys which can help in this task are the Matrix Cookbook by Petersen & Pedersen (2006) and the Matrix Reference Manual by Brookes (2005), both freely available online.

2.3.1 Riemannian submanifolds

Let $\overline{\mathcal{M}}$ be a Riemannian manifold. Naturally, if $\mathcal{M} \subset \overline{\mathcal{M}}$ is a submanifold of $\overline{\mathcal{M}}$, it can be endowed with a Riemannian structure simply by restricting the metric of $\overline{\mathcal{M}}$ to the tangent spaces of \mathcal{M} .

Definition 2.15 (Riemannian submanifold). *Let $(\overline{\mathcal{M}}, \overline{g})$ be a Riemannian manifold and let (\mathcal{M}, g) be such that \mathcal{M} is a submanifold of $\overline{\mathcal{M}}$ and such that g is the restriction of \overline{g} to the tangent spaces of \mathcal{M} . More precisely, for all $p \in \mathcal{M}$ and for all tangent vectors $\xi, \eta \in T_p\mathcal{M} \subset T_p\overline{\mathcal{M}}$, the metrics g and \overline{g} are compatible in the sense that $g_p(\xi, \eta) = \overline{g}_p(\xi, \eta)$. Then, \mathcal{M} is a Riemannian submanifold of $\overline{\mathcal{M}}$.*

Because an inner product is defined on all of the embedding tangent space $T_p \bar{\mathcal{M}}$, the subspace $T_p \mathcal{M}$ admits an orthogonal complement, called the *normal space*, defined as

$$T_p^\perp \mathcal{M} := \{\xi \in T_p \bar{\mathcal{M}} : \langle \xi, \eta \rangle_p = 0 \ \forall \eta \in T_p \mathcal{M}\}.$$

All vectors of $T_p \bar{\mathcal{M}}$ are uniquely decomposed as $\xi = \text{Proj}_p(\xi) + \text{Proj}_p^\perp(\xi)$ where Proj_p and Proj_p^\perp are orthogonal projectors on the following spaces:

$$\begin{aligned} \text{Proj}_p &: T_p \bar{\mathcal{M}} \rightarrow T_p \mathcal{M} \text{ and} \\ \text{Proj}_p^\perp &: T_p \bar{\mathcal{M}} \rightarrow T_p^\perp \mathcal{M}. \end{aligned}$$

Let \bar{f} be a scalar field on $\bar{\mathcal{M}}$ and let f be its restriction to \mathcal{M} (thus, f is a scalar field on \mathcal{M}). Then,

$$\text{grad} f(p) = \text{Proj}_p \text{grad} \bar{f}(p).$$

Indeed, decomposing $\text{grad} \bar{f}(p)$ into its normal and tangent components, it is not difficult to check that Definition 2.14 holds: for all ξ in $T_p \mathcal{M}$,

$$\begin{aligned} Df(p)[\xi] &= D\bar{f}(x)[\xi] = \left\langle \text{Proj}_p \text{grad} \bar{f}(p) + \text{Proj}_p^\perp \text{grad} \bar{f}(p), \xi \right\rangle_p \\ &= \langle \text{Proj}_p \text{grad} \bar{f}(p), \xi \rangle_p. \end{aligned}$$

In particular, if \mathcal{M} is a Riemannian submanifold of a Euclidean space \mathbb{R}^n , then

$$\text{grad} f(x) = \text{Proj}_x \nabla \bar{f}(x),$$

that is, a classical gradient followed by an orthogonal projection on the tangent space.

Example 2.3 (continued from Example 2.2). *The Riemannian metric on the sphere is obtained by restricting the metric on \mathbb{R}^3 to \mathbb{S}^2 . Hence, for $x \in \mathbb{S}^2$ and $v_1, v_2 \in T_x \mathbb{S}^2$, $\langle v_1, v_2 \rangle_x = v_1^\top v_2$. The orthogonal projector on the tangent space $T_x \mathbb{S}^2$ is $\text{Proj}_x = I - xx^\top$.*

2.3.2 Riemannian quotient manifolds

Let $(\bar{\mathcal{M}}, \bar{g})$ be a Riemannian manifold and let $\mathcal{M} = \bar{\mathcal{M}}/\sim$ be a quotient manifold of $\bar{\mathcal{M}}$. We will now leverage the Riemannian structure of $\bar{\mathcal{M}}$ to equip \mathcal{M} with a Riemannian structure as well. To this end, we first single out one horizontal distribution (see Section 2.2.2) as follows. For all $\bar{x} \in \bar{\mathcal{M}}$,

$$H_{\bar{x}} := V_{\bar{x}}^\perp = \{\bar{\xi} \in T_{\bar{x}} \bar{\mathcal{M}} : \bar{g}_{\bar{x}}(\bar{\xi}, \bar{\eta}) = 0 \ \forall \bar{\eta} \in V_{\bar{x}}\}.$$

Thus, the horizontal lift of an abstract tangent vector $\xi \in T_x \mathcal{M}$ at $\bar{x} \in x$ is the unique horizontal vector $\bar{\xi} \in H_{\bar{x}}$ such that $D\pi(\bar{x})[\bar{\xi}] = \xi$. If for every $x \in \mathcal{M}$ and every $\xi, \eta \in T_x \mathcal{M}$ the inner product $\bar{g}_{\bar{x}}(\bar{\xi}, \bar{\eta})$ does not depend on the choice of lifting point \bar{x} , then

$$g_x(\xi, \eta) := \bar{g}_{\bar{x}}(\bar{\xi}, \bar{\eta})$$

defines a Riemannian metric on \mathcal{M} and (\mathcal{M}, g) is a *Riemannian quotient manifold* of $\overline{\mathcal{M}}$. Furthermore, the canonical projection $\pi: \overline{\mathcal{M}} \rightarrow \mathcal{M}$ is a *Riemannian submersion*, i.e., the restriction of $D\pi(\bar{x})$ to $H_{\bar{x}}$ is an isometry: for all $\bar{\xi}, \bar{\eta} \in H_{\bar{x}}$,

$$\bar{g}_{\bar{x}}(\bar{\xi}, \bar{\eta}) = g_x(D\pi(\bar{x})[\bar{\xi}], D\pi(\bar{x})[\bar{\eta}]).$$

Consider a scalar field f on the quotient space \mathcal{M} . We now demonstrate how to compute a horizontal lift of the gradient of f at $x \in \mathcal{M}$. To this end, choose any scalar field \bar{f} on $\overline{\mathcal{M}}$ such that $\bar{f} = f \circ \pi$. Notice that the directional derivatives of \bar{f} along vertical vectors are necessarily zero:

$$D\bar{f}(\bar{x})[\bar{\xi}] = Df(\pi(\bar{x}))[D\pi(\bar{x})[\bar{\xi}]] = D\pi(x)[0] = 0.$$

Thus, the gradient of \bar{f} is a horizontal vector field: $\forall \bar{x} \in \overline{\mathcal{M}}, \text{grad}\bar{f}(\bar{x}) \in H_{\bar{x}}$. This horizontal vector field is actually the horizontal lift of the gradient of f :

$$\overline{\text{grad}f(x)} = \text{grad}\bar{f}(\bar{x}),$$

where the left hand side denotes the horizontal lift of $\text{grad}f(x)$ at \bar{x} . Indeed, for all $x \in \mathcal{M}$ and $\xi \in T_x \mathcal{M}$ and for any lifting point $\bar{x} \in x$,

$$\begin{aligned} g_x(\text{grad}f(x), \xi) &= g_x(D\pi(\bar{x})[\overline{\text{grad}f(x)}], D\pi(\bar{x})[\bar{\xi}]) \\ &= g_x(D\pi(\bar{x})[\text{grad}\bar{f}(\bar{x})], D\pi(\bar{x})[\bar{\xi}]) \\ &= \bar{g}_{\bar{x}}(\text{grad}\bar{f}(\bar{x}), \bar{\xi}) \\ &= D\bar{f}(\bar{x})[\bar{\xi}] = Df(x)[\xi]. \end{aligned}$$

The orthogonal projectors on the horizontal and vertical spaces at \bar{x} are denoted by, respectively, $\text{Proj}_{\bar{x}}^h: T_{\bar{x}} \overline{\mathcal{M}} \rightarrow H_{\bar{x}}$ and $\text{Proj}_{\bar{x}}^v: T_{\bar{x}} \overline{\mathcal{M}} \rightarrow V_{\bar{x}}$. Orthogonality is understood w.r.t. the metric \bar{g} .

2.4 Connections and Hessians

Let \mathcal{M} be a Riemannian manifold and X, Y be vector fields on \mathcal{M} . We would like to define the derivative of Y at $x \in \mathcal{M}$ along the direction X_x . If \mathcal{M} were a Euclidean space, we would write:

$$DY(x)[X_x] = \lim_{t \rightarrow 0} \frac{Y(x + tX_x) - Y(x)}{t}.$$

Of course, when \mathcal{M} is not a vector space, the above equation does not make sense because $x + tX_x$ is undefined. Furthermore, even if we give meaning to this sum—and we will in Section 2.6— $Y(x + tX_x)$ and $Y(x)$ would not belong to the same vector spaces, hence their difference would be undefined too.

To overcome these difficulties, we need the concept of *connection*. A connection is an additional structure on top of the differentiable manifold structure that, loosely stated, makes it possible to compare vectors in tangent spaces of nearby points. This can be generically defined for manifolds. Since we are mainly interested in Riemannian manifolds, we focus on the so called Riemannian, or Levi-Civita connections. The derivatives defined via these connections are notably interesting because they give a coordinate-free means of defining acceleration along a curve (i.e., the derivative of the velocity vector) as well as the Hessian of a scalar field (i.e., the derivative of the gradient vector field).

We now go over the definition of affine connection for manifolds and the Levi-Civita theorem, specific to Riemannian manifolds. This leads to a notion of Riemannian Hessian. Useful theorems to specialize these notions for Riemannian submanifolds and Riemannian quotient manifolds are then provided.

Definition 2.16 (affine connection). *Let $\mathfrak{X}(\mathcal{M})$ denote the set of smooth vector fields on \mathcal{M} and $\mathfrak{F}(\mathcal{M})$ denote the set of smooth scalar fields on \mathcal{M} . An affine connection ∇ on a manifold \mathcal{M} is a mapping*

$$\nabla: \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \rightarrow \mathfrak{X}(\mathcal{M}): (X, Y) \mapsto \nabla_X Y$$

which satisfies the following properties:

- (1) $\mathfrak{F}(\mathcal{M})$ -linearity in X : $\nabla_{fX+gY} Z = f\nabla_X Z + g\nabla_Y Z$,
- (2) \mathbb{R} -linearity in Y : $\nabla_X (aY + bZ) = a\nabla_X Y + b\nabla_X Z$,
- (3) Product rule (Leibniz' law): $\nabla_X (fY) = (Xf)Y + f\nabla_X Y$,

in which $X, Y, Z \in \mathfrak{X}(\mathcal{M})$, $f, g \in \mathfrak{F}(\mathcal{M})$ and $a, b \in \mathbb{R}$.

The symbol ∇ is pronounced “nabla” or “del”; it is *not* the gradient operator. We have used a standard interpretation of vector fields as derivations on \mathcal{M} . The notation Xf stands for a scalar field on \mathcal{M} such that $Xf(p) = Df(p)[X_p]$. Compare the above properties to the usual properties of derivations in \mathbb{R}^n . Every smooth manifold admits infinitely many affine connections. This approach is called an axiomatization: we state the properties we desire in the definition, then only investigate whether such objects exist.

Definition 2.17 (covariant derivative). *The vector field $\nabla_X Y$ is called the covariant derivative of Y with respect to X for the affine connection ∇ . Since $(\nabla_X Y)_p \in T_p \mathcal{M}$ depends on X only through X_p , we can make sense of the notation $\nabla_\xi Y$ where $\xi \in T_p \mathcal{M}$ as $\nabla_\xi Y = (\nabla_X Y)_p$ for an arbitrary $X \in \mathfrak{X}(\mathcal{M})$ such that $X_p = \xi$.*

At each point $p \in \mathcal{M}$, the vector $(\nabla_X Y)_p$ captures how the vector field Y varies at p along the direction X_p . The following example shows a natural affine connection in Euclidean space.

Example 2.4. *In \mathbb{R}^n , the classical directional derivative defines an affine connection:*

$$(\nabla_X Y)_x = \lim_{t \rightarrow 0} \frac{Y(x + tX_x) - Y(x)}{t} = D Y(x)[X_x].$$

This should give us confidence that Definition 2.16 is a good definition. As often, the added structure of Riemannian manifolds makes for stronger results. The Levi-Civita theorem singles out one particular affine connection for each Riemannian manifold.

Theorem 2.2 (Levi-Civita). *On a Riemannian manifold \mathcal{M} there exists a unique affine connection ∇ that satisfies*

- (1) $\nabla_X Y - \nabla_Y X = [X, Y]$ (symmetry), and
- (2) $Z \langle X, Y \rangle = \langle \nabla_Z X, Y \rangle + \langle X, \nabla_Z Y \rangle$ (compatibility with the Riemannian metric),

for all $X, Y, Z \in \mathfrak{X}(\mathcal{M})$. This affine connection is called the *Levi-Civita connection* or the *Riemannian connection*.

In the above definition, we used the notation $[X, Y]$ for the Lie bracket of X and Y , which is a vector field defined by $[X, Y]f = X(Yf) - Y(Xf)$, $\forall f \in \mathfrak{F}(\mathcal{M})$, again using the interpretation of vector fields as derivations. Not surprisingly, the connection exposed in Example 2.4 is the Riemannian connection on Euclidean spaces for the canonical inner product.

Since connections provide a notion of derivative of a vector field, for a Riemannian manifold we may define a notion of Hessian as the derivative of the gradient vector field.

Definition 2.18 (Riemannian Hessian). *Given a scalar field f on a Riemannian manifold \mathcal{M} equipped with the Riemannian connection ∇ , the Riemannian Hessian of f at a point $x \in \mathcal{M}$ is the linear mapping $\text{Hess } f(x)$ from $T_x \mathcal{M}$ into itself defined by*

$$\text{Hess } f(x)[\xi] = \nabla_\xi \text{grad } f = (\nabla_X \text{grad } f)_x,$$

where X is any vector field on \mathcal{M} such that $X_x = \xi$.

In particular, the Riemannian Hessian is a symmetric operator with respect to the Riemannian metric:

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

For the special cases of Riemannian submanifolds and Riemannian quotient manifolds, connections and Hessians are often simpler to compute than for the general case.

2.4.1 Riemannian submanifolds

The next theorem is an important result about the Riemannian connection of a submanifold of a Riemannian manifold taken from (Absil *et al.*, 2008). This situation is illustrated on Figure 2.4.

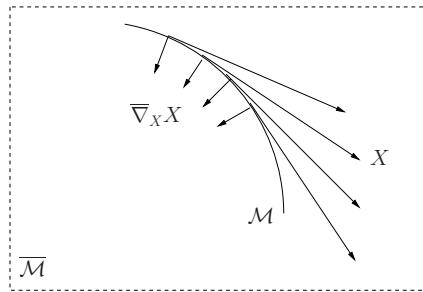


Figure 2.4: Riemannian connection $\bar{\nabla}$ in a Euclidean space $\bar{\mathcal{M}}$ applied to a tangent vector field X to a circle. We observe that $\bar{\nabla}_X X$ is not tangent to the circle, hence simply restricting $\bar{\nabla}$ to the circle is not an option. As Theorem 2.3 shows, we need to project $(\bar{\nabla}_X X)_x$ on the tangent space $T_x \mathcal{M}$ to obtain $(\nabla_X X)_x$. Figure courtesy of Absil *et al.* (2008).

Theorem 2.3. *Let \mathcal{M} be a Riemannian submanifold of a Riemannian manifold $\bar{\mathcal{M}}$ and let ∇ and $\bar{\nabla}$ denote the Riemannian connections on \mathcal{M} and $\bar{\mathcal{M}}$. Then,*

$$(\nabla_X Y)_p = \text{Proj}_p(\bar{\nabla}_X Y)_p$$

for all $X, Y \in \mathfrak{X}(\mathcal{M})$.

In particular, if $\bar{\mathcal{M}}$ is a Euclidean space (Example 2.4), then

$$(\nabla_X Y)_x = \text{Proj}_x(DY(x)[X_x]). \quad (2.1)$$

This means that the Riemannian connection on \mathcal{M} can be computed via a classical directional derivative in the embedding space followed by a projection on the tangent space. Thus, for the Riemannian Hessian it holds

that:

$$\text{Hess}f(x)[\xi] = \text{Proj}_x(D(x \mapsto \text{Proj}_x \nabla f(x))[\xi]),$$

where $\nabla f(x)$ denotes the classical gradient of f seen as a scalar field on the embedding Euclidean space. In other words: compute the classical gradient of f , project it, then compute the classical directional derivative of the result and project it.

2.4.2 Riemannian quotient manifolds

The Riemannian connections of a Riemannian manifold and one of its Riemannian quotient manifolds are tightly related.

Theorem 2.4. *Let $\overline{\mathcal{M}}$ be a Riemannian manifold and $\mathcal{M} = \overline{\mathcal{M}}/\sim$ be a Riemannian quotient manifold of $\overline{\mathcal{M}}$. Let ∇ and $\overline{\nabla}$ be the Riemannian connections on \mathcal{M} and $\overline{\mathcal{M}}$ respectively. Then,*

$$\overline{(\nabla_X Y)_x} = \text{Proj}_{\bar{x}}^h(\overline{\nabla}_{\bar{X}} \bar{Y})_{\bar{x}}$$

for all $X, Y \in \mathfrak{X}(\mathcal{M})$, $x \in \mathcal{M}$, $\bar{x} \in x$. Overlines denote horizontal lifts and $\text{Proj}_{\bar{x}}^h$ is the orthogonal projector onto the horizontal space at \bar{x} .

In particular, if the structure space $\overline{\mathcal{M}}$ is a Euclidean space, this reduces to

$$\overline{(\nabla_X Y)_x} = \text{Proj}_{\bar{x}}^h(D\bar{Y}(\bar{x})[\bar{X}_{\bar{x}}]),$$

that is, a classical directional derivative of the horizontal vector field \bar{Y} followed by a horizontal projection. For the Riemannian Hessian, this is spelled out as:

$$\text{Hess}f(x)[\xi] = \text{Proj}_{\bar{x}}^h(D\bar{\nabla}f(\bar{x})[\bar{\xi}]),$$

where $\bar{\nabla}f$ is the classical gradient of f seen as a function on the total space \mathcal{M} (remember that this is naturally a horizontal vector field). In other words: compute the classical gradient of f , compute its classical directional derivatives and project to the horizontal space.

2.5 Distances and geodesic curves

A characteristic of line segments in \mathbb{R}^n seen as curves with arc-length parameterization is that they have zero acceleration. The next definitions generalize the concept of straight lines, preserving this zero acceleration characteristic, to manifolds.

Let us first introduce a notation for tangent vectors to curves (velocity vectors). Given a curve of class \mathcal{C}^1 , $\gamma: [a, b] \rightarrow \mathcal{M}$, and $t \in [a, b]$, define another such curve on \mathcal{M} by shifting its parameter:

$$\gamma_t : [a - t, b - t] \rightarrow \mathcal{M} : \tau \mapsto \gamma_t(\tau) = \gamma(t + \tau).$$

This curve is such that $\gamma_t(0) = \gamma(t)$. Thus, the equivalence class $[\gamma_t] \in T_{\gamma(t)}\mathcal{M}$ is a vector tangent to γ at time t (Definition 2.8). We propose to write

$$\dot{\gamma}(t) \triangleq [\gamma_t].$$

When using Definition 2.7 for tangent vectors to submanifolds of a Euclidean space \mathbb{R}^n , $\dot{\gamma}(t)$ is identified with $\gamma'(t)$, the classical derivative of γ seen as a curve in \mathbb{R}^n .

Definition 2.19 (acceleration along a curve). *Let \mathcal{M} be a smooth manifold equipped with a connection ∇ . Let $\gamma: I \rightarrow \mathcal{M}$ with I an open interval of \mathbb{R} be a \mathcal{C}^2 curve on \mathcal{M} . The acceleration along γ is given by:*

$$t \mapsto \nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) \in T_{\gamma(t)}\mathcal{M}.$$

The above equation abuses the notation for $\dot{\gamma}$, which is tacitly supposed to be smoothly extended to an arbitrary vector field $X \in \mathfrak{X}(\mathcal{M})$ such that $X_{\gamma(t)} = \dot{\gamma}(t)$ for all t (proceed locally if γ crosses itself). For submanifolds of a Euclidean space \mathbb{R}^n , by equation (2.1) and using Definition 2.7 for tangent vectors, this reduces to:

$$\nabla_{\dot{\gamma}(t)} \dot{\gamma}(t) = \text{Proj}_{\dot{\gamma}(t)} \gamma''(t),$$

where $\gamma''(t)$ is the classical second-derivative of γ seen as a curve in \mathbb{R}^n .

Definition 2.20 (geodesic). *A curve $\gamma: I \rightarrow \mathcal{M}$ with I an open interval of \mathbb{R} is a geodesic if and only if it has zero acceleration on all its domain.*

Notice that the choice of connection ∇ induces a notion of acceleration and hence defines the corresponding geodesics on \mathcal{M} . If \mathcal{M} is a Riemannian manifold and ∇ is the Riemannian connection on \mathcal{M} , then these geodesics have additional extremal properties we outline now.

For Riemannian manifolds \mathcal{M} , the availability of inner products on the tangent spaces makes for an easy definition of curve length and distance.

Definition 2.21 (length of a curve). *The length of a curve of class \mathcal{C}^1 , $\gamma: [a, b] \rightarrow \mathcal{M}$, on a Riemannian manifold (\mathcal{M}, g) , with $\langle \xi, \eta \rangle_p \triangleq g_p(\xi, \eta)$, is defined by*

$$\text{length}(\gamma) = \int_a^b \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle_{\gamma(t)}} dt = \int_a^b \|\dot{\gamma}(t)\|_{\gamma(t)} dt.$$

If \mathcal{M} is a Riemannian submanifold of a Euclidean space \mathbb{R}^n , $\dot{\gamma}(t)$ can be replaced by $\gamma'(t)$.

Definition 2.22 (Riemannian distance). *The Riemannian distance (or geodesic distance) on \mathcal{M} is given by:*

$$\text{dist}: \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}^+: (p, q) \mapsto \text{dist}(p, q) = \inf_{\gamma \in \Gamma} \text{length}(\gamma),$$

where Γ is the set of all C^1 curves $\gamma: [0, 1] \rightarrow \mathcal{M}$ such that $\gamma(0) = p$ and $\gamma(1) = q$.

Under very reasonable conditions (see (Absil *et al.*, 2008, p. 46)), one can show that the Riemannian distance defines a metric. The definition above captures the idea that the distance between two points is the length of the shortest path joining these two points. In a Euclidean space, such a path would simply be the line segment joining the points

For close points, geodesics as defined by the Riemannian connection are shortest paths w.r.t. the Riemannian metric. This is not true for any two points on a geodesic though. Indeed, think of two points on the equator of the unit sphere in \mathbb{R}^3 . The equator itself, parameterized by arc-length, is a geodesic. Following this geodesic, one can join the two points via a path of length r or a path of length $2\pi - r$. Unless $r = \pi$, one of these paths is bound to be suboptimal. Most often, we implicitly consider *minimal geodesics*, that is, geodesics of minimal length.

2.6 Exponential and logarithmic maps

Exponentials are mappings that, given a point x on a manifold and a tangent vector ξ at x , generalize the concept of “ $x + \xi$ ”. In a Euclidean space, the sum $x + \xi$ is a point in space that can be reached by leaving x in the direction ξ and traveling a distance equal to the length of ξ . On a manifold equipped with a connection, $\text{Exp}_x(\xi)$ is a point on the manifold that can be reached by leaving x and moving in the direction ξ while remaining on the manifold. Furthermore, the trajectory followed is a geodesic (zero acceleration). For a Riemannian manifold equipped with the Riemannian connection, the distance traveled equals the norm of ξ .

Definition 2.23 (exponential map). *Let \mathcal{M} be a smooth manifold endowed with a connection ∇ and let $x \in \mathcal{M}$. For every $\xi \in T_x \mathcal{M}$, there exists an open interval $I \ni 0$ and a unique geodesic $\gamma(t; x, \xi): I \rightarrow \mathcal{M}$ such that $\gamma(0) = x$ and $\dot{\gamma}(0) = \xi$. Moreover, we have the homogeneity property $\gamma(t; x, a\xi) = \gamma(at; x, \xi)$. The mapping*

$$\text{Exp}_x: T_x \mathcal{M} \rightarrow \mathcal{M}: \xi \mapsto \text{Exp}_x(\xi) = \gamma(1; x, \xi)$$

is called the exponential map at x . In particular, $\text{Exp}_x(0) = x$, $\forall x \in \mathcal{M}$.

In Section 5.2, the geometry of the group of rotations is introduced. There, it will be noted that the exponential map is explicitly computable using the matrix exponential, whence the name. The domain of definition I of the geodesic $\gamma(t; x, \xi)$ does not necessarily include $t = 1$ for all ξ , so that Exp_x is not necessarily defined over the whole tangent space at x .

Definition 2.24 (geodesically complete manifold). *When for all $x \in \mathcal{M}$, Exp_x is defined over the whole tangent space $T_x \mathcal{M}$, the manifold \mathcal{M} is said to be geodesically complete.*

Exponentials can be expensive to compute. The concept of *retraction* admits a simpler definition which requires neither a connection nor a metric, but still captures the most important aspects of exponentials as far as optimization is concerned. Essentially, we drop the requirement that the trajectory γ be a geodesic, as well as the equality between distance traveled and $\|\xi\|$. Figure 2.5 illustrates the concept.

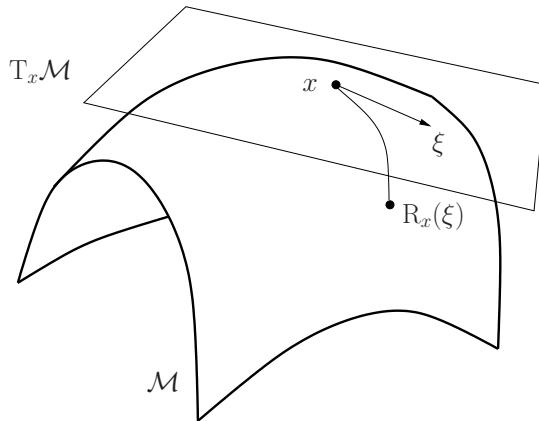


Figure 2.5: Retraction. Figure courtesy of Absil *et al.* (2008).

Definition 2.25 (retraction). *A retraction on a manifold \mathcal{M} is a smooth mapping R from the tangent bundle $T\mathcal{M}$ onto \mathcal{M} with the following properties. For all x in \mathcal{M} , Let R_x denote the restriction of R to $T_x\mathcal{M}$. Then,*

- (1) $R_x(0) = x$, where 0 is the zero element of $T_x\mathcal{M}$, and
- (2) The differential $(DR_x)_0 : T_0(T_x\mathcal{M}) \equiv T_x\mathcal{M} \rightarrow T_x\mathcal{M}$ is the identity map on $T_x\mathcal{M}$, that is, $(DR_x)_0 = \text{Id}$ (local rigidity).

Equivalently, the local rigidity condition can be stated as: $\forall \xi \in T_x \mathcal{M}$, the curve $\gamma_\xi : t \mapsto R_x(t\xi)$ satisfies $\dot{\gamma}_\xi(0) \triangleq [\gamma_\xi] = \xi$. In particular, an exponential map is a retraction. One can think of retractions as mappings that share the important properties we need with the exponential map, while being defined in a flexible enough way that we will be able to propose retractions that are, computationally, cheaper than exponentials. Retractions are the core concept needed to generalize descent algorithms to manifolds.

A related concept is the logarithmic map. Not surprisingly, it is defined as the inverse mapping of the exponential map. For two points x and y , logarithms generalize the concept of “ $y - x$ ”. This is useful notably to define a notion of error vector $\text{Log}_\theta(\hat{\theta})$ in estimation theory, where $\theta \in \mathcal{M}$ is a parameter to estimate and $\hat{\theta} \in \mathcal{M}$ is an estimate of θ . In that context, $\text{Log}_\theta(\hat{\theta})$ is a tangent vector at θ which quantifies the estimation error “ $\hat{\theta} - \theta$ ” in both magnitude and direction.

Definition 2.26 (logarithmic map). *Let \mathcal{M} be a Riemannian manifold. We define*

$$\begin{aligned} \text{Log}_x : \mathcal{M} &\rightarrow T_x \mathcal{M} : y \mapsto \text{Log}_x(y) = \xi, \\ \text{such that } \text{Exp}_x(\xi) &= y \text{ and } \|\xi\|_x = \text{dist}(x, y). \end{aligned}$$

Given a root point x and a target point y , the logarithmic map returns a tangent vector at x pointing toward y and such that $\|\text{Log}_x(y)\| = \text{dist}(x, y)$. As is, this definition is not perfect. There might indeed be more than one eligible ξ . For example, think of the sphere S^2 and place x and y at the poles: for any vector $\eta \in T_x S^2$ such that $\|\eta\| = \pi$, we have $\text{Exp}_x(\eta) = y$. For a more careful definition of the logarithm, see for example (do Carmo, 1992). As long as x and y are not “too far apart”, this definition is satisfactory.

2.7 Parallel translation

In Euclidean spaces, it is natural to compare vectors rooted at different points in space, so much that the notion of root of a vector is utterly unimportant. On manifolds, each tangent vector belongs to a tangent space specific to its root point. Vectors from different tangent spaces cannot be compared immediately. We need a mathematical tool capable of *transporting* vectors between tangent spaces while retaining the information they contain.

The proper tool from differential geometry for this is called *parallel translation*. Let us consider two points $x, y \in \mathcal{M}$, a vector $\xi \in T_x \mathcal{M}$ and a curve γ on \mathcal{M} such that $\gamma(0) = x$ and $\gamma(1) = y$. We introduce X , a vector field defined along the trajectory of γ and such that $X_x = \xi$ and $\nabla_{\dot{\gamma}(t)} X(\gamma(t)) \equiv 0$.

We say that X is *constant* along γ . The transported vector is X_y ; it depends on γ .

In general, computing X_y requires one to solve a differential equation on \mathcal{M} . Just like we introduced retractions as a simpler proxy for exponentials, we now introduce the concept of *vector transport* as a proxy for parallel translation. This concept was first described by Absil *et al.* (2008, § 8.1).

The notion of vector transport defines how to transport a vector $\xi \in T_x\mathcal{M}$ from a point $x \in \mathcal{M}$ to a point $R_x(\eta) \in \mathcal{M}$, $\eta \in T_x\mathcal{M}$. We first introduce the *Whitney sum* then quote the definition of vector transport.

$$T\mathcal{M} \oplus T\mathcal{M} = \{(\eta, \xi) : \eta, \xi \in T_x\mathcal{M}, x \in \mathcal{M}\}$$

Hence $T\mathcal{M} \oplus T\mathcal{M}$ is the set of pairs of tangent vectors belonging to a same tangent space. In the next definition, one of them will be the vector to transport and the other will be the vector along which to transport. This definition is illustrated on Figure 2.6.

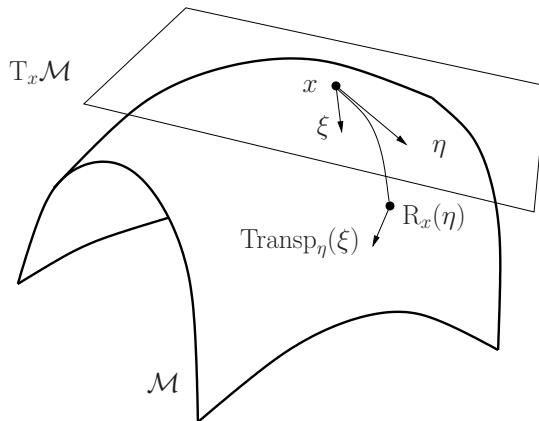


Figure 2.6: Vector transport. Figure courtesy of Absil *et al.* (2008).

Definition 2.27 (vector transport). *A vector transport on a manifold \mathcal{M} is a smooth mapping*

$$\text{Transp}: T\mathcal{M} \oplus T\mathcal{M} \rightarrow T\mathcal{M}: (\eta, \xi) \mapsto \text{Transp}_\eta(\xi)$$

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

- (1) (associated retraction) There exists a retraction R , called the retraction associated with Transp , such that $\text{Transp}_\eta(\xi) \in T_{R_x(\eta)}\mathcal{M}$,
- (2) (consistency) $\text{Transp}_0(\xi) = \xi$ for all $\xi \in T_x\mathcal{M}$,

$$(3) \text{ (linearity)} \quad \text{Transp}_\eta(a\xi + b\zeta) = a\text{Transp}_\eta(\xi) + b\text{Transp}_\eta(\zeta), \quad \forall \eta, \xi, \zeta \in T_x\mathcal{M}, a, b \in \mathbb{R}.$$

This definition is permissive on purpose: it is sufficient to analyze a number of optimization algorithms while authorizing much freedom on the user side.

In this work, we will more often be interested in transporting a vector ξ from a point x to a point y rather than along a vector η . The following notation is more useful in such contexts:

$$\text{Transp}_{y \leftarrow x}(\xi) := \text{Transp}_{R_x^{-1}(y)}(\xi).$$

The mapping $\text{Transp}_{y \leftarrow x} : T_x\mathcal{M} \rightarrow T_y\mathcal{M}$ is a vector transport provided it depends smoothly on x and y , it is linear in ξ and $\text{Transp}_{x \leftarrow x}$ is the identity map.

Example 2.5. *A valid retraction on the sphere \mathbb{S}^2 is given by:*

$$R_x(\eta) = \frac{x + \eta}{\|x + \eta\|}.$$

An associated vector transport is:

$$\text{Transp}_\eta(\xi) = \left(I - \frac{(x + \eta)(x + \eta)^\top}{(x + \eta)^\top(x + \eta)} \right) \xi.$$

On the right-hand side, x, η and ξ are to be treated as elements of \mathbb{R}^3 . Equivalently,

$$\text{Transp}_{y \leftarrow x}(\xi) = (I - yy^\top) \xi = \text{Proj}_y \xi.$$

Thus, ξ is considered as a vector in the ambient space \mathbb{R}^3 and projected orthogonally on the tangent space at y .

Vector transports are notably useful to define the Riemannian conjugate gradients method for optimization, see Section 3.1.

2.8 Curvature

We briefly outline the concept of curvature of a Riemannian manifold. The exposition in this section is limited to a few concepts that come up in the second part of this thesis. The monograph by Lee (1997) offers a thorough introduction to curvature and serves as reference for this section.

A Riemannian manifold \mathcal{M} is *flat* if it is locally isometric to a Euclidean space, that is, if for all x in \mathcal{M} , there exists a neighborhood $U \subset \mathcal{M}$ of x and an isometry $\varphi: U \rightarrow V \subset \mathbb{R}^d$. An isometry preserves distances, that

is, $\text{dist}(x, y) = \|\varphi(x) - \varphi(y)\|$, with $\|\cdot\|$ denoting the Euclidean norm on \mathbb{R}^d . The intuition behind this definition is that a manifold is flat if it can be locally flattened without distortion. Naturally, the sphere $\mathbb{S}^2 \subset \mathbb{R}^3$ with the usual Riemannian submanifold geometry is not flat: cutting out a small piece of an orange peel and trying to flatten it will necessarily result in tearing or wearing. Probably less naturally, a cylinder $\mathbb{R} \times \mathbb{S}^1 \subset \mathbb{R}^3$ with the usual Riemannian submanifold geometry is flat. One way to see this is to notice that, by the above definition, all one-dimensional Riemannian manifolds are flat. Since furthermore a product of flat spaces is flat too, the cylinder must be flat (and the circle \mathbb{S}^1 too).

The arguably counterintuitive notion that circles and cylinders are flat according to the present definition results from the difference between intrinsic and extrinsic curvature. The circle may be embedded in \mathbb{R}^2 in many different ways without changes in notions of distance, thus without changes in its Riemannian structure. Various embeddings may result in various extrinsic (or apparent) curvatures. On the contrary, an imaginary being living on the curve, oblivious to its surroundings (\mathbb{R}^2), would be unable to perceive that curvature (at least locally) because it cannot sense the specific way in which it is embedded in \mathbb{R}^2 . Thus, from an *intrinsic* point of view, the circle has no curvature, and that is what the above definition captures. The sphere \mathbb{S}^2 on the other hand has both extrinsic and intrinsic curvature.

Now consider a Riemannian manifold \mathcal{M} with its Riemannian connection ∇ (Theorem 2.2). Theorem 7.3 in (Lee, 1997) states that \mathcal{M} is flat if and only if for all vector fields X, Y, Z ,

$$\nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z = \nabla_{[X,Y]} Z,$$

where $[X, Y]$ denotes the Lie bracket, as defined below Theorem 2.2. Thus, for vector fields X, Y such that $[X, Y] = 0$, the covariant derivatives commute on a flat manifold. This generalizes the well-known fact that for smooth maps on \mathbb{R}^n , partial derivatives commute.

Consequently, as a means to quantify departure from flatness, the following tensor is defined.

Definition 2.28 (Riemannian curvature tensor). *For any given vector fields X, Y, Z on a Riemannian manifold \mathcal{M} equipped with the Riemannian connection ∇ , the Riemannian curvature tensor $\mathcal{R}: \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \rightarrow \mathfrak{X}(\mathcal{M})$ is defined as*

$$\mathcal{R}(X, Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z.$$

The manifold \mathcal{M} is flat if and only if \mathcal{R} vanishes identically.

The curvature tensor enjoys the following symmetries, as in (Lee, 1997,

Prop. 7.4):

$$\begin{aligned}\mathcal{R}(X, Y) &= -\mathcal{R}(Y, X) \\ \langle \mathcal{R}(X, Y)Z, W \rangle &= -\langle \mathcal{R}(X, Y)W, Z \rangle \\ \mathcal{R}(X, Y)Z + \mathcal{R}(Z, X)Y + \mathcal{R}(Y, Z)X &= 0.\end{aligned}\quad (2.2)$$

Lee (1997) refers to \mathcal{R} as the Riemannian curvature endomorphism and refers to the map from $\mathfrak{X}(\mathcal{M})^4$ to $\mathfrak{F}(\mathcal{M})$

$$(X, Y, Z, W) \mapsto \langle \mathcal{R}(X, Y)Z, W \rangle$$

as the Riemannian curvature tensor instead. Both are linear in each of their (three or four) arguments. The following symmetry follows from the three above:

$$\langle \mathcal{R}(X, Y)Z, W \rangle = \langle \mathcal{R}(Z, W)X, Y \rangle. \quad (2.3)$$

Although it is not directly obvious from the definition of \mathcal{R} , note that the scalar $\langle \mathcal{R}(X, Y)Z, W \rangle_p$ is only a function of X_p, Y_p, Z_p and W_p . Indeed, it is certainly true that the dependence on W is only through W_p . Then, owing to (2.2), it also only depends on Z through Z_p . The symmetry (2.3) similarly shows the dependence on X and Y is limited to X_p and Y_p . See also (O'Neill, 1983, p. 74). This legitimates the notation $\langle \mathcal{R}(x, y)z, w \rangle_p$ in the second part of this thesis, where x, y, z, w are tangent vectors at p . Indeed, this quantity is equal to $\langle \mathcal{R}(X, Y)Z, W \rangle_p$ for any vectors fields X, Y, Z, W such that $X_p = x, Y_p = y, Z_p = z$ and $W_p = w$. The limited dependence also makes it possible to define sectional curvatures as follows.

Definition 2.29 (sectional curvature). *Let $p \in \mathcal{M}$ and X, Y be two vector fields on \mathcal{M} such that X_p, Y_p form a basis of a two-dimensional subspace $\Pi \subset T_p \mathcal{M}$. The sectional curvature of \mathcal{M} associated with Π is defined as the real number*

$$K(\Pi) = K(X_p, Y_p) = \frac{\langle \mathcal{R}(X, Y)Y, X \rangle_p}{\|X_p\|^2 \|Y_p\|^2 - \langle X_p, Y_p \rangle}.$$

In the second part of this thesis, K_{\max} refers to $\max |K(\Pi)|$, where the maximum is taken over all $p \in \mathcal{M}$ and all two-dimensional planes $\Pi \subset T_p \mathcal{M}$.

Part I

Optimization

Chapter 3

Optimization on manifolds

Optimization on manifolds, or Riemannian optimization, is a fast growing research topic in the field of nonlinear optimization. Its purpose is to provide efficient numerical algorithms to find (at least local) optimizers for problems of the form

$$\min_{x \in \mathcal{M}} f(x), \quad (3.1)$$

where the search space \mathcal{M} is a Riemannian manifold, as we defined in Chapter 2. In a nutshell, this means \mathcal{M} can be linearized locally at each point x as a tangent space $T_x \mathcal{M}$ and an inner product $\langle \cdot, \cdot \rangle_x$ which smoothly depends on x is available on $T_x \mathcal{M}$. For example, when \mathcal{M} is a submanifold of $\mathbb{R}^{n \times m}$, a typical inner product is $\langle H_1, H_2 \rangle_X = \text{trace}(H_1^\top H_2)$.

Such geometric structure in an optimization problem originates in mainly two ways. In some scenarios, problem (3.1) is a constrained optimization problem for x in a Euclidean space, say \mathbb{R}^n , such that \mathcal{M} is a smooth submanifold of \mathbb{R}^n . For example, $\mathcal{M} = \{x \in \mathbb{R}^n : x^\top x = 1\}$. In other scenarios, problem (3.1) comes from an unconstrained problem $\min_{u \in \mathbb{R}^n} f(u)$ such that f presents symmetries in the form of an equivalence relation \sim over \mathbb{R}^n : $u \sim v \Rightarrow f(u) = f(v)$. Then, f is constant over equivalence classes $x = [u] = \{v \in \mathbb{R}^n : u \sim v\}$ and descends as a well-defined function over the quotient manifold $\mathcal{M} = \mathbb{R}^n / \sim = \{[u] : u \in \mathcal{M}\}$.

As covered in Chapter 2, the rich geometry of Riemannian manifolds makes it possible to define gradients and Hessians of cost functions f , as well as systematic procedures (called *retractions*) to move on the manifold starting at a point x , along a specified tangent direction at x . Those are sufficient ingredients to generalize standard nonlinear optimization methods such as gradient descent, conjugate-gradients, quasi-Newton, trust-regions, etc.

Building upon many earlier results not reviewed here, the recent monograph by Absil *et al.* (2008) sets an algorithmic framework to analyze problems of the form (3.1) when f is a smooth function, with a strong emphasis on building a theory that leads to efficient numerical algorithms on special manifolds. In particular, it describes the necessary ingredients to design first- and second-order algorithms on Riemannian submanifolds and quotient manifolds of linear spaces. These algorithms come with numerical costs and convergence guarantees essentially matching those of the Euclidean counterparts they generalize. For example, the Riemannian trust-region method converges globally (that is, regardless of the initial iterate) toward critical points and converges locally (that is, once close enough to convergence) quadratically when the Hessian of f is available.

In this chapter, we present two Riemannian optimization methods: the Riemannian conjugate gradients method and the Riemannian trust-region method. Both of these methods are discussed in (Absil *et al.*, 2008, Ch. 8). As a small contribution for these background sections, we give an explicit treatment of preconditioners for these algorithms. This is not new per se, but rarely mentioned explicitly in the Riemannian setting.

The maturity of the theory of smooth Riemannian optimization, its widespread applicability and its excellent track record performance-wise prompted us to build the Manopt toolbox: a user-friendly piece of software to help researchers and practitioners experiment with these tools. Code and documentation are available at www.manopt.org. The last part of this chapter presents Manopt, which we use in the application chapters of this first part of the thesis.

3.1 Riemannian conjugate gradients

When it comes to solving a continuous, unconstrained, nonlinear optimization problem of the form

$$\min_{x \in \mathbb{R}^n} f(x),$$

such that f is continuously differentiable, the steepest descent (SD) or gradient descent method is arguably one of the simplest and most well-known algorithms available. Given an initial guess or initial iterate $x_0 \in \mathbb{R}^n$, it attempts to iteratively improve its predicament by greedily following the most promising direction. More precisely, it generates a sequence of iterates $x_0, x_1, \dots \in \mathbb{R}^n$ according to the update equation

$$x_{k+1} = x_k + \alpha_k d_k, \tag{3.2}$$

where $d_k = -\nabla f(x_k)$ is the steepest-descent direction at x_k and $\alpha_k > 0$ is a well-chosen step size. The nonlinear conjugate gradients (CG) method adds

a sophistication layer to this simple algorithm by constructing an alternative search direction d_k which is a carefully crafted linear combination of both $-\nabla f(x_k)$ and the previous search direction d_{k-1} , thus incorporating a form of inertia in the search procedure:

$$d_k = -\nabla f(x_k) + \beta_{k-1} d_{k-1}. \quad (3.3)$$

SD can be conceived as a special case of CG by letting $\beta_k = 0$ for all k .

From the update equation (3.2) and the search direction equation (3.3), it is apparent that the CG method relies on the vector space structure of \mathbb{R}^n , by composing points and vectors using linear combinations. This dependence is not fundamental though, and both equations can be modified so that they will still make sense for optimization problems of the form (3.1) where the search space \mathcal{M} is a Riemannian manifold. We do need the Riemannian structure so that we have a notion of gradient.

The update equation (3.2) produces x_{k+1} , a new point on the search space, by moving away from x_k along the direction $\alpha_k d_k$. The notion of retraction (Definition 2.25) embodies this very same idea and suggests the more general update formula:

$$x_{k+1} = \text{R}_{x_k}(\alpha_k d_k),$$

where $d_k \in T_{x_k} \mathcal{M}$ is a tangent vector at x_k . Similarly, the search direction equation (3.3) produces the tangent vector d_k by combining two vectors: $-\text{grad}f(x_k)$ and d_{k-1} , where the former is the Riemannian gradient of f at x_k (Definition 2.14). Those are, respectively, tangent vectors at x_k and x_{k-1} . As a result, they cannot be combined directly: they do not belong to the same subspace. One way of fixing this issue is to transport d_{k-1} to x_k using a vector transport (Definition 2.27):

$$d_{k-1}^+ = \text{Transp}_{x_k \leftarrow x_{k-1}}(d_{k-1}).$$

The search direction equation then becomes:

$$d_k = -\text{grad}f(x_k) + \beta_{k-1} d_{k-1}^+.$$

Notice that the vector transport is not needed for the SD method.

A standard trick to accelerate the CG algorithm is to precondition the iterations by operating a change of variables on the tangent spaces $T_{x_k} \mathcal{M}$ (Hager & Zhang, 2006, §8). This change of variable should be chosen so as to decrease the condition number of the Hessian of the cost function. Typically, this is achieved by aiming for a change of variables closely related to the inverse of the Hessian nearby or at a critical point. Of course, a change of variables on $T_{x_k} \mathcal{M}$ amounts to a change of Riemannian metric g_{x_k} , so that it is theoretically sufficient to describe a CG method on

Riemannian manifolds without explicitly allowing for preconditioning. In practice though, it is convenient to separate the work of describing manifolds (giving them a Riemannian structure, defining retractions, geodesics, projectors, etc.) and that of describing a cost function. Since the preconditioner depends on the cost function, we allow for explicit preconditioning of the Riemannian CG method, with the following preconditioner:

$$\text{Precon } f(x) : T_x \mathcal{M} \rightarrow T_x \mathcal{M}.$$

The linear operator $\text{Precon } f(x)$ must be symmetric w.r.t. the Riemannian metric, positive definite and, ideally, be some kind of cheap approximation of $(\text{Hess } f(x))^{-1}$. The search direction equation now reads:

$$d_k = -\text{Precon } f(x_k)[\text{grad } f(x_k)] + \beta_{k-1} d_{k-1}^+.$$

Notice that if $\text{Precon } f(x) = (\text{Hess } f(x))^{-1}$ and $\beta_{k-1} = 0$, this is a Newton step. When no preconditioner is available or necessary, it is replaced by the identity operator.

The step size α_k is chosen by a line search algorithm which approximately solves the one-dimensional optimization problem

$$\min_{\alpha > 0} \phi(\alpha) := f(R_{x_k}(\alpha d_k)). \quad (3.4)$$

If d_k is a descent direction for f (which is typically enforced), then $\phi'(0) < 0$ and it is necessarily possible to decrease ϕ (and hence f) with a positive step size. It does not matter whether we solve (3.4) exactly or not. Typically, it is sufficient to compute a large enough step size such that a *sufficient decrease* is obtained, according to the Armijo criterion:

$$\begin{aligned} f(x_{k+1}) &= \phi(\alpha_k) \leq \phi(0) + c_{\text{decrease}} \alpha_k \phi'(0) \\ &= f(x_k) + c_{\text{decrease}} \cdot Df(x_k)[\alpha_k d_k]. \end{aligned}$$

The constant $0 < c_{\text{decrease}} < 1$ is the sufficient decrease parameter, set to 10^{-4} by default in our case. The simple backtracking line search, Algorithm 2, guarantees this condition is satisfied. Default values for the other parameters, $0 < c_{\text{initial}}, 0 < c_{\text{optimism}}$ and $0 < c_{\text{contraction}} < 1$, are $c_{\text{initial}} = 1$, $c_{\text{optimism}} = 1.1$ and $c_{\text{contraction}} = 0.5$.

The line search problem (3.4) is not any different from the standard line search problem studied in classical textbooks. For example, our line search, Algorithm 2, is based on recommendations in (Nocedal & Wright, 1999, § 3.5). Notice that it is invariant under offsetting and positive scaling of f (assuming a fixed preconditioner). This is a good property: if the cost function changes from $f(x)$ to $8f(x) + 3$, arguably, any reasonable optimization algorithm should still make the same steps. The line search

algorithm is also invariant under rescaling of the search direction d in the following sense: the output α is such that the product αd is not a function of $\|d\|$. Consequently, the combination of Algorithms 1 and 2 as a whole is invariant under offsetting and positive scaling.

For the special case $\beta_k \equiv 0$ (SD), the combination of Algorithms 1 and 2 fits the framework in (Absil *et al.*, 2008, § 4.2). Indeed, noting $\alpha_{k,0}$ the first α tried by Algorithm 2 at iteration k , it is easily checked that $\{\alpha_{k,0}d_k\}$ is a *gradient-related sequence* (Absil *et al.*, 2008, Definition 4.2.1) since $\alpha_{k,0}\|d_k\|$ is bounded away from zero. Corollary 4.3.2 in that reference then guarantees global convergence toward critical points provided the level set $\{x \in \mathcal{M} : f(x) \leq f(x_0)\}$ is compact. Global convergence means that regardless of the initial guess x_0 , in the limit, $\|\text{grad}f(x_k)\|$ goes to zero.

It remains to specify how the inertia parameters β_k are computed. The survey paper by Hager & Zhang (2006) covers a number of suggestions that have appeared in the literature. Those are readily adapted to the Riemannian setting, with special care as outlined in (Hager & Zhang, 2006, § 8) in the presence of a preconditioner. As we already mentioned, the trivial choice $\beta_k = 0$ yields the SD method. A more sophisticated choice known as the modified Hestenes-Stiefel rule is displayed in Algorithm 1. This choice is motivated by its automatic restart property. Indeed, when a negative β_k would be produced (meaning that the next step would revert some of the previous progress), β_k is set to zero instead. This induces a steepest descent step, often considered a restart of the CG algorithm. Refer to (Hager & Zhang, 2006) for more rules together with an analysis of when which rules work best.

Even in the Euclidean case $\mathcal{M} = \mathbb{R}^n$, the convergence analysis of nonlinear CG is not a simple matter, see for example (Gilbert & Nocedal, 1992). In recent work, Sato & Iwai (2013) show how a careful choice of both the β_k coefficients (following the Fletcher-Reeves rule) and the vector transport, together with a line search which satisfies strong Wolfe conditions, can lead to global convergence guarantees for Riemannian CG. The overall algorithm is more involved than the combination of Algorithms 1 and 2 proposed here, especially in its requirements regarding vector transports. In view of the satisfactory numerical performance of the latter combination of algorithms in applications, we choose to carry on with the simple implementation for the present work.

Algorithm 1 RCG : preconditioned Riemannian conjugate gradients

```

1: Given:  $x_0 \in \mathcal{M}$ 
2: Init:  $g_0 = \text{grad } f(x_0), p_0 = \text{Precon } f(x_0)[g_0], d_0 = -p_0, k = 0$ 
3: repeat
4:   if  $\langle g_k, d_k \rangle \geq 0$  then ▷ if  $d_k$  is not a descent direction
5:      $d_k = -p_k$  ▷ restart
6:   end if
7:    $\alpha_k = \text{LINESEARCH}(x_k, d_k, x_{k-1})$  ▷ Armijo backtracking
8:    $x_{k+1} = \text{R}_{x_k}(\alpha_k d_k)$  ▷ make the step
9:    $g_{k+1} = \text{grad } f(x_{k+1})$ 
10:   $p_{k+1} = \text{Precon } f(x_{k+1})[g_{k+1}]$ 
11:   $d_k^+ = \text{Transp}_{x_{k+1} \leftarrow x_k}(d_k)$  ▷ transport to the new tangent space
12:   $g_k^+ = \text{Transp}_{x_{k+1} \leftarrow x_k}(g_k)$ 
13:   $\beta_k = \max(0, \langle g_{k+1} - g_k^+, p_{k+1} \rangle / \langle g_{k+1} - g_k^+, d_k^+ \rangle)$  ▷ HS+
14:   $d_{k+1} = -p_{k+1} + \beta_k d_k^+$  ▷ new search direction
15:   $k = k + 1$ 
16: until a stopping criterion triggers

```

Algorithm 2 Linesearch : modified Armijo backtracking

```

1: Given:  $x \in \mathcal{M}, d \in T_x \mathcal{M}$  (optional:  $x_{\text{prev}} \in \mathcal{M}$ )
2:  $\alpha := \begin{cases} c_{\text{optimism}} \cdot 2^{\frac{f(x) - f(x_{\text{prev}})}{\text{Df}(x)[d]}} & \text{if } x_{\text{prev}} \text{ is available,} \\ c_{\text{initial}} / \|d\| & \text{otherwise.} \end{cases}$ 
3: if  $\alpha \|d\| < 10^{-12}$  then ▷ Make sure  $\alpha$  is neither negative nor too small
4:    $\alpha := c_{\text{initial}} / \|d\|$ 
5: end if
6: while  $f(\text{R}_x(\alpha d)) > f(x) + c_{\text{decrease}} \cdot \text{Df}(x)[\alpha d]$  do
7:    $\alpha := c_{\text{contraction}} \cdot \alpha$ 
8: end while
9: return  $\alpha$ 

```

3.2 Riemannian trust-regions

The Riemannian trust-region (RTR) method (Absil *et al.*, 2007)(Absil *et al.*, 2008, Ch. 7) is a generalization of the classical trust-region optimization scheme (Conn *et al.*, 2000), (Nocedal & Wright, 1999, Ch. 4) to problems of the form (3.1). For smooth cost functions f , the convergence analysis for RTR guarantees global convergence toward critical points (Absil *et al.*, 2007, Thm 4.4, Cor. 4.6). Global convergence means the algorithm converges regardless of the initial iterate. Furthermore, when the true Hessian is available, the local convergence rate is superlinear (Absil *et al.*, 2007, Thm 4.14) (quadratic even, if the parameter θ defined below is set to 1, which is typically the case).

The RTR method is an iterative descent method. Just like the classical trust-region method, it consists in an outer algorithm (Algorithm 3) which uses an inner algorithm (Algorithm 4) to (approximately) minimize a model of the cost function within a trust-region around the current iterate. Depending on the performance of the inner solve, the outer algorithm decides to accept or reject the proposed step, and possibly decides to increase or reduce the size of the trust-region. Similarly to the discussion of the CG algorithm in the previous section, we give a description of the *preconditioned* Riemannian trust-region method. In the absence of a preconditioner, assume Precon $f(x) = \text{Id}$ for all x .

The inner problem at the current iterate $x \in \mathcal{M}$ is the following:

$$\min_{\substack{\eta \in T_x \mathcal{M} \\ \|\eta\|_M \leq \Delta}} m_x(\eta) := f(x) + \langle \eta, \text{grad } f(x) \rangle + \frac{1}{2} \langle \eta, \text{Hess } f(x)[\eta] \rangle, \quad (3.5)$$

where $m_x: T_x \mathcal{M} \rightarrow \mathbb{R}$ is a quadratic model of the *lifted cost* function $f \circ R_x$ defined on the same space and the M -norm on $T_x \mathcal{M}$ is defined via the preconditioner as:

$$\|\eta\|_M^2 := \langle \eta, (\text{Precon } f(x))^{-1}[\eta] \rangle_x.$$

Since the preconditioner is a positive definite operator supposed to resemble the inverse of the Hessian, the trust-region constraint $\|\eta\|_M \leq \Delta$ corresponds more or less to a bound on the quadratic term in m_x . Another point of view is that we *trust* the quadratic model only in a ball of radius Δ , the ball in question being distorted into an ellipsoid by the preconditioner. Because the lifted cost and the model are both defined over a linear subspace, the classical methods to solve this inner problem are available for the task. Algorithm 4 is the truncated Steihaug-Toint method (tCG), as championed in (Absil *et al.*, 2007), based on (Conn *et al.*, 2000, Alg. 7.5.1). The resulting (optimal or suboptimal) vector η is retracted to produce a candidate next iterate $x^+ = R_x(\eta)$. Algorithm 3 dictates when this candidate is accepted.

Algorithm 3 RTR : preconditioned Riemannian trust-region method

```

1: Given:  $x_0 \in \mathcal{M}$ ,  $0 < \Delta_0 \leq \bar{\Delta}$  and  $\rho' > 0$ 
2: Init:  $k = 0$ 
3: repeat
4:    $\eta_k = \text{tCG}(x_k, \Delta_k)$             $\triangleright$  solve inner problem (approximately)
5:    $x_k^+ = R_{x_k}(\eta_k)$                   $\triangleright$  candidate next iterate
6:    $\rho_1 = f(x_k) - f(x_k^+)$               $\triangleright$  actual improvement
7:    $\rho_2 = -\langle \text{grad } f(x_k), \eta_k \rangle - \frac{1}{2} \langle \text{Hess } f(x_k)[\eta_k], \eta_k \rangle$   $\triangleright$  model improv.
8:   if  $\rho_1/\rho_2 < 1/4$  then            $\triangleright$  if the model made a poor prediction
9:      $\Delta_{k+1} = \Delta_k/4$             $\triangleright$  reduce the trust region radius
10:     $\triangleright$  if the model is good but the region is too small
11:   else if  $\rho_1/\rho_2 > 3/4$  and tCG hit the boundary then
12:      $\Delta_{k+1} = \min(2\Delta_k, \bar{\Delta})$         $\triangleright$  enlarge the radius
13:   else
14:      $\Delta_{k+1} = \Delta_k$ 
15:   end if                                 $\triangleright$  if the decrease is sufficient
16:    $x_{k+1} = x_k^+$                        $\triangleright$  accept the step
17:   else                                 $\triangleright$  otherwise
18:      $x_{k+1} = x_k$                        $\triangleright$  reject it
19:   end if
20:    $k = k + 1$ 
21: until a stopping criterion triggers

```

As detailed in the notes following (Conn *et al.*, 2000, Alg. 7.5.1), it is never necessary to apply the inverse of the preconditioner in practice to compute the M -norm: access to Precon $f(x)$ as a black box is sufficient.

RTR requires three parameters. The step acceptance threshold ρ' is set to 0.1 by default. The other two are the maximum and initial trust-region radii, respectively $\bar{\Delta}$ and Δ_0 . The trust-region radius at a given iterate is the upper bound on the M -norm of acceptable steps—see eq. (3.5).

The two parameters for the tCG algorithm are θ and κ (see (Absil *et al.*, 2007)), which we set to 1 and 0.1 respectively by default. These serve in the stopping criterion of tCG. Setting $\theta = 1$ forces a locally quadratic convergence rate for RTR when the true Hessian is available. The number of inner iterations can be limited too.

We note that, close to convergence, the ratio ρ_1/ρ_2 becomes challenging to evaluate accurately, given that both numbers become small and ρ_1 is obtained as the difference between two possibly large numbers. Heuristics such as the one proposed in (Conn *et al.*, 2000, § 17.4.2) address this issue.

Algorithm 4 tCG : Steihaug-Toint truncated CG method

```

1: Given:  $x \in \mathcal{M}$  and  $\Delta, \theta, \kappa > 0$ 
2: Init:  $\eta_0 = 0 \in T_x \mathcal{M}$ ,  $r_0 = \text{grad } f(x)$ ,  $z_0 = \text{Precon } f(x)[r_0]$ ,  $\delta_0 = -z_0$ 
3: for  $k = 0 \dots \text{max inner iterations} - 1$  do
4:    $\kappa_k = \langle \delta_k, \text{Hess } f(x)[\delta_k] \rangle$ 
5:    $\alpha_k = \langle z_k, r_k \rangle / \kappa_k$ 
6:   if  $\kappa_k \leq 0$  or  $\|\eta_k + \alpha_k \delta_k\|_M \geq \Delta$  then
       $\triangleright$  the model Hessian has negative curvature or TR exceeded:
7:     Set  $\tau$  to be the positive root of  $\|\eta_k + \tau \delta_k\|_M^2 = \Delta^2$ ,
       as in (Conn et al., 2000, eqs.(7.5.5–7))
8:      $\eta_{k+1} = \eta_k + \tau \delta_k$   $\triangleright$  hit the boundary
9:     return  $\eta_{k+1}$ 
10:    end if
11:     $\eta_{k+1} = \eta_k + \alpha_k \delta_k$ 
12:     $r_{k+1} = r_k + \alpha_k \text{Hess } f(x)[\delta_k]$ 
13:    if  $\|r_{k+1}\| \leq \|r_0\| \cdot \min(\|r_0\|^\theta, \kappa)$  then
14:      return  $\eta_{k+1}$   $\triangleright$  this approximate solution is good enough
15:    end if
16:     $z_{k+1} = \text{Precon } f(x)[r_{k+1}]$ 
17:     $\beta_k = \langle z_{k+1}, r_{k+1} \rangle / \langle z_k, r_k \rangle$ 
18:     $\delta_{k+1} = -z_{k+1} + \beta_k \delta_k$ 
19:  end for
20:  return  $\eta_{\text{last}}$ 

```

3.3 Manopt, a Matlab toolbox for optimization on manifolds

Manopt is a Matlab toolbox for optimization on manifolds. We started its development at UCL, originally with Pierre Borckmans (UCL) and now actively with Bamdev Mishra (Université de Liège). The toolbox originated as a project of the RANSO group, led by Pierre-Antoine Absil, Yurii Nesterov and Rodolphe Sepulchre. The purpose of Manopt is to facilitate experimentation with optimization on manifolds as well as sharing geometries and algorithms.

The toolbox architecture is based on a separation of the manifolds, the solvers and the problem descriptions. For basic use, one only needs to pick a manifold from the library, describe the cost function (and possible derivatives) on this manifold and pass it on to a solver. Accompanying tools help the user in common tasks such as numerically checking whether the cost function agrees with its derivatives up to the appropriate order, approximating the Hessian based on the gradient of the cost, etc.

Manifolds in Manopt are represented as structures and are obtained by calling a factory. The manifold descriptions include projections on tangent spaces, retractions, helpers to convert Euclidean derivatives (gradient and Hessian) to Riemannian derivatives, etc. See the next section for a list of supported manifolds.

Solvers are functions in Manopt that implement generic Riemannian minimization algorithms. All options have default values. Solvers log standard information at each iteration and comply with standard stopping criteria. Extra information can be logged via callbacks and, similarly, user-defined stopping criteria are allowed. Currently available solvers include Riemannian trust-regions (based on (Absil *et al.*, 2007)) and conjugate-gradients (both with preconditioning), as well as steepest-descent and a couple of derivative-free schemes. More solvers can be added, with an outlook toward BFGS (Ring & Wirth, 2012), stochastic gradients (Bonnabel, 2013), nonsmooth subgradients schemes (Dirr *et al.*, 2007), etc.

An optimization problem in Manopt is represented as a problem structure. The latter includes a field which contains a structure describing a manifold, as obtained from a factory. Additionally, the problem structure hosts function handles for the cost function f and (possibly) its derivatives. An abstraction layer at the interface between the solvers and the problem description offers great flexibility in the cost function description. As the needs grow during the life-cycle of the toolbox and new ways of describing f become necessary (subdifferentials, partial gradients, etc.), it will be sufficient to update this interface.

Computing $f(x)$ typically produces intermediate results which can be reused in order to compute the derivatives of f at x . To prevent redundant computations, Manopt incorporates an (optional) caching system, which becomes useful when transitioning from a proof-of-concept draft of the algorithm to a convincing implementation.

3.3.1 Some supported manifolds

This list of manifolds which work out of the box with the current version of Manopt is intended to give a feeling of the types of optimization problems which can be tackled with Riemannian optimization techniques in general, and with Manopt in particular. More could be added of course, such as the shape space (Ring & Wirth, 2012), the set of low-rank tensors (Kressner *et al.*, 2013), etc. Cartesian products of known manifolds are automatically supported too, via tools named `productmanifold` and `powermanifold`.

- The **oblique manifold**

$$\mathcal{M} = \{X \in \mathbb{R}^{n \times m} : \text{diag}(X^\top X) = \mathbf{1}_m\}$$

is a product of spheres. That is, $X \in \mathcal{M}$ if each column of X has unit 2-norm in \mathbb{R}^n . Absil & Gallivan (2006) show how independent component analysis can be cast on this manifold as non-orthogonal joint diagonalization.

- When furthermore it is only the product $Y = X^\top X$ which matters (with X in the oblique manifold), matrices of the form QX are equivalent for all orthogonal Q . Quotienting out this equivalence relation yields the **fixed-rank ellipope**

$$\mathcal{M} = \{Y \in \mathbb{R}^{m \times m} : Y = Y^\top \succeq 0, \text{rank}(Y) = n, \text{diag}(Y) = \mathbf{1}_m\}.$$

For increasing $n \geq 2$, this yields increasingly relaxed search spaces for max-cut, ultimately culminating in the acclaimed SDP relaxation of max-cut for $n = m$. Journée *et al.* (2010b) show how to exploit this sequence of relaxed formulations of max-cut as Riemannian optimization problems to efficiently compute good cuts. See the example below for application to the max-cut problem. The packing problem on the sphere, where one wishes to place m points on the unit sphere in \mathbb{R}^n such that the two closest points are as far apart as possible (Dirr *et al.*, 2007), is another example of an optimization problem on the fixed-rank ellipope. Grubišić & Pietersz (2007) optimize over this set to produce low-rank approximations of covariance matrices.

- The (compact) **Stiefel manifold** is the Riemannian submanifold of orthonormal matrices,

$$\mathcal{M} = \{X \in \mathbb{R}^{n \times m} : X^\top X = I_m\}.$$

Amari (1999) and Theis *et al.* (2009) formulate versions of independent component analysis with dimensionality reduction as optimization over the Stiefel manifold. Journée *et al.* (2010a) investigate sparse principal component analysis via optimization over the Stiefel manifold.

- The **Grassmann manifold** is the manifold

$$\mathcal{M} = \{\text{col}(X) : X \in \mathbb{R}_*^{n \times m}\},$$

where $\mathbb{R}_*^{n \times m}$ is the set of full-rank matrices in $\mathbb{R}^{n \times m}$ and $\text{col}(X)$ denotes the subspace spanned by the columns of X . That is, $\text{col}(X) \in \mathcal{M}$ is a subspace of \mathbb{R}^n of dimension m . It is often given the geometry of a Riemannian quotient manifold of either $\mathbb{R}_*^{n \times m}$ or of the Stiefel manifold, where two matrices are equivalent if their columns span the same subspace. Among other things, optimization over the Grassmann manifold proves useful in low-rank matrix completion, where it

is observed that if one knows the column space spanned by the sought matrix, then completing the matrix according to a least-squares criterion is easy, see Chapter 4. See also the landmark paper by Edelman *et al.* (1998) for algorithms and applications on both the Stiefel and the Grassmann manifolds.

- **The special orthogonal group**

$$\mathcal{M} = \{X \in \mathbb{R}^{n \times n} : X^\top X = I_n \text{ and } \det(X) = 1\}$$

is the group of rotations, typically considered as a Riemannian submanifold of $\mathbb{R}^{n \times n}$. Optimization problems involving rotation matrices notably occur in robotics and computer vision, when estimating the attitude of vehicles or the pose of cameras, see Chapter 5.

- **The set of fixed-rank matrices**

$$\mathcal{M} = \{X \in \mathbb{R}^{n \times m} : \text{rank}(X) = k\}$$

admits a number of different Riemannian structures. Vandereycken (2013) proposes an embedded geometry for \mathcal{M} and exploits Riemannian optimization on that manifold to address the low-rank matrix completion problem. Shalit *et al.* (2012) use the same geometry to address similarity learning. Mishra *et al.* (2012a) cover a number of quotient geometries for \mathcal{M} and similarly address low-rank matrix completion.

- **Symmetric, positive semidefinite, fixed-rank matrices**

$$\mathcal{M} = \{X \in \mathbb{R}^{n \times n} : X = X^\top \succeq 0, \text{rank}(X) = k\}$$

also form a manifold. Meyer *et al.* (2011b) exploit this to propose low-rank algorithms for metric learning. This space is tightly related to that of **Euclidean distance matrices** X such that X_{ij} is the squared distance between two fixed points $x_i, x_j \in \mathbb{R}^k$. Mishra *et al.* (2011a) leverage this geometry to formulate efficient low-rank algorithms for Euclidean distance matrix completion.

- **The fixed-rank spectrahedron**

$$\mathcal{M} = \{X \in \mathbb{R}^{n \times n} : X = X^\top \succeq 0, \text{trace}(X) = 1 \text{ and } \text{rank}(X) = k\},$$

without the rank constraint, is a convex set which can be used to solve relaxed (lifted) formulations of the sparse PCA problem. Journée *et al.* (2010b) show how optimizing over the fixed-rank spectrahedron can lead to efficient algorithms for sparse PCA.

3.3.2 Example I: the maximum cut problem

Given an undirected graph with n nodes and weights $w_{ij} \geq 0$ on the edges such that $W \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix and $D \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix with $D_{ii} = \sum_j w_{ij}$, the graph Laplacian is the positive semidefinite matrix $L = D - W$. The max-cut problem consists in building a partition $s \in \{+1, -1\}^n$ of the nodes in two classes such that $\frac{1}{4}s^\top L s = \sum_{i < j} w_{ij} \frac{(s_i - s_j)^2}{4}$, i.e., the sum of the weights of the edges connecting the two classes, is maximum. Let $X = ss^\top$. Then, max-cut is equivalent to:

$$\max_{X \in \mathbb{R}^{n \times n}} \text{trace}(LX)/4$$

s.t. $X = X^\top \succeq 0$, $\text{diag}(X) = \mathbf{1}_n$ and $\text{rank}(X) = 1$.

Goemans & Williamson (1995) proposed and analyzed the famous relaxation of this problem which consists in dropping the rank constraint, yielding a semidefinite program. Alternatively relaxing the rank constraint to be $\text{rank}(X) \leq r$ for some $1 < r < n$ yields a tighter but nonconvex relaxation. Journée *et al.* (2010b) observe that fixing the rank with the constraint $\text{rank}(X) = r$ turns the search space into a smooth manifold, the fixed-rank ellipope, which can be optimized over using Riemannian optimization. In Manopt, simple code for this reads (with $Y \in \mathbb{R}^{n \times r}$ such that $X = YY^\top$):

```
% The problem structure hosts a manifold structure as well as
% function handles to define the cost function and its derivatives
% (here provided as Euclidean derivatives, which will be converted
% to their Riemannian equivalent).
problem.M = ellipopefactory(n, r);
problem.cost = @(Y) -trace(Y'*L*Y)/4;
problem.egrad = @(Y) -(L*Y)/2;
problem.ehess = @(Y, U) -(L*U)/2; % optional

% These diagnostics tools help make sure the gradient and Hessian
% are correctly implemented.
checkgradient(problem); pause;
checkhessian(problem); pause;

% Minimize with trust-regions, a random initial guess and default
% options.
Y = trustregions(problem);
```

Randomly projecting Y yields a cut: $s = \text{sign}(Y * \text{randn}(r, 1))$. The Manopt distribution includes advanced code for this example, where the caching functionalities are used to avoid redundant computations of the product LY in the cost and the gradient, and the rank r is increased gradually to obtain a global solution of the max-cut SDP (and hence a formal upperbound), following the procedure in (Journée *et al.*, 2010b).

3.3.3 Example II: sphere packing on the sphere

As a second example, we consider the problem of placing points x_1, \dots, x_n on a sphere $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : x^\top x = 1\}$ such that the two closest points (w.r.t. the geodesic distance dist) are as far apart as possible (Cohn & Kumar, 2007). This problem, known as the Thomson or Tammes problem and also as spherical coding or packing, is directly linked to that of placing as many points as possible on a sphere such that no two points are closer to each other than a given tolerance. Applications may be found in coding theory. In such a setting, one wishes to discretize the sphere such that sending the symbol x_i over a noisy channel, resulting in the receiver receiving $x_i + \text{noise} \in \mathbb{S}^{d-1}$, will as often as possible lead to correct decoding: $\operatorname{argmin}_{x_j \in \{x_1, \dots, x_n\}} \text{dist}(x_i + \text{noise}, x_j) = x_i$.

Formally, the optimization problem is the following structured non-smooth problem:

$$\max_{x_1, \dots, x_n \in \mathbb{S}^{d-1}} \min_{1 \leq i < j \leq n} \text{dist}(x_i, x_j).$$

Dirr *et al.* (2007) address the similar problem of sphere packing on the Grassmannian directly, demonstrating the applicability of Clarke's calculus in a Riemannian setting for nonsmooth optimization problems, resulting in subgradient methods. We follow a different route and propose a smoothing of the cost instead.

First, observe that the geodesic distance $\text{dist}(x_i, x_j) = \arccos(x_i^\top x_j)$ is a strictly decreasing function of $x_i^\top x_j$, such that

$$\min_{x_1, \dots, x_n \in \mathbb{S}^{d-1}} \max_{1 \leq i < j \leq n} x_i^\top x_j \tag{3.6}$$

is an equivalent formulation of the problem. We now resort to the classical log-sum-exp approximation of the max function, based on the following bounds. Let $y_1, \dots, y_m \in \mathbb{R}$ and $y_{\max} = \max_i y_i$. Then, since $y_i - y_{\max} \leq 0$ for all i ,

$$\begin{aligned} y_{\max} &\leq \varepsilon \log \left(\sum_i \exp \left(\frac{y_i}{\varepsilon} \right) \right) = \varepsilon \log \left(\exp \left(\frac{y_{\max}}{\varepsilon} \right) \sum_i \exp \left(\frac{y_i - y_{\max}}{\varepsilon} \right) \right) \\ &\leq \varepsilon \log \left(m \exp \left(\frac{y_{\max}}{\varepsilon} \right) \right) \\ &= y_{\max} + \varepsilon \log(m). \end{aligned}$$

Thus, for a fixed value of $\varepsilon > 0$, the following is a smooth approximation of (3.6) which can be tackled using the optimization algorithms described

in this chapter:

$$\min_{x_1, \dots, x_n \in \mathbb{S}^{d-1}} f(x_1, \dots, x_n) = \varepsilon \log \left(\sum_{1 \leq i < j \leq n} \exp \left(\frac{x_i^\top x_j}{\varepsilon} \right) \right).$$

Let $X \in \mathbb{R}^{n \times d}$ such that x_1, \dots, x_n denote its (unit-norm) rows. We write $f(x_1, \dots, x_n) = f(X)$. Furthermore, it is apparent that f is only a function of XX^\top . Thus, $f(X) = f(XQ)$ for any orthogonal matrix $Q \in O(d)$. Indeed: applying a global rotation to the points on the sphere does not change the distances separating them. The set of acceptable matrices XX^\top is exactly the fixed-rank ellipope from the previous example, thus we optimize f over that manifold. Since f is smooth over this smooth manifold, both RCG and RTR can be used to obtain a sphere packing algorithm.

The code below, run for $d = 3$ and various values of n , generates the configurations depicted in Figure 3.1. We also compare against a collection of best known packings in Figure 3.2 and find that the algorithm attains decent solutions.

```
% Pick a small enough value to get a good approximation of the max
% function, but a large enough value to avoid numerical trouble.
epsilon = 0.0015;

M = elliptopefactory(n, d);

% Define the cost function with caching system used: the store
% structure we receive as input is tied to the input point X.
% Every time this cost function is called at this point X, we
% will receive the same store structure back. We may modify the
% store structure inside the function and return it:
% the changes are remembered for next time.
function [f store] = cost(X, store)
    if ~isfield(store, 'ready')
        XXt = X*X';
        expXXt = exp(XXt/epsilon);
        expXXt(1:(n+1):end) = 0;           % Zero out the diagonal
        u = sum(sum(triu(expXXt, 1)));
        store.XXt = XXt;
        store.expXXt = expXXt;
        store.u = u;
        store.ready = true;
    end
    u = store.u;
    f = epsilon*log(u);
end

% Define the gradient of the cost. When the gradient is called at
% a point X for which the cost was already called, the store
% structure we receive remembers everything that the cost function
% stored in it, so we can reuse previously computed elements.
function [g store] = grad(X, store)
    if ~isfield(store, 'ready')
        [~, store] = cost(X, store);
    end
    % Compute the Euclidean gradient
    eg = store.expXXt*X / store.u;
    % Convert to the Riemannian gradient (by projection)
    g = M.egrad2rgrad(X, eg);
end

% Setup the problem structure with its manifold and cost+grad
problem.M = M;
problem.cost = @cost;
problem.grad = @grad;

% Call a solver on our problem with a few options defined.
% A random initial guess (default) is not too bad for this problem:
% it corresponds to a uniformly random sample on the sphere.
opts.tolgradnorm = 1e-8;
opts.maxtime = 10;
opts.maxiter = 1e3;
X = conjugategradient(problem, [], opts);
```

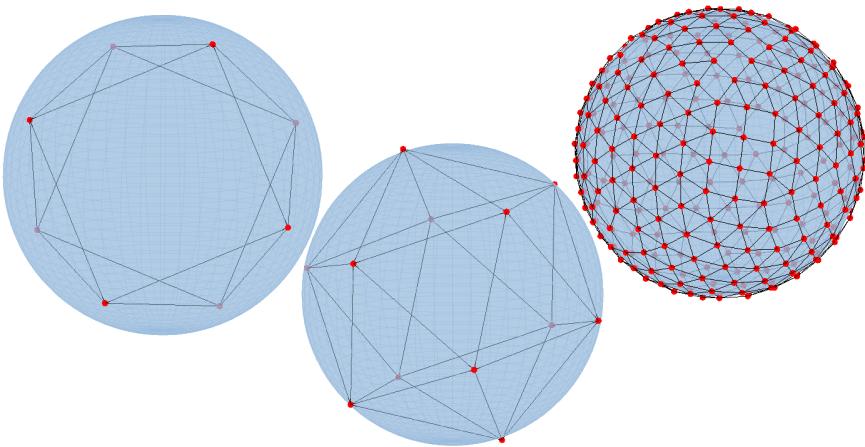


Figure 3.1: Computed sphere packings on the sphere in \mathbb{R}^3 , with $n = 8, 12$ and 300 points. Two vertices are linked by an edge if they are separated by a distance no more than 20% above the smallest distance. For $n = 12$, the solution appears to be an icosahedron, which is a platonic solid. This is not always the case, as the solution for $n = 8$ demonstrates: it is not a cube. The maximum inner products $x_i^\top x_j$ are 0.2615 , 0.4472 and 0.9789 . These packings were produced in 1.0 , 1.7 and 10.2 seconds on a desktop computer from 2010.

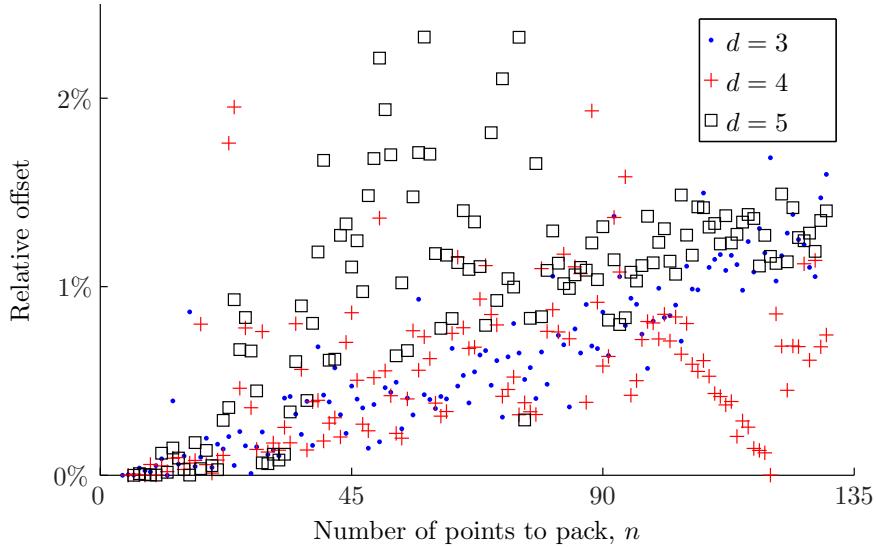


Figure 3.2: On his web page <http://neilsloane.com/packings/>, N.J.A. Sloane collects the best known packings of n points on the sphere in \mathbb{R}^d for $d = 3, 4, 5$ and $n = d+1 \dots 130$ and reports the minimal angle α_{best} between any two points for each of these packings (which is to be maximized). For each pair (d, n) , we compute 5 packings (with a different random initial guess each time) and record the best one, α_{ours} . This figure represents how far away our packings are from the best known ones (on January 7, 2014), as a relative offset: $(\alpha_{\text{best}} - \alpha_{\text{ours}})/\alpha_{\text{best}}$. We never outperform the best ones, but we get close overall. It is interesting to note that for $d = 4$ and $n = 120$, we recover the best known packing, where any two points are separated by an angle of at least 36° . For 95% of the instances, the computation times are well under 20 seconds.

Chapter 4

Low-rank matrix completion

We address the problem of recovering a low-rank $m \times n$ matrix X of which a few entries are observed, possibly with noise. Throughout, we assume that $r = \text{rank}(X) \ll m \leq n$ is known and note $\Omega \subset \{1 \dots m\} \times \{1 \dots n\}$ the set of indices of the observed entries of X , i.e., X_{ij} is known iff $(i, j) \in \Omega$.

It was shown in numerous publications referenced below that low-rank matrix completion is applicable in various situations, notably to build recommender systems. In this setting, the rows of the matrix may correspond to items and the columns may correspond to users. The known entries are the ratings given by users to some items. The aim is to predict the unobserved ratings to generate personalized recommendations. Such applications motivate the study of scalable algorithms given the size of practical instances (billions of entries to predict based on millions of observed entries). However, it is also clear by now, as exemplified by the winning entry of the Netflix prize (Bell *et al.*, 2008), that low-rank matrix completion is not sufficient: effective recommendation systems require more application-specific insight than the sole, algorithmically motivated low-rank prior. This is why the focus of the present chapter is the mathematical problem of low-rank matrix completion and not recommendation systems per se. We propose an algorithm based on Riemannian optimization and test it on synthetic data under different scenarios. Each scenario challenges the proposed method and prior art on a different aspect of the problem, such as the size of the problem, the conditioning of the target matrix, the sampling process, etc. Finally, we demonstrate the applicability of the proposed algorithms on the Netflix dataset.

Some of the more technical parts of this chapter are concerned with the

computation of the Hessian of the cost function we introduce. The derivation of this Hessian is instrumental in developing second-order optimization schemes as well as in deriving an appropriate preconditioner. Nevertheless, these derivations are not important for the implementation of the first-order optimization algorithms investigated. Hence, the technical derivations dedicated to the Hessian may be skipped without loss of continuity.

Related work

In the noiseless case, one could state the minimum rank matrix recovery problem as follows:

$$\min_{\hat{X} \in \mathbb{R}^{m \times n}} \text{rank } \hat{X}, \text{ such that } \hat{X}_{ij} = X_{ij} \quad \forall (i, j) \in \Omega. \quad (4.1)$$

This problem, however, is NP-hard (Candès & Recht, 2009). A possible convex relaxation of (4.1) introduced by Candès & Recht (2009) is to use the nuclear norm of \hat{X} as objective function, i.e., the sum of its singular values, noted $\|\hat{X}\|_*$. The SVT method (Cai *et al.*, 2010) for example attempts to solve such a convex problem using tools from compressed sensing. The ADMiRA method (Lee & Bresler, 2010) does so using matching pursuit-like techniques. One important advantage of proceeding with convex relaxations is that the resulting algorithms can be analyzed thoroughly. In this line of work, a number of algorithms have been proven to attain exact recovery in noiseless scenarios and stable recovery in the face of noise.

In noisy scenarios, one may want to minimize a least-squares data fitting term regularized with a nuclear norm term. For example, NNLS (Toh & Yun, 2010) is an accelerated proximal gradient method for nuclear norm-regularized least-squares problems of this kind.

Jellyfish (Recht *et al.*, 2011) by Recht and Ré is a stochastic gradient method to solve this type of problems on parallel computers. They focus on obviating fine-grained locking, which enables them to tackle very large scale problems. Another parallel approach for very large scale matrix completion is the divide-and-conquer scheme by Mackey *et al.* (2011), where the errors introduced by the division step are statistically described and their effect on the global problem is controlled during the conquer step.

As an alternative to (4.1), one may minimize the discrepancy between \hat{X} and X at entries Ω under the constraint that $\text{rank}(\hat{X}) \leq r$ for some small constant r . Since any matrix \hat{X} of rank at most r may be written in the form UW with $U \in \mathbb{R}^{m \times r}$ and $W \in \mathbb{R}^{r \times n}$, a reasonable formulation of the problem reads:

$$\min_{U \in \mathbb{R}^{m \times r}} \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i, j) \in \Omega} ((UW)_{ij} - X_{ij})^2. \quad (4.2)$$

This is also NP-hard (Gillis & Glineur, 2011), but only requires manipulating small matrices. The LMaFit method (Wen *et al.*, 2012) addresses this problem by alternatively fixing either of the variables and solving the resulting least-squares problem efficiently. The IRLS-M method (Fornasier *et al.*, 2011) similarly proceeds by solving successive least-squares problems.

The factorization of a matrix \hat{X} into the product UW is not unique. Indeed, for any $r \times r$ invertible matrix M , we have $UW = (UM)(M^{-1}W)$. All the matrices UM share the same column space. Hence, the optimal value of the inner optimization problem in (4.2) is a function of $\text{col}(U)$ —the column space of U —rather than U specifically. Dai *et al.* (Dai *et al.*, 2011, 2012) exploit this to recast (4.2) on the Grassmann manifold $\text{Gr}(m, r)$, i.e., the set of r -dimensional linear subspaces of \mathbb{R}^m (see Section 4.1):

$$\min_{\mathcal{U} \in \text{Gr}(m, r)} \min_{W \in \mathbb{R}^{r \times n}} \sum_{(i,j) \in \Omega} ((UW)_{ij} - X_{ij})^2, \quad (4.3)$$

where $U \in \mathbb{R}^{m \times r}$ is any matrix such that $\text{col}(U) = \mathcal{U}$ and is often chosen to be orthonormal. Unfortunately, the objective function of the outer minimization in (4.3) may be discontinuous at points \mathcal{U} for which the least-squares problem in W does not have a unique solution. Dai *et al.* (2011) propose ingenious ways to deal with the discontinuity. Their focus, though, is on deriving theoretical performance guarantees rather than developing fast algorithms. Likewise, Balzano *et al.* (2010) introduce GROUSE, a stochastic gradient descent method for subspace identification, applicable to matrix completion. They also work on a single Grassmannian but with more emphasis on computational efficiency.

Keshavan & Oh (2009) state the problem on the Grassmannian too, but propose to simultaneously optimize on the row and column spaces, yielding a smaller, largely overdetermined least-squares problem which is likely to have a unique solution, resulting in a smooth objective function. In a related paper (Keshavan & Montanari, 2010), they solve:

$$\min_{\mathcal{U} \in \text{Gr}(m, r), \mathcal{V} \in \text{Gr}(n, r)} \min_{S \in \mathbb{R}^{r \times r}} \sum_{(i,j) \in \Omega} ((USV^\top)_{ij} - X_{ij})^2 + \lambda^2 \|USV^\top\|_F^2, \quad (4.4)$$

where U and V are any orthonormal bases of \mathcal{U} and \mathcal{V} , respectively, and λ is a regularization parameter. The authors propose an efficient SVD-based initial guess for \mathcal{U} and \mathcal{V} which they refine using a steepest descent method, along with strong theoretical guarantees. Ngo & Saad (2012) exploit this idea further by applying a Riemannian conjugate gradient method to this formulation. They endow the Grassmannians with a preconditioned metric in order to better capture the conditioning of low-rank matrix completion, with excellent results.

Mishra *et al.* (2011b) propose another geometric approach. They address the problem of low-rank trace norm minimization and propose an algorithm that alternates between fixed-rank optimization and rank-one updates, with applications to low-rank matrix completion.

Vandereycken (2013) investigates a Riemannian conjugate gradient approach based on a submanifold geometry for the manifold of fixed-rank matrices. Meyer *et al.* (2011a) and Absil *et al.* (2013) propose a few quotient geometries for the manifold of fixed-rank matrices. In recent work, Mishra *et al.* (2012b) endow these geometries with preconditioned metrics, akin to the ones developed simultaneously by Ngo & Saad (2012) on a double Grassmannian, and use Riemannian conjugate gradient methods, also with excellent results.

There are many variations on the theme of low-rank matrix completion. For example, Tao & Yuan (2011), among others, focus on identifying a sum of low-rank and sparse matrices as the target matrix. This notably applies to background extraction in videos.

Our contribution and outline of the chapter

Dai et al.’s initial formulation (4.3) has a discontinuous objective function on the Grassmannian. The OptSpace formulation (4.4) on the other hand has a continuous objective, but optimizes on a higher-dimensional search space (two Grassmannians), while it is arguably preferable to keep the dimension of the manifold search space low, even at the expense of a larger least-squares problem. Furthermore, the OptSpace regularization term is efficiently computable since $\|USV^\top\|_F = \|S\|_F$, but it penalizes all entries instead of just the entries $(i, j) \notin \Omega$. The preconditioned metrics introduced in (Mishra *et al.*, 2012b) and (Ngo & Saad, 2012) bring notable improvements, suggesting that one should strive for a formulation which admits efficient preconditioners.

To keep the nonlinear part of the optimization problem small, we favor an approach with a single Grassmannian, that is, we optimize over the column space (which is smallest). This is particularly useful for rectangular matrices ($m \ll n$), which is often the case in applications. We equip (4.3) with a regularization term weighted by $\lambda > 0$ as follows:

$$\min_{\mathcal{U} \in \text{Gr}(m, r)} \min_{W \in \mathbb{R}^{r \times n}} \frac{1}{2} \sum_{(i,j) \in \Omega} C_{ij}^2 ((UW)_{ij} - X_{ij})^2 + \frac{\lambda^2}{2} \sum_{(i,j) \notin \Omega} (UW)_{ij}^2. \quad (4.5)$$

A positive value of λ ensures that the inner least-squares problem has a unique solution which is a continuous function of U , so that the cost function in \mathcal{U} is smooth—see Section 4.2.3. The confidence indices $C_{ij} > 0$ for each observation X_{ij} may be useful in applications. Mathematically, introducing

a regularization term is essential to ensure smoothness of the objective and hence obtain good convergence properties. For real datasets, regularization is practically important, as Section 4.5 demonstrates.

In computing the Hessian of the resulting cost function on $\text{Gr}(m, r)$, it turns out that the latter is cheap to compute, motivating the study of second-order methods. More importantly, having access to an explicit expression for the true Hessian, a simplified analysis is carried out to propose a preconditioner for it. The resulting preconditioner is very similar to the preconditioned metrics proposed in (Mishra *et al.*, 2012b; Ngo & Saad, 2012). The differences are: (i) it operates over a single Grassmannian, and (ii) we do not scale the Riemannian metric, but rather precondition the iterations of the optimization algorithms used. This means the standard geometry of the Grassmannian is all we need.

The cost function is then minimized using a preconditioned Riemannian trust-region method (RTRMCp) or a preconditioned Riemannian conjugate gradient method (RCGMCp), as described in Chapter 3.

Section 4.1 covers essential tools on the Grassmann manifold. Section 4.2 specifies the cost function and develops expressions for its gradient and Hessian, along with a preconditioner for the latter. Section 4.3 details how the optimization algorithms from Chapter 3 are set in place. Sections 4.4 and 4.5 show a few results of numerical experiments demonstrating the effectiveness of the proposed approach.

4.1 Geometry of the Grassmann manifold

We tackle low-rank matrix completion as an optimization problem on the Grassmann manifold. The objective function (which we construct later on) f (4.18) is defined over said manifold $\text{Gr}(m, r)$, the set of r -dimensional linear subspaces of \mathbb{R}^m . Absil *et al.* (2008) give a computation-oriented description of the geometry of this manifold. This section only gives a summary of the required tools. The standard differential geometric concepts used in this chapter are covered in Chapter 2.

Each point $\mathcal{U} \in \text{Gr}(m, r)$ is a linear subspace we may represent numerically as the column space of a full-rank matrix U :

$$\text{Gr}(m, r) = \{\mathcal{U} = \text{col}(U) : U \in \mathbb{R}_*^{m \times r}\}.$$

The notation $\mathbb{R}_*^{m \times r}$ stands for the set of full-rank $m \times r$ matrices. For numerical reasons, we only use orthonormal matrices $U \in \text{St}(m, r)$ to represent subspaces. The set $\text{St}(m, r)$ is the (compact) Stiefel manifold:

$$\text{St}(m, r) = \{U \in \mathbb{R}_*^{m \times r} : U^\top U = I_r\}. \quad (4.6)$$

We view $\text{St}(m, r)$ as a Riemannian submanifold of the Euclidean space $\mathbb{R}^{m \times r}$, endowed with the classical metric $\langle H_1, H_2 \rangle = \text{trace}(H_1^\top H_2)$. We further endow $\text{Gr}(m, r)$ with the unique Riemannian metric such that $\text{Gr}(m, r)$ is a Riemannian quotient manifold of $\text{St}(m, r)$. In other words, the mapping

$$\text{col}: \text{St}(m, r) \rightarrow \text{Gr}(m, r): U \mapsto \text{col}(U) = \text{the column space spanned by } U$$

becomes a Riemannian submersion. The Riemannian metric on $\text{Gr}(m, r)$ is the (essentially) unique metric that is invariant by rotation of \mathbb{R}^m (Leichtweiss, 1961). The submersion col induces an equivalence relation such that U and U' are equivalent if $\text{col}(U) = \text{col}(U')$, that is, if U and U' represent the same column space. Let

$$\text{O}(r) = \{Q \in \mathbb{R}_*^{r \times r} : Q^\top Q = I_r\}$$

denote the set of $r \times r$ orthogonal matrices. Since U and U' are equivalent if and only if there exists some $Q \in \text{O}(r)$ such that $U' = UQ$, we say that $\text{Gr}(m, r)$ is a quotient of $\text{St}(m, r)$ by the action of $\text{O}(r)$:

$$\text{Gr}(m, r) = \text{St}(m, r)/\text{O}(r). \quad (4.7)$$

The Grassmannian is a manifold, and as such admits a tangent space at each point \mathcal{U} , noted $T_{\mathcal{U}}\text{Gr}(m, r)$. The latter is a linear subspace of dimension $\dim \text{Gr}(m, r) = r(m - r)$. A tangent vector $\mathcal{H} \in T_{\mathcal{U}}\text{Gr}(m, r)$, where U represents \mathcal{U} , is represented by a matrix $H \in \mathbb{R}^{m \times r}$ verifying $\frac{d}{dt} \text{col}(U + tH)|_{t=0} = \mathcal{H}$. This representation H of \mathcal{H} , known as its *horizontal lift* at U , is one-to-one if we further impose $U^\top H = 0$. For practical purposes, we often refer to \mathcal{U} and \mathcal{H} using their matrix counterparts U and H instead. This slight abuse of notation has the benefit of making it clearer how one can numerically work with the abstract objects \mathcal{U} and \mathcal{H} . In simplified notation then, the tangent space to $\text{Gr}(m, r)$ at U is the set:

$$T_U\text{Gr}(m, r) = \{H \in \mathbb{R}^{m \times r} : U^\top H = 0\}.$$

Each tangent space is endowed with an inner product (the Riemannian metric) that varies smoothly from point to point. It is inherited from the embedding space $\mathbb{R}^{m \times r}$ of the matrix representation of tangent vectors:

$$\forall H_1, H_2 \in T_U\text{Gr}(m, r), \quad \langle H_1, H_2 \rangle_U = \text{trace}(H_1^\top H_2).$$

The orthogonal projector from $\mathbb{R}^{m \times r}$ onto the tangent space $T_U\text{Gr}(m, r)$ is given by:

$$\text{Proj}_U: \mathbb{R}^{m \times r} \rightarrow T_U\text{Gr}(m, r): H \mapsto \text{Proj}_U H = (I - UU^\top)H.$$

One can similarly define the tangent space at U to the Stiefel manifold:

$$T_U \text{St}(m, r) = \{H \in \mathbb{R}^{m \times r} : U^\top H + H^\top U = 0\}.$$

The projector from the ambient space $\mathbb{R}^{m \times r}$ onto the tangent space of the Stiefel manifold is given by:

$$\begin{aligned} \text{Proj}_U^{\text{St}} : \mathbb{R}^{m \times r} &\rightarrow T_U \text{St}(m, r) \\ &: H \mapsto \text{Proj}_U^{\text{St}} H = (I - UU^\top)H + U \text{skew}(U^\top H), \end{aligned}$$

where $\text{skew}(A) = (A - A^\top)/2$ extracts the skew-symmetric part of A .

We now concern ourselves with the differentiation of functions defined on the Grassmannian. Let \bar{f} be a suitably smooth mapping from $\mathbb{R}_{*}^{m \times r}$ to \mathbb{R} . Let $\bar{f}|_{\text{St}}$ denote its restriction to the Stiefel manifold and let us further assume that

$$\forall U \in \text{St}(m, r), Q \in \text{O}(r), \quad \bar{f}|_{\text{St}}(U) = \bar{f}|_{\text{St}}(UQ).$$

Under this assumption, $\bar{f}|_{\text{St}}$ is only a function of the column space of its argument, hence

$$f : \text{Gr}(m, r) \rightarrow \mathbb{R} : \text{col}(U) \mapsto f(\text{col}(U)) = \bar{f}|_{\text{St}}(U)$$

is well defined. The gradient of f at U is the unique tangent vector $\text{grad } f(U)$ in $T_U \text{Gr}(m, r)$ satisfying

$$\forall H \in T_U \text{Gr}(m, r), \quad \langle \text{grad } f(U), H \rangle_U = Df(U)[H],$$

where $Df(U)[H]$ is the directional derivative of f at U along H ,

$$Df(U)[H] = \lim_{t \rightarrow 0} \frac{f(\text{col}(U + tH)) - f(\text{col}(U))}{t}.$$

Observe that $\text{grad } f(U)$ is an abuse of notation. In fact, $\text{grad } f(U)$ is the so-called horizontal lift of $\text{grad } f(\mathcal{U})$ at U , and the way we abuse notations is justified by the theory of Riemannian submersions, see (Absil *et al.*, 2008, § 3.6.2, § 5.3.4). A similar definition holds for $\text{grad } \bar{f}$ (the usual gradient) and $\text{grad } \bar{f}|_{\text{St}}$. Since $\text{St}(m, r)$ is a Riemannian submanifold of $\mathbb{R}_{*}^{m \times r}$, Section 2.3.1 has it that

$$\text{grad } \bar{f}|_{\text{St}}(U) = \text{Proj}_U^{\text{St}} \text{grad } \bar{f}(U). \tag{4.8}$$

That is, the gradient of the restricted function is obtained by computing the gradient of \bar{f} in the usual way, then projecting the resulting vector onto

the tangent space to the Stiefel manifold. Furthermore, since $\text{Gr}(m, r)$ is a Riemannian quotient manifold of $\text{St}(m, r)$, Section 2.3.2 has it that

$$\text{grad } f(U) = \text{grad } \bar{f}|_{\text{St}}(U). \quad (4.9)$$

The notation $\text{grad } f(U)$ denotes the matrix representation of the abstract tangent vector $\text{grad } f(\text{col}(U))$ with respect to the (arbitrary) choice of orthonormal basis U . They are related by:

$$\frac{d}{dt} \text{col}(U + t \text{grad } f(U)) \Big|_{t=0} = \text{grad } f(\text{col}(U)). \quad (4.10)$$

Notice that $T_U \text{Gr}(m, r)$ is a linear subspace of $T_U \text{St}(m, r)$, so that $\text{Proj}_U \circ \text{Proj}_U^{\text{St}} = \text{Proj}_U$. Since $\text{grad } f(U)$ belongs to $T_U \text{Gr}(m, r)$, it is invariant under Proj_U . Combining (4.8) and (4.9) and applying Proj_U on both sides, we finally obtain a practical means of computing the gradient of f :

$$\text{grad } f(U) = \text{Proj}_U \text{grad } \bar{f}(U) = (I - UU^\top) \text{grad } \bar{f}(U). \quad (4.11)$$

In practice, this means that we need only compute the gradient of \bar{f} in the usual way and then project accordingly.

Similar techniques apply to derive the Hessian of f at U along H in the tangent space $T_U \text{Gr}(m, r)$. Define the vector field $\bar{F}: \mathbb{R}_*^{m \times r} \rightarrow \mathbb{R}^{m \times r}$:

$$\bar{F}(U) = (I - UU^\top) \text{grad } \bar{f}(U).$$

The restriction of \bar{F} to the Stiefel manifold, $\bar{F}|_{\text{St}}$, is a tangent vector field, i.e., $\bar{F}(U) \in T_U \text{St}(m, r)$ for all $U \in \text{St}(m, r)$. Then, for all H in $T_U \text{Gr}(m, r) \subset T_U \text{St}(m, r)$, following Section 2.4.1,

$$\bar{\nabla}_H \bar{F}|_{\text{St}}(U) = \text{Proj}_U^{\text{St}} D\bar{F}(U)[H],$$

where $D\bar{F}(U)[H]$ is the usual directional derivative of \bar{F} at U along H and $\bar{\nabla}_H$ denotes the Levi-Civita connection on the Stiefel manifold w.r.t. any smooth tangent vector field X such that $X_U = H$. This is the analog on manifolds of directional derivatives of vector-valued functions. Furthermore, Section 2.4.2 yields:

$$\text{Hess } f(U)[H] = \text{Proj}_U \bar{\nabla}_H \bar{F}|_{\text{St}}(U),$$

where $\text{Hess } f(U)[H]$ is the derivative at U along H (w.r.t. the Levi-Civita connection on the Grassmannian) of the gradient vector field $\text{grad } f$. Putting these two statements together and remembering that $\text{Proj}_U \circ \text{Proj}_U^{\text{St}} = \text{Proj}_U$, we find a simple expression for the Hessian of f at $\text{col}(U)$ along H w.r.t. the (arbitrary) choice of orthonormal basis U :

$$\text{Hess } f(U)[H] = \text{Proj}_U D\bar{F}(U)[H] = (I - UU^\top) D\bar{F}(U)[H]. \quad (4.12)$$

In practice then, we simply need to differentiate the expression for $\text{grad } f(U)$ “as if it were defined on $\mathbb{R}_*^{m \times r}$ ” and project accordingly.

We use the following retraction (Definition 2.25) on $\text{Gr}(m, r)$ to move away from a given point U along a prescribed direction H while remaining on the manifold:

$$\text{R}_U(H) = \text{polar}(U + H),$$

where $\text{polar}(A) \in \text{St}(m, r)$ designates the $m \times r$ orthonormal factor of the polar decomposition of $A \in \mathbb{R}^{m \times r}$. This is computed using the thin SVD for example: $A = U\Sigma V^\top$ and $\text{polar}(A) = UV^\top$. In abstract terms, this corresponds to having $\text{col}(\text{R}_U(H)) = \text{col}(U + H)$. For tangent vectors, $U^\top H = 0$ so that $U + H$ is always full rank and this is well-defined. Notice that if H is a tangent vector at U such that $\text{R}_U(H) = V$, then $\text{R}_{UQ}HQ = VQ$ for all orthogonal matrices Q .

For the Riemannian conjugate gradient method (Section 3.1) it is necessary to compare vectors belonging to different tangent spaces. Typically, this happens when one wants to combine the gradient at the present iterate with the search direction followed at the previous iterate. One proper way of achieving this is to use a vector transport in accordance with the chosen retraction, see Definition 2.27. We use the following simple procedure to transport a tangent vector H at U to the tangent space at V :

$$\text{Transp}_{V \leftarrow U}(H) = (I - VV^\top)H. \quad (4.13)$$

It is readily checked that $\text{Transp}_{VQ \leftarrow UQ}(HQ) = (\text{Transp}_{V \leftarrow U}(H))Q$ for all orthogonal matrices Q . This invariance property guarantees that (4.13) consistently induces a vector transport on the Grassmann manifold.

4.2 The cost function and its derivatives

We seek an $m \times n$ matrix \hat{X} of rank at most r (and usually exactly r) such that \hat{X} agrees as much as possible with a matrix X whose entries at the observation set Ω are given. Furthermore, we are given a weight matrix $C \in \mathbb{R}^{m \times n}$ indicating the confidence we have in each observed entry of X . The matrix C is positive at entries in Ω and zero elsewhere. To this end, we propose to minimize the following cost function w.r.t. $U \in \mathbb{R}_*^{m \times r}$ and $W \in \mathbb{R}^{r \times n}$, where $(X_\Omega)_{ij}$ equals X_{ij} if $(i, j) \in \Omega$ and is zero otherwise:

$$\begin{aligned} g: \mathbb{R}_*^{m \times r} \times \mathbb{R}^{r \times n} &\rightarrow \mathbb{R} \\ : (U, W) &\mapsto g(U, W) = \frac{1}{2} \|C \odot (UW - X_\Omega)\|_\Omega^2 + \frac{\lambda^2}{2} \|UW\|_{\Omega_c}^2. \end{aligned} \quad (4.14)$$

The notation \odot denotes the entry-wise product, $\lambda > 0$ is a regularization parameter, Ω_c is the complement of the set Ω and

$$\|M\|_\Omega^2 \triangleq \sum_{(i,j) \in \Omega} M_{ij}^2.$$

The interpretation is as follows: we are looking for an optimal matrix $\hat{X} = UW$ of rank at most r ; we have confidence C_{ij} that \hat{X}_{ij} should equal X_{ij} for $(i, j) \in \Omega$ and smaller confidence λ that \hat{X}_{ij} should equal 0 for $(i, j) \notin \Omega$.

For a fixed U , computing the matrix W that minimizes (4.14) is a least-squares problem. As we shall see in Section 4.2.3, the solution to that problem exists and is unique since we assume $\lambda > 0$. Let us note $g_U(W) \triangleq g(U, W)$. The mapping between U and this unique optimal W ,

$$W_U = W(U) = \underset{W \in \mathbb{R}^{r \times n}}{\operatorname{argmin}} g_U(W),$$

is smooth and easily computable—see Section 4.2.3. It is thus natural to consider the following cost function defined over the set of full-rank matrices $U \in \mathbb{R}_*^{m \times r}$:

$$\hat{f}: \mathbb{R}_*^{m \times r} \rightarrow \mathbb{R}: U \mapsto \hat{f}(U) = \frac{1}{2} \|C \odot (UW_U - X_\Omega)\|_\Omega^2 + \frac{\lambda^2}{2} \|UW_U\|_\Omega^2. \quad (4.15)$$

By virtue of the discussion in the introduction of this chapter, we expect that the function \hat{f} be constant over sets of full-rank matrices U spanning the same column space. Let

$$\operatorname{GL}(r) = \{M \in \mathbb{R}^{r \times r} : M \text{ is invertible}\}$$

denote the general linear group. The following holds:

$$\forall M \in \operatorname{GL}(r), \quad W_{UM} = M^{-1}W_U.$$

Indeed, since $g(U, W)$ merely depends on the product UW , for any $M \in \operatorname{GL}(r)$ we have that $g_U(W)$ and $g_{UM}(M^{-1}W)$ are two identical functions of W . Hence, since W_U is the unique minimizer of g_U , it holds that $M^{-1}W_U$ is the unique minimizer of g_{UM} , i.e., $W_{UM} = M^{-1}W_U$. As a consequence, $\hat{X} = UW_U = (UM)W_{UM}$ for all $M \in \operatorname{GL}(r)$. For such matrices M , it then follows as expected that:

$$\hat{f}(UM) = \hat{f}(U).$$

This induces an equivalence relation \sim over the $m \times r$ matrices of full rank, $\mathbb{R}_*^{m \times r}$. Two such matrices are equivalent if and only if they have the same column space:

$$U \sim U' \Leftrightarrow \exists M \in \operatorname{GL}(r) \text{ s.t. } U' = UM \Leftrightarrow \operatorname{col}(U) = \operatorname{col}(U').$$

This is equivalent to stating that U and U' are equivalent if they lead to the same reconstruction model $\hat{X} = UW_U = U'W_{U'}$, which certainly makes sense for our purpose. For each $U \in \mathbb{R}_*^{m \times r}$, we write

$$[U] = \{UM : M \in \mathrm{GL}(r)\} = \{U' \in \mathbb{R}_*^{m \times r} : \mathrm{col}(U') = \mathrm{col}(U)\}$$

for the equivalence class of U , and identify it with $\mathrm{col}(U)$, the column space of U . The set of all such equivalence classes is the Grassmann manifold $\mathrm{Gr}(m, r)$: the set of r -dimensional linear subspaces embedded in \mathbb{R}^m —see Section 4.1. Under this description, the Grassmannian is seen as the quotient space $\mathbb{R}_*^{m \times r}/\mathrm{GL}(r)$, which is an alternative to the quotient structure $\mathrm{St}(m, r)/\mathrm{O}(r)$ (4.7) developed in Section 4.1.

Consequently, \hat{f} descends to a well-defined function over the Grassmann manifold. Our task is to minimize this function. Doing so singles out a column space $\mathrm{col}(U)$. We may then pick any basis of that column space, say U , and compute W_U . The product $\hat{X} = UW_U$ (which is invariant w.r.t. the choice of basis U of $\mathrm{col}(U)$) is then our completion of the matrix X . In the next section, we rearrange the terms in \hat{f} to make it easier to compute and give a slightly modified definition of the objective function.

4.2.1 Rearranging the objective function

Considering (4.15), it may seem that evaluating $\hat{f}(U)$ requires the computation of the product UW_U at the entries in Ω and Ω_c , i.e., we would need to compute the whole matrix UW_U , which cannot cost much less than $\mathcal{O}(mn)$. Since applications typically involve very large values of the product mn , this is not acceptable. Fortunately, the regularization term $\|UW_U\|_{\Omega_c}^2$ can be computed cheaply based on the computations that need to be executed for the principal term. Indeed, observe that:

$$\|UW_U\|_\Omega^2 + \|UW_U\|_{\Omega_c}^2 = \|UW_U\|_{\mathrm{F}}^2 = \mathrm{trace}(U^\top UW_U W_U^\top). \quad (4.16)$$

The right-most quantity is computable in $\mathcal{O}((m+n)r^2)$ flops and since $(UW_U)_\Omega$ has to be computed for the first term in the objective function, $\|UW_U\|_{\Omega_c}^2$ turns out to be cheap to obtain. As a result, we see that computing $\hat{f}(U)$ as a whole only requires the computation of $(UW_U)_\Omega$ as opposed to the whole product UW_U , conferring to \hat{f} a computational cost that is linear in the number of observed entries $k = |\Omega|$.

We have the freedom to represent a column space with any of its bases. From a numerical standpoint, it is sound to restrict our attention to orthonormal bases. The set of orthonormal bases U is termed the Stiefel manifold (4.6). Assuming $U \in \mathrm{St}(m, r)$, $U^\top U = I_r$ and equation (4.16)

yield a simple expression for the regularization term:

$$\|UW_U\|_{\Omega_c}^2 = \|W_U\|_F^2 - \|UW_U\|_\Omega^2.$$

Based on this observation, we introduce the following function over $\mathbb{R}_*^{m \times r}$:

$$\begin{aligned} \bar{f}: \mathbb{R}_*^{m \times r} &\rightarrow \mathbb{R}: U \mapsto \bar{f}(U) = \frac{1}{2} \|C \odot (UW_U - X_\Omega)\|_\Omega^2 \\ &\quad + \frac{\lambda^2}{2} (\|W_U\|_F^2 - \|UW_U\|_\Omega^2). \end{aligned} \quad (4.17)$$

In particular, it is the restriction of \bar{f} to $\text{St}(m, r) \subset \mathbb{R}_*^{m \times r}$ that makes sense for our problem:

$$\bar{f}|_{\text{St}}: \text{St}(m, r) \rightarrow \mathbb{R}: U \mapsto \bar{f}|_{\text{St}}(U) = \bar{f}(U).$$

Notice that this restriction coincides with the original cost function: $\hat{f}|_{\text{St}} \equiv \bar{f}|_{\text{St}}$. We then define our objective function f over the Grassmannian:

$$f: \text{Gr}(m, r) \rightarrow \mathbb{R}: \text{col}(U) \mapsto f(\text{col}(U)) = \bar{f}|_{\text{St}}(U), \quad (4.18)$$

where U is any orthonormal basis of the column space $\text{col}(U)$. This is well-defined since $\bar{f}|_{\text{St}}(U) = \bar{f}|_{\text{St}}(UQ)$ for all orthogonal Q . On the other hand, notice that \bar{f} does *not* reduce to a function on the Grassmannian (it does not have the invariance property $\bar{f}(UM) = \bar{f}(U) \forall M \in \text{GL}(r)$), which explains why we had to first go through the Stiefel manifold.

Computing $f(\text{col}(U))$ only requires the computation of UW_U at entries in Ω , at a cost of $\mathcal{O}(kr)$ flops, where $k = |\Omega|$ is the number of known entries. Computing $\|W_U\|_F^2$ costs $\mathcal{O}(nr)$ flops, hence a total evaluation cost of $\mathcal{O}((k+n)r)$ flops, to which we add the (dominating) cost of computing W_U in Section 4.2.3 to obtain the total complexity of evaluating f .

4.2.2 Gradient and Hessian of the objective function

We now obtain the first- and second-order derivatives of f (4.18). As outlined in Section 4.1, $\text{grad } f(\text{col}(U))$ is a tangent vector to the quotient manifold $\text{Gr}(m, r)$. Because of the abstract nature of quotient manifolds, this vector is an abstract object too. In practice, we represent it as a concrete matrix $\text{grad } f(U)$ w.r.t. an (arbitrary) orthonormal basis U of $\text{col}(U)$. Equation (4.10) establishes the link between $\text{grad } f(\text{col}(U))$ and $\text{grad } f(U)$. Following (4.11), we have a convenient expression for $\text{grad } f(U)$:

$$\text{grad } f(U) = (I - UU^\top) \text{grad } \bar{f}(U).$$

We thus first set out to compute $\text{grad } \bar{f}(U)$, which is a classical gradient.

Introduce the function $h: \mathbb{R}_{*}^{m \times r} \times \mathbb{R}^{r \times n} \rightarrow \mathbb{R}$ as follows:

$$h(U, W) = \frac{1}{2} \|C \odot (UW - X_{\Omega})\|_{\Omega}^2 + \frac{\lambda^2}{2} (\|W\|_{\text{F}}^2 - \|UW\|_{\Omega}^2). \quad (4.19)$$

Obviously, h is related to \bar{f} via

$$\bar{f}(U) = \min_W h(U, W) = h(U, W_U).$$

By definition of the classical gradient, $\text{grad } \bar{f}(U) \in \mathbb{R}^{m \times r}$ is the unique vector that satisfies the following condition:

$$\forall H \in \mathbb{R}^{m \times r}, \quad D\bar{f}(U)[H] = \langle H, \text{grad } \bar{f}(U) \rangle,$$

where $D\bar{f}(U)[H]$ is the directional derivative of \bar{f} at U along H and $\langle A, B \rangle = \text{trace}(A^T B)$ is the usual inner product on $\mathbb{R}^{m \times r}$. We thus need to compute the directional derivatives of \bar{f} , which can be done in terms of the directional derivatives of h . Indeed, by the chain rule, it holds that:

$$D\bar{f}(U)[H] = D_1 h(U, W_U)[H] + D_2 h(U, W_U)[W_{U,H}],$$

where D_i indicates differentiation w.r.t. the i^{th} argument and

$$W_{U,H} \triangleq DW(U)[H]$$

is the directional derivative of the mapping $U \mapsto W_U$ at U along H . Since

$$W_U = \underset{W}{\operatorname{argmin}} h(U, W),$$

W_U is a critical point of $h(U, \cdot)$ and it holds that $D_2 h(U, W_U) = 0$. This substantially simplifies the computations as now $D\bar{f}(U)[H] = D_1 h(U, W_U)[H]$: we simply need to differentiate h w.r.t. U , considering W_U as constant. Let us define the mask $\Lambda \in \mathbb{R}^{m \times n}$ as:

$$\Lambda_{ij} = \begin{cases} \lambda & \text{if } (i, j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

Using this notation, we may rewrite h in terms of Frobenius norms only:

$$h(U, W) = \frac{1}{2} \|C \odot (UW - X_{\Omega})\|_{\text{F}}^2 + \frac{\lambda^2}{2} \|W\|_{\text{F}}^2 - \frac{1}{2} \|\Lambda \odot (UW)\|_{\text{F}}^2.$$

This is convenient for differentiation, since for suitably smooth mappings g ,

$$D(X \mapsto 1/2 \|g(X)\|_{\text{F}}^2)(X)[H] = \langle Dg(X)[H], g(X) \rangle.$$

The following holds for all real matrices A, B, C of identical sizes:

$$\langle A \odot B, C \rangle = \langle B, A \odot C \rangle,$$

It thus follows that:

$$\begin{aligned} D\bar{f}(U)[H] &= D_1 h(U, W_U)[H] \\ &= \langle C \odot (HW_U), C \odot (UW_U - X_\Omega) \rangle - \langle \Lambda \odot (HW_U), \Lambda \odot (UW_U) \rangle \\ &= \left\langle H, \left[C^{(2)} \odot (UW_U - X_\Omega) \right] W_U^\top - \left[\Lambda^{(2)} \odot (UW_U) \right] W_U^\top \right\rangle \\ &= \left\langle H, \left[(C^{(2)} - \Lambda^{(2)}) \odot (UW_U - X_\Omega) \right] W_U^\top - \lambda^2 X_\Omega W_U^\top \right\rangle \\ &= \langle H, \text{grad } \bar{f}(U) \rangle. \end{aligned} \quad (4.20)$$

Throughout this chapter, we use the notation $M^{(n)}$ for entry-wise exponentiation, i.e.,

$$(M^{(n)})_{ij} \triangleq (M_{ij})^n.$$

For ease of notation, let us define the following $m \times n$ matrix with the sparsity structure induced by Ω :

$$\hat{C} = C^{(2)} - \Lambda^{(2)}. \quad (4.21)$$

We also introduce the sparse residue matrix R_U :

$$R_U = \hat{C} \odot (UW_U - X_\Omega) - \lambda^2 X_\Omega. \quad (4.22)$$

By identification in (4.20), we obtain a simple expression for the sought gradient:

$$\text{grad } \bar{f}(U) = R_U W_U^\top.$$

We pointed out that $D_2 h(U, W_U) = 0$ because W_U is a critical point of $h(U, \cdot)$. This translates into the following matrix statement:

$$\begin{aligned} \forall H \in \mathbb{R}^{r \times n}, \quad 0 &= D_2 h(U, W_U)[H] \\ &= \langle C \odot (UH), C \odot (UW_U - X_\Omega) \rangle + \lambda^2 \langle H, W_U \rangle \\ &\quad - \langle \Lambda \odot (UH), \Lambda \odot (UW_U) \rangle \\ &= \langle H, U^\top R_U + \lambda^2 W_U \rangle. \end{aligned}$$

Hence,

$$U^\top R_U + \lambda^2 W_U = 0. \quad (4.23)$$

Summing up, we obtain the gradient of f (4.18):

$$\text{grad } f(U) = (I - UU^\top)R_U W_U^\top = R_U W_U^\top + \lambda^2 U(W_U W_U^\top), \quad (4.24)$$

We now differentiate (4.24) according to the identity (4.12) for the Hessian of f . To this end, consider $\bar{F}: \mathbb{R}_*^{m \times r} \rightarrow \mathbb{R}^{m \times r}$:

$$\bar{F}(U) = R_U W_U^\top + \lambda^2 U(W_U W_U^\top).$$

According to (4.12), the Hessian of f is given by:

$$\text{Hess } f(U)[H] = (I - UU^\top)\text{D}\bar{F}(U)[H]. \quad (4.25)$$

Let us compute the differential of \bar{F} :

$$\begin{aligned} \text{D}\bar{F}(U)[H] &= [\hat{C} \odot (HW_U + UW_{U,H})]W_U^\top + R_U W_{U,H}^\top \\ &\quad + \lambda^2 H(W_U W_U^\top) + \lambda^2 U(W_{U,H} W_U^\top + W_U W_{U,H}^\top). \end{aligned}$$

Applying the projector $I - UU^\top$ to $\text{D}\bar{F}(U)[H]$ cancels out all terms of the form UM (since $(I - UU^\top)U = 0$) and leave all terms of the form HM unaffected (since $U^\top H = 0$). As a consequence of (4.23), applying the projector to $R_U W_{U,H}^\top$ yields:

$$(I - UU^\top)R_U W_{U,H}^\top = R_U W_{U,H}^\top + \lambda^2 UW_U W_{U,H}^\top.$$

Applying these observations to (4.25), we obtain an expression for the Hessian of our objective function on the Grassmann manifold:

$$\begin{aligned} \text{Hess } f(U)[H] &= (I - UU^\top) \left[\hat{C} \odot (HW_U + UW_{U,H}) \right] W_U^\top \\ &\quad + R_U W_{U,H}^\top + \lambda^2 H(W_U W_U^\top) + \lambda^2 U(W_U W_{U,H}^\top). \end{aligned} \quad (4.26)$$

Not surprisingly, the formula for the Hessian requires the computation of $W_{U,H}$, the differential of the mapping $U \mapsto W_U$ along H . The next section provides formulas for W_U and $W_{U,H}$.

4.2.3 W_U and its derivative $W_{U,H}$

We still need to provide explicit formulas for W_U and $W_{U,H}$. We assume $U \in \text{St}(m, r)$ since we use orthonormal matrices to represent points on the Grassmannian and $U^\top H = 0$ since $H \in T_U \text{Gr}(m, r)$.

The vectorization operator, vec , transforms matrices into vectors by stacking their columns—in Matlab notation, $\text{vec}(A) = A(:)$. Denoting the

Kronecker product of two matrices by \otimes , the following well-known identity holds, for matrices A, Y, B of appropriate sizes (Brookes, 2005):

$$\text{vec}(AYB) = (B^\top \otimes A)\text{vec}(Y).$$

We also write I_Ω for the orthonormal $k \times mn$ matrix such that

$$\text{vec}_\Omega(M) = I_\Omega \text{vec}(M)$$

is a vector of length $k = |\Omega|$ corresponding to the entries M_{ij} for $(i, j) \in \Omega$, taken in order from $\text{vec}(M)$.

Computing W_U comes down to minimizing the least-squares cost $h(U, W)$ (4.19) with respect to W . We manipulate h to reach a standard form for least-squares. To this end, first define $S \in \mathbb{R}^{k \times mn}$:

$$S = I_\Omega \text{diag}(\text{vec}(C)).$$

This will come in handy through the identity

$$\begin{aligned} \|C \odot M\|_\Omega^2 &= \|\text{vec}_\Omega(C \odot M)\|_2^2 = \|I_\Omega \text{vec}(C \odot M)\|_2^2 \\ &= \|I_\Omega \text{diag}(\text{vec}(C)) \text{vec}(M)\|_2^2 = \|S \text{vec}(M)\|_2^2. \end{aligned}$$

We use this in the following transformation of h :

$$\begin{aligned} h(U, W) &= \frac{1}{2} \|C \odot (UW - X_\Omega)\|_\Omega^2 + \frac{\lambda^2}{2} \|W\|_{\text{F}}^2 - \frac{\lambda^2}{2} \|UW\|_\Omega^2 \\ &= \frac{1}{2} \|S \text{vec}(UW) - \text{vec}_\Omega(C \odot X_\Omega)\|_2^2 + \frac{\lambda^2}{2} \|\text{vec}(W)\|_2^2 \\ &\quad - \frac{\lambda^2}{2} \|\text{vec}_\Omega(UW)\|_2^2 \\ &= \frac{1}{2} \|S(I_n \otimes U) \text{vec}(W) - \text{vec}_\Omega(C \odot X_\Omega)\|_2^2 + \frac{1}{2} \|\lambda I_{rn} \text{vec}(W)\|_2^2 \\ &\quad - \frac{1}{2} \|\lambda I_\Omega(I_n \otimes U) \text{vec}(W)\|_2^2 \\ &= \frac{1}{2} \left\| \begin{bmatrix} S(I_n \otimes U) \\ \lambda I_{rn} \end{bmatrix} \text{vec}(W) - \begin{bmatrix} \text{vec}_\Omega(C \odot X_\Omega) \\ 0_{rn} \end{bmatrix} \right\|_2^2 \\ &\quad - \frac{1}{2} \left\| [\lambda I_\Omega(I_n \otimes U)] \text{vec}(W) \right\|_2^2 \\ &= \frac{1}{2} \|A_1 w - b_1\|_2^2 - \frac{1}{2} \|A_2 w\|_2^2 \\ &= \frac{1}{2} w^\top (A_1^\top A_1 - A_2^\top A_2) w - b_1^\top A_1 w + \frac{1}{2} b_1^\top b_1, \end{aligned}$$

with $w = \text{vec}(W) \in \mathbb{R}^{rn}$, $0_{rn} \in \mathbb{R}^{rn}$ is the zero-vector and the obvious definitions for A_1, A_2 and b_1 . If $A_1^\top A_1 - A_2^\top A_2$ is positive-definite there is a unique minimizing vector $\text{vec}(W_U)$, given by:

$$\text{vec}(W_U) = (A_1^\top A_1 - A_2^\top A_2)^{-1} A_1^\top b_1.$$

It is easy to compute the following:

$$\begin{aligned} A_1^\top A_1 &= (I_n \otimes U^\top)(S^\top S)(I_n \otimes U) + \lambda^2 I_{rn}, \\ A_2^\top A_2 &= (I_n \otimes U^\top)(\lambda^2 I_\Omega^\top I_\Omega)(I_n \otimes U), \\ A_1^\top b_1 &= (I_n \otimes U^\top)S^\top \text{vec}_\Omega(C \odot X_\Omega) = (I_n \otimes U^\top)\text{vec}(C^{(2)} \odot X_\Omega). \end{aligned}$$

Note that $S^\top S - \lambda^2 I_\Omega^\top I_\Omega = \text{diag}(\text{vec}(\hat{C}))$. Let us call this matrix B :

$$B \triangleq S^\top S - \lambda^2 I_\Omega^\top I_\Omega = \text{diag}(\text{vec}(\hat{C})).$$

Then define $A \in \mathbb{R}^{rn \times rn}$ as:

$$A \triangleq A_1^\top A_1 - A_2^\top A_2 = (I_n \otimes U^\top)B(I_n \otimes U) + \lambda^2 I_{rn}. \quad (4.27)$$

Observe that the matrix A is block-diagonal, with n symmetric blocks of size r . This structure stems from the fact that each column of W_U can be computed separately from the others. Each block is indeed positive-definite provided $\lambda > 0$ (making A positive-definite too). Thanks to the sparsity of \hat{C} , we can compute these n blocks with $\mathcal{O}(kr^2)$ flops. To solve systems in A , we compute the Cholesky factorization of each block, at a total cost of $\mathcal{O}(nr^3)$ flops. Once these factorizations are computed, each system only costs $\mathcal{O}(nr^2)$ flops to solve (Trefethen & Bau, 1997). Collecting all equations in this section, we obtain a closed-form formula for W_U :

$$\begin{aligned} \text{vec}(W_U) &= A^{-1}(I_n \otimes U^\top)\text{vec}(C^{(2)} \odot X_\Omega) \\ &= A^{-1}\text{vec}\left(U^\top[C^{(2)} \odot X_\Omega]\right), \end{aligned} \quad (4.28)$$

where A is a function of U and we have a fast way of applying A^{-1} to vectors.

We would like to differentiate W_U with respect to U , that is, compute

$$\begin{aligned} \text{vec}(W_{U,H}) &= D(U \mapsto \text{vec}(W_U))(U)[H] \\ &= D(U \mapsto A^{-1})(U)[H] \cdot \text{vec}\left(U^\top[C^{(2)} \odot X_\Omega]\right) \\ &\quad + A^{-1}\text{vec}\left(H^\top[C^{(2)} \odot X_\Omega]\right). \end{aligned} \quad (4.29)$$

Using the formula $D(Y \mapsto Y^{-1})(X)[H] = -X^{-1}HX^{-1}$ (Brookes, 2005) for the differential of the inverse of a matrix, we obtain

$$\begin{aligned} D(U \mapsto A^{-1})(U)[H] &= -A^{-1} \cdot D(U \mapsto A)(U)[H] \cdot A^{-1} \\ &= -A^{-1} \left[(I_n \otimes H^\top)B(I_n \otimes U) + (I_n \otimes U^\top)B(I_n \otimes H) \right] A^{-1}. \end{aligned}$$

Plug this back in (4.29), recalling (4.28) for W_U :

$$\begin{aligned} \text{vec}(W_{U,H}) &= -A^{-1} \left[(I_n \otimes H^\top)B(I_n \otimes U) + (I_n \otimes U^\top)B(I_n \otimes H) \right] \text{vec}(W_U) \\ &\quad + A^{-1} \text{vec}(H^\top[C^{(2)} \odot X_\Omega]) \\ &= -A^{-1} \left[(I_n \otimes H^\top)B \text{vec}(UW_U) + (I_n \otimes U^\top)B \text{vec}(HW_U) \right] \\ &\quad + A^{-1} \text{vec}(H^\top[C^{(2)} \odot X_\Omega]) \\ &= -A^{-1} \left[(I_n \otimes H^\top) \text{vec}(\hat{C} \odot UW_U) + (I_n \otimes U^\top) \text{vec}(\hat{C} \odot HW_U) \right] \\ &\quad + A^{-1} \text{vec}(H^\top[C^{(2)} \odot X_\Omega]) \\ &= -A^{-1} \left[\text{vec}(H^\top[\hat{C} \odot UW_U]) + \text{vec}(U^\top[\hat{C} \odot HW_U]) \right] \\ &\quad + A^{-1} \text{vec}(H^\top[C^{(2)} \odot X_\Omega]). \end{aligned} \tag{4.30}$$

Now recall the definition of R_U (4.22) and observe that

$$\hat{C} \odot UW_U - C^{(2)} \odot X_\Omega = \hat{C} \odot UW_U - \hat{C} \odot X_\Omega - \Lambda^{(2)} \odot X_\Omega = R_U.$$

Plugging the latter in (4.30) yields a compact expression for the directional derivative $W_{U,H}$:

$$\text{vec}(W_{U,H}) = -A^{-1} \text{vec}(H^\top R_U + U^\top(\hat{C} \odot (HW_U))). \tag{4.31}$$

The most expensive operation involved in computing $W_{U,H}$ ought to be solving a linear system in A . Fortunately, we already factored the n small diagonal blocks of A in Cholesky form to compute W_U . Consequently, after computing W_U , computing $W_{U,H}$ is cheap. The computational complexities are summarized in Section 4.2.5.

4.2.4 Preconditioning the Hessian

As Section 4.4 demonstrates with the numerical experiment (scenario 3), the Hessian of the cost function $\text{Hess } f$ (4.26) can be badly conditioned when the target matrix X is itself badly conditioned. Such conditioning issues slow

down optimization algorithms, and it is known that good preconditioners can have a dramatic effect on performance in such cases (Conn *et al.*, 2000; Hager & Zhang, 2006). In this section, we consider a simplified version of the low-rank matrix completion problem, which allows us to simplify the expression of the Hessian. This, in turn, yields an approximation of the inverse of the Hessian. That operator is then used to precondition the Hessian in the optimization algorithms discussed in Section 4.3. Figure 4.4 demonstrates the effectiveness of this preconditioner.

In order to (drastically) simplify the problem at hand, assume all entries of X are observed, with identical confidence $C_{ij} = c$. Hence, $\hat{C} = (c^2 - \lambda^2)\mathbb{1}_{m \times n}$ where $\mathbb{1}$ is the all-ones matrix. Then, remembering that $U^\top H = 0$, we get successively for R_U (4.22) and the Hessian (4.26):

$$\begin{aligned} R_U &= c^2(UW_U - X_\Omega) - \lambda^2 UW_U, \text{ and} \\ \text{Hess } f(U)[H] &= c^2 H(W_U W_U^\top) + (R_U + \lambda^2 UW_U) W_{U,H}^\top \\ &= c^2 H(W_U W_U^\top) + c^2(UW_U - X_\Omega) W_{U,H}^\top. \end{aligned}$$

Now consider $W_{U,H}$. Recalling equations (4.31) for $W_{U,H}$ and (4.27) for A , still under the same assumption on \hat{C} , it follows that:

$$\begin{aligned} \text{Avec}(W_{U,H}) &= -\text{vec}(H^\top R_U) \\ &= \text{vec}(\lambda^2 W_{U,H} + (c^2 - \lambda^2) W_{U,H}) = c^2 \text{vec}(W_{U,H}). \end{aligned}$$

Hence,

$$\begin{aligned} W_{U,H} &= -\frac{1}{c^2} H^\top R_U = H^\top X_\Omega, \text{ and} \\ \text{Hess } f(U)[H] &= c^2 H(W_U W_U^\top) + c^2(UW_U - X_\Omega) X_\Omega^\top H. \end{aligned}$$

Further assuming we are close to convergence and the observations are not too noisy, that is, $X_\Omega \approx UW_U$, then $UW_U - X_\Omega \approx 0$ and $X_\Omega^\top H \approx 0$. The second term is thus small and the Hessian may be approximated by the mapping $H \mapsto c^2 H(W_U W_U^\top)$. Notice that this mapping is linear and symmetric, from and to the tangent space at U . This observation prompts the following formula for a preconditioner:

$$\text{Precon } f(U)[H] = \frac{1}{c^2} H(W_U W_U^\top)^{-1}. \quad (4.32)$$

The small $r \times r$ matrix $W_U W_U^\top$ is already computed when the gradient at U is computed. Applying the preconditioner further requires solving linear systems in that matrix. The cost of this is $\mathcal{O}(r^3)$ to prepare a Cholesky factorization of $W_U W_U^\top$ (once per iteration) and an additional $\mathcal{O}(mr^2)$ per

application. Notice that this cheap cost is independent of the number of observed entries k . In practice, c can be chosen to be the average value of the positive C_{ij} 's.

It is important that a preconditioner be symmetric and positive definite on the tangent space at U . The proposed preconditioner indeed fulfills these requirements provided W_U is full-rank. The factor W_U is expected to be full-rank near convergence provided the lowest-rank matrix compatible with the observation X_Ω is of rank at least r . In practice, we could monitor the condition number of $W_U W_U^\top$ at each iteration, and decrease r if it becomes too large (indicating we overshot the true rank of the sought matrix X).

Preconditioning the Hessian with (4.32) is tightly related to the approaches favored by Mishra *et al.* (2012b) and by Ngo & Saad (2012). In the latter reference, the authors pose low-rank matrix completion as an optimization problem on two Grassmannians (one for the row space and one for the column space). If (U, V) is a couple of orthonormal matrices representing the row and column spaces of the current estimate USV^\top , then the metric on the first Grassmannian is scaled by $SS^\top = (SV^\top)(SV^\top)^\top$ and likewise the metric on the second Grassmannian is scaled by $S^\top S = (US)^\top(US)$ (notice the cross-talk between U and V , akin to our preconditioning iterations on U using the W factor). Mishra et al. (Mishra *et al.*, 2012b) represent low-rank matrices on a quotient space of factorizations of the form GH^\top . The metric on G is scaled by $H^\top H$ and likewise the metric on H is scaled by $G^\top G$. The effect of changing the metric is similar to the effect of preconditioning, in that it makes the cost function “look more isotropic”.

Underlying the preconditioner (4.32) is the approximation of the Hessian at U as

$$(\text{Precon } f(U))^{-1}[H] = c^2 H(W_U W_U^\top).$$

This operator induces a norm on the tangent space at U , which we call the M -norm (following notation of (Conn *et al.*, 2000, Alg. 7.5.1)):

$$\|H\|_M^2 = \langle H, (\text{Precon } f(U))^{-1}[H] \rangle_U = c^2 \langle HW_U, HW_U \rangle. \quad (4.33)$$

This norm appears in the description of the preconditioned Riemannian trust-region method (Section 3.2) but needs never be computed explicitly.

4.2.5 Numerical complexities

By exploiting the sparsity of many of the matrices involved and the special structure of the matrix A (4.27) appearing in the computation of W_U and $W_{U,H}$, it is possible to compute the objective function f (4.18) as well as its gradient (4.24) and its Hessian (4.26) on the Grassmannian in time linear in the size of the data $k = |\Omega|$. Memory complexities are also linear in k . We summarize the computational complexities in Table 4.1. Note that most

computations are easily parallelizable, but we do not take advantage of it here.

The computational cost of W_U (4.28) is dominated by the computation of the n diagonal blocks of A of size $r \times r$ — $\mathcal{O}(kr^2)$ —and by the Cholesky factorization of these— $\mathcal{O}(nr^3)$ —hence a total cost of $\mathcal{O}(kr^2 + nr^3)$ flops. The computation of $f(U)$ is dominated by the cost of computing W_U , hence they have the same complexity. Computing the gradient of f once W_U is known involves just a few supplementary matrix-matrix multiplications. Exploiting the sparsity of these matrices keeps the cost low: $\mathcal{O}(kr + (m + n)r^2)$ flops. Computing the Hessian of f along H requires (on top of W_U) the computation of $W_{U,H}$ and a few (structured) matrix-matrix products. Computing $W_{U,H}$ involves solving a linear system in A . Since we computed W_U already, we have a Cholesky-factored representation of A , hence solving a system in A is cheap: $\mathcal{O}(nr^2)$ flops. The total cost of computing $W_{U,H}$ and $\text{Hess } f(U)[H]$ is $\mathcal{O}(kr + (m + n)r^2)$ flops.

Notice that computing the gradient and the Hessian is *cheaper* than computing f . This stems from the fact that once we have computed f at a certain point U , much of the work (such as computing and factoring the diagonal blocks of A) can be reused to compute higher-order information. This prompts us to investigate methods that exploit second-order information.

Computation	Complexity	By-products	Formulas
W_U and $f(U)$	$\mathcal{O}(kr^2 + nr^3)$	Cholesky of A	(4.17)–(4.18), (4.27), (4.28)
$\text{grad } f(U)$	$\mathcal{O}(kr + (m + n)r^2)$	$R_U, W_U W_U^\top$	(4.21), (4.22), (4.24)
$\text{Hess } f(U)[H]$	$\mathcal{O}(kr + (m + n)r^2)$	$W_{U,H}$	(4.26), (4.31)
Precon $f(U)[H]$	$\mathcal{O}(mr^2)$		(4.32)

Table 4.1: All complexities are at most linear in $k = |\Omega|$, the number of observed entries.

4.3 Riemannian optimization setup

To minimize the cost function f on the Grassmann manifold, we choose Riemannian optimization algorithms which can be preconditioned: the Riemannian trust-region method (RTR) and the Riemannian conjugate gradient method (RCG). Both of these methods are described in Chapter 3 and are implemented in Manopt. RTR can make full use of the Hessian information, while RCG does not use the Hessian at all. The algorithms are

implemented while making full use of Manopt’s built-in caching capabilities, which help prevent redundant computations.

4.3.1 Initial guess

Both the RTR and the RCG methods require an initial guess for the column space, $\text{col}(U_0)$. To this end, compute the r dominant left singular vectors of the masked matrix X_Ω . In Matlab, this is achieved by calling $[U_0, S_0, V_0] = \text{svds}(X_\Omega, r)$. Since X_Ω is sparse, this has a reasonable cost (linear in k). The initial model is thus $U_0 W_{U_0}$. Alternative methods to compute an initial guess, with analysis, as well as to guess the rank r if it is unknown can be found in (Keshavan & Oh, 2009) and, more recently, in (Chatterjee, 2012).

4.3.2 Preconditioned Riemannian trust-regions

We use RTR (Section 3.2) with and without preconditioner, and call the resulting methods RTRMC 2p and RTRMC 2, respectively. RTR accepts a few parameters. Among them, the maximum and initial trust-region radii are denoted respectively $\bar{\Delta}$ and Δ_0 .

The trust-region radius at a given iterate is the upper bound on the M -norm (4.33) of the steps one is willing to take—see eq. (3.5). To keep things proportioned when talking about both the preconditioned and the unpreconditioned variants, let

$$s^2 = \lambda_{\max}(c^2 W_{U_0} W_{U_0}^\top)$$

when the preconditioner is used, and let $s = 1$ otherwise. That is: s is the 2-norm of the approximation of the square root of the Hessian underlying the preconditioner. Since the Grassmann manifold is compact, it makes sense to choose $\bar{\Delta}$ in proportion to the diameter of this manifold, i.e., the largest geodesic distance between any two points on $\text{Gr}(m, r)$. The distance between two subspaces is $\sqrt{\theta_1^2 + \dots + \theta_r^2}$, where the θ_i ’s are the principal angles between these spaces. Since these angles are bounded by $\pi/2$, we set $\bar{\Delta} = s\pi\sqrt{r}/2$. Accordingly, we set the initial trust-region radius as $\Delta_0 = \bar{\Delta}/8$.

The number of inner iterations for the tCG algorithm (inner solve) is limited to 500. While this limit may jeopardize the quadratic convergence rate of RTR, we find that it is seldom (if ever) reached for reasonably well-conditioned problems, and otherwise prevents excessive running times.

All other parameters are set to their default values.

As a means to investigate the role of second-order information in this algorithm, we also experiment with RTRMC 1, which is the same method but the Hessian is “approximated” by the identity matrix. The analysis of the RTR method still guarantees global convergence for this setup.

4.3.3 Preconditioned Riemannian conjugate gradients

The RCG method from Section 3.1 is applied as is to the matrix completion problem, with and without preconditioning. The resulting methods are referred to as RCGMCp and RCGMC respectively.

4.4 Numerical experiments

The proposed algorithms are tested on synthetic data and compared against ADMiRA (Lee & Bresler, 2010), OptSpace (Keshavan & Oh, 2009), SVT (Cai *et al.*, 2010), Balanced Factorization (Meyer *et al.*, 2011a), GROUSE (Balzano *et al.*, 2010), LMaFit (Wen *et al.*, 2012), LRGeom (Vandereycken, 2013), qGeomMC-CG (Mishra *et al.*, 2012b) and ScGrass-CG (Ngo & Saad, 2012) in terms of accuracy and computation time. We observed that the first four mentioned algorithms are never competitive with the best algorithms, so that we omit them in the discussion. Jellyfish (Recht *et al.*, 2011) and the divide-and-conquer approach of Mackey *et al.* (2011) explicitly target parallel architectures. The previously mentioned algorithms could be parallelized but this was not the focus of the authors, nor is it ours. We thus did not compare with parallel implementations. We could not compare against NNLS (Toh & Yun, 2010) since the code provided by the authors often crashes (as also observed by the authors of Jellyfish), but we do compare with GROUSE (Balzano *et al.*, 2010), whose authors point out it outperforms NNLS. We significantly enhanced the implementation of GROUSE by implementing the rank-one updates it performs using a C-Mex file. This file calls the BLAS level 2 routine `dger` directly. GROUSE's performance is sensitive to its step-size parameter. After experiments on a wide range of values, we decided to set it to 0.3 or 0.5, whichever performs best.

All algorithms are run sequentially by Matlab on the same computer.¹ This computer has 12 cores. Even though none of the tested codes are explicitly multithreaded, some of them get some mileage out of the multicore architecture owing to Matlab's built-in parallelization of some tasks. All Matlab implementations call subroutines in C-Mex code to efficiently deal with the sparsity of the matrices involved. All of these C-Mex codes are single-threaded even though they are typically embarrassingly parallelizable. Profiling indicates a lot of computation time could be saved there, but this is beyond our scope. See (Recht *et al.*, 2011) for emphasis on parallel computing for LRMC.

The proposed methods (RTRMC 2 and 2p, RCGMC and RCGMCp) as well as the competing methods GROUSE, LRGeom, qGeomMC-CG and

¹HP DL180 + Intel Xeon X5670 2.93 GHz (12 core), 144Go RAM, Matlab 7.10 (R2010a), Linux (64 bits)

ScGrass-CG require knowledge of the target rank r . LMaFit includes a mechanism to guess the rank, but benefits from knowing it, hence we provide the target rank to LMaFit too.

Remark 4.1 (About guessing the rank). *If one over-estimates the rank, the factorization UW_U results in a rank deficient factor W_U . This is detectable by monitoring the condition number of the $r \times r$ matrix $W_U W_U^\top$. If this number becomes too large, r could be reduced. If one under-estimates the rank, the algorithm is expected to converge toward a lower-rank approximation of the target matrix, which typically is the desired behavior—see scenario 7 below. Guessing strategies for the rank (some of them rather refined and sophisticated) have been proposed that can be used with any fixed-rank matrix completion algorithm (Chatterjee, 2012; Keshavan & Oh, 2009; Wen et al., 2012). It has also been shown that starting from a rank 1 approximation and iteratively increasing the rank until no significant improvement is detected can be beneficial, see for example the so-called homotopy strategy in (Vandereycken, 2013). All algorithms tested here could be adapted to work iteratively with increasing rank.*

We use the root mean square error (RMSE) criterion to assess the quality of reconstruction of X with \hat{X} :

$$\text{RMSE}(X, \hat{X}) = \|X - \hat{X}\|_{\text{F}} / \sqrt{mn}.$$

This quantity is cheap to compute when the target matrix is given in factored low-rank form $X = AB$ and \hat{X} is (by construction) in the same form $\hat{X} = UW$. The RMSE may then accurately be computed in $\mathcal{O}((m+n)r^2)$ flops observing that $AB - UW = [A \ U] [B^\top \ -W^\top]^\top$. Computing thin (rank $2r$) QR decompositions of both terms as $Q_1 R_1 = [A \ U]$ and $Q_2 R_2 = [B^\top \ -W^\top]$ yields the following formula: $\|AB - UW\|_{\text{F}} = \|Q_1 R_1 R_2^\top Q_2^\top\|_{\text{F}} = \|R_1 R_2^\top\|_{\text{F}}$. This is much more accurate than the algorithm we used previously in (Boumal & Absil, 2011c). For the purpose of comparing the algorithms, we add code to all implementations so that the RMSE is computed at each iterate. The time spent in this calculation is discounted from the reported timings.

A number of factors intervene in the difficulty of a low-rank matrix completion task. Obviously, the size $m \times n$ of the matrix X to recover and its rank r are fundamental quantities. Among others, the presence or absence of noise is important. If the observations X_Ω are noisy, then X_Ω is not the masked version of a low-rank matrix X , but of a matrix which is close to being low-rank: $X + \text{noise}$. Of course, different noise distributions (with and without outliers etc.) can be investigated. The search space—the manifold of $m \times n$ matrices of rank r —has dimension $d = r(m+n-r)$. The oversampling ratio k/d is a crucial quantity: the larger it is, the easier the

task is. The *sampling process* also plays a role in the difficulty of matrix completion. Under uniform sampling for example, all entries of the matrix X have identical probability of being observed. Uniform sampling prevents pathological cases (where some rows or columns have no observed entry at all for example) from happening with high probability. Real datasets often have nonuniform samplings. For example, some movies are particularly popular and some users rate particularly many movies. Finally, the *conditioning* of the low-rank matrix X (the ratio of its largest to its smallest nonzero singular values) may affect the difficulty of matrix completion too.

The following numerical experiments explore these various pitfalls. For the noiseless scenarios, in our methods, we let $\lambda = 0$ (no regularization). All observed entries are trusted with the same confidence $C_{ij} = 1$. For the preconditioner, let $c = 1$ (the average confidence). In practice, because it is numerically convenient, we scale the whole cost function by $1/k$.

Scenario 1: low oversampling ratio We first compare the convergence behavior of the different methods with square matrices $m = n = 10\,000$ and rank $r = 10$. We generate $A \in \mathbb{R}^{m \times r}$ and $B \in \mathbb{R}^{r \times n}$ with i.i.d. normal entries of zero mean and unit variance. The target matrix is $X = AB$. We sample $3d$ entries uniformly at random without noise, which yields a sampling ratio of 0.6%. This is fairly low. Figure 4.1 is typical and shows the evolution of the RMSE as a function of time. Most modern methods are efficient on such a standard task.

Scenario 2: rectangular matrices In this second test, we repeat the previous experiment with rectangular matrices: $m = 1\,000, n = 30\,000, r = 5$ and a sampling ratio of 2.6% ($5d$ known entries), see Figure 4.2. We expect and confirm that RTRMC, RCGMC and GROUSE perform well on rectangular matrices since they optimize over the smallest of either the column space or the row space, and not both at the same time.

Scenario 3: bad conditioning For this third test, we generate A and B as in Scenario 1 with $m = n = 1\,000, r = 10$. We then compute the thin SVD of the product $AB = USV^\top$, which can be done efficiently using economic QR factorizations of A and B separately. The diagonal $r \times r$ matrix S is replaced with a diagonal matrix S_+ whose diagonal entries decay exponentially as follows: $(S_+)_{ii} = \sqrt{mn} \exp(-5(i-1)/(r-1))$, for $i = 1 \dots r$. The product $X = US_+V^\top$ is then formed and this is the rank- r target matrix, of which we observe $5d$ entries uniformly at random (that is about 10%). Notice that X has much worse conditioning ($e^5 \approx 148$) than the original product AB (typical conditioning below 2) without being unrealistically bad. From the numerical results in Figure 4.3, it appears that

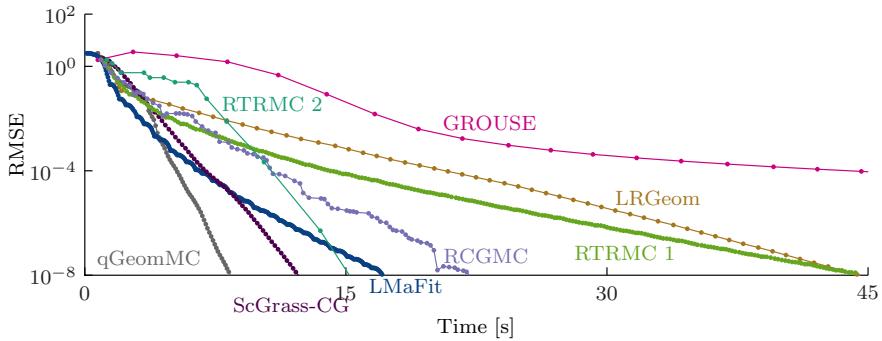


Figure 4.1: Scenario 1: standard completion task on a square $10\,000 \times 10\,000$ matrix of rank 10, with a low oversampling ratio of 3, that is, 99.4% of the entries are unknown. Most methods perform well. As the test is repeated, the ranking of the top-performing algorithms varies a little. Since the problem is well-conditioned, the preconditioned variants of our algorithms perform almost the same as the standard algorithms. They are omitted for legibility.

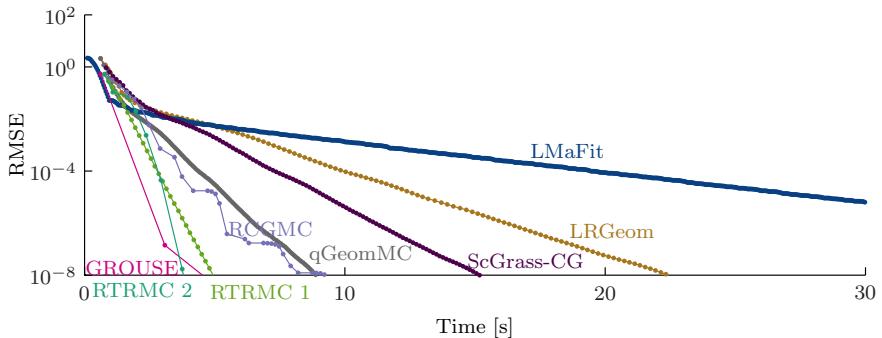


Figure 4.2: Scenario 2: completion task on a rectangular matrix of size $1\,000 \times 30\,000$ of rank 5, with an oversampling ratio of 5. For rectangular matrices, RTRMC and GROUSE are especially efficient since they optimize over a single Grassmann manifold. As a consequence, the dimension of their nonlinear search space grows linearly in $\min(m, n)$, whereas for most methods the growth is linear in $m + n$. Preconditioned variants perform about the same and are omitted for legibility.

most methods have difficulties solving this task, while our preconditioned algorithms RTRMC 2p and RCGMCp quickly solve them to high accuracy. RTRMC 2 also succeeds, at the cost of more Hessian evaluations than before.

We venture an explanation of the better performance of the proposed second-order and preconditioned methods here by studying the condition number of the Hessian of the cost function f at the solution $\text{col}(U)$ —see Figure 4.4. This Hessian is a symmetric linear operator on the linear subspace $T_U \text{Gr}(m, r)$ of dimension $r(m-r) = 9900$. For the present experiment (target US_+V^\top), we compute the 9900 associated eigenvalues with Matlab’s `eigs`. They are all positive. The condition number of the Hessian is 72 120. The fact that the bad conditioning of X translates into even worse conditioning of the Hessian at the solution can be explained by the approximate expression for the Hessian, $c^2 W_U W_U^\top$ (Section 4.2.4). Then, at the solution U ,

$$\text{cond}(\text{Hess } f(U)) \approx \text{cond}(W_U W_U^\top) = \text{cond}^2(X). \quad (4.34)$$

As seen from Figure 4.4, the preconditioner nicely reduces the condition number of the Hessian to 7.3 by an appropriate change of variable.

Scenario 4: nonuniform sampling As a fourth test, we generate A and B as in Scenario 1 with $m = 1\,000, n = 10\,000$ and $r = 10$. The target matrix is $X = AB$, of which we observe entries with a *nonuniform* distribution. The chosen artificial sampling mimics a situation where rows correspond to movies and columns correspond to raters. Some of the movies are much more often rated than others, and some of the raters rate many more movies than others. Each of the 100 first movies (they are the least popular ones) has a probability of being rated that is 5 times smaller than the 800 following movies. The 100 last movies are 5 times as likely to be rated as the 800 latter (they are the popular ones). Furthermore, each rater rates between 15 and 50 movies, uniformly at random, resulting in an oversampling ratio of 2.94 (3.2%). Figure 4.5 shows the associated mask probability, where raters (columns) have been sorted by number of given ratings.

Figure 4.6 shows the behavior of the various methods tested on this instance of the problem. It appears that the proposed algorithms can cope with some non-uniformity in the sampling procedure.

Scenario 5: larger instances In this fifth test, we try out the various algorithms on a larger instance of matrix completion: $m = 10\,000, n = 100\,000, r = 20$ with oversampling ratio of 5, that is, 1.1% of the entries are observed, sampled uniformly at random. The target matrix $X = AB$

(formed as previously) has a billion entries. Figure 4.7 shows that RTRMC 2(p) performs well on such instances.

Scenario 6: noisy observations As a sixth test, we try out RTRMC on a class of noisy instances of matrix completion with $m = n = 5\,000$, $r = 10$ and oversampling ratio of 4, that is, 1.6% of the entries are observed, sampled uniformly at random. The target matrix $X = AB$ is formed as before with $A \in \mathbb{R}^{m \times r}$, $B \in \mathbb{R}^{r \times n}$ whose entries are i.i.d. normal random variables. Notice that this implies the X_{ij} 's are also zero-mean Gaussian variables but with variance r and not independent. We then generate a noise matrix N_Ω , such that the $(N_\Omega)_{ij}$'s for (i, j) in Ω are i.i.d. normal random variables (Gaussian distribution with zero mean, unit variance), and the other entries of N are zero. The observed matrix is $X_\Omega + \sigma N_\Omega$, where σ^2 is the noise variance. The signal to noise ratio (SNR) is thus r/σ^2 . This is the same setup as the standard scenario in (Keshavan *et al.*, 2009).

All algorithms based on a least-squares strategy should perform rather well on this scenario, since least-squares are particularly well-suited to filter out Gaussian noise. And indeed, as they perform almost the same, we only show results for RTRMC. We should however expect those same algorithms to perform suboptimally in the face of outliers. RTRMC makes no claim of being robust against outliers, hence we only test against Gaussian noise and show excellent behavior in that case on Figure 4.8. For comparison, we use the same oracle as in (Keshavan *et al.*, 2009), that is: we compare the RMSE obtained by RTRMC with the RMSE we could obtain if we knew the column space $\text{col}(X)$. This is known to be equal to $\text{RMSE}_{\text{oracle}} = \sigma\sqrt{(2nr - r^2)/k}$ (in expectation). Figure 4.8 illustrates the fact that, not surprisingly, RTRMC reaches almost the same RMSE as the oracle as soon as the SNR is large enough.

Scenario 7: underestimating the rank As a seventh test, the target matrix X has true rank 32, with positive singular values decaying exponentially from \sqrt{mn} to $\sqrt{mn} \cdot 10^{-10}$ (see Scenario 3). We challenge the algorithms to reconstruct a matrix of rank 8 which best approximates X . Such a scenario is motivated in (Vandereycken, 2013), in the context of approximating almost separable functions of two variables.

The size of X is given by $m = n = 5\,000$ and 10d entries are observed uniformly at random, that is, 3.2%. An oracle knowing the matrix X perfectly would simply return the SVD of X truncated to rank 8, committing an RMSE of $3 \cdot 10^{-3}$. In repeated realizations of this test, the only methods we observed converging close to the oracle bound are preconditioned methods, see Figure 4.9.

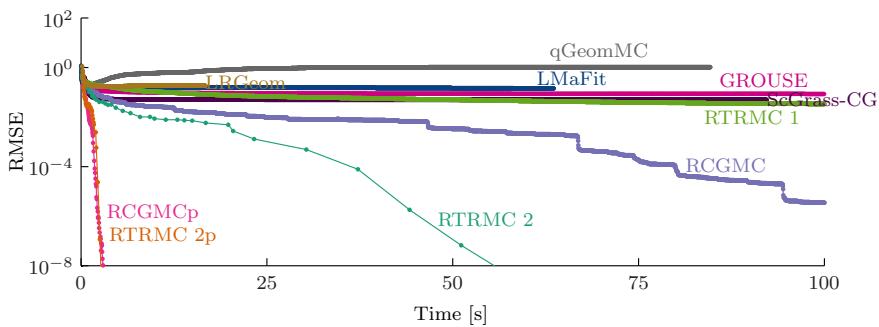


Figure 4.3: Scenario 3: completion task on a square $1\,000 \times 1\,000$ matrix of rank 10 with an oversampling factor of 5 and a condition number of about 150. RTRMC 2, using more Hessian applications than on better conditioned problems, shows good convergence quality. Our preconditioned algorithms RCGMCp and RTRMC 2p perform even better. Surprisingly, ScGrassCG and qGeomMC, which both are (in a slightly different way) preconditioned too, sometimes solve the problem too, but less often than not and require more time.

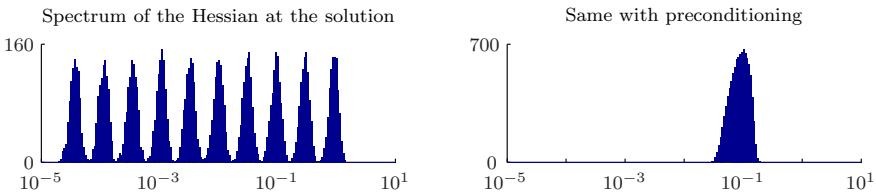


Figure 4.4: Spectrum (in \log_{10}) of the Hessian and of the preconditioned Hessian at the solution of scenario 3. The target matrix has a condition number of about 150. This translates into a challenging Hessian condition number of more than 72 000. Preconditioning the Hessian controls this condition number back to 7.3, explaining the success of RTRMC 2p and RCGMCp on scenario 3. Notice that the spectrum of the Hessian has $r = 10$ modes.



Figure 4.5: Proposed nonuniform sampling density for Scenario 4. This image represents a 1000×10000 matrix. Each entry is colored on a grayscale. The lighter the color, the slimmer the chances that this entry is observed. We see that entries in the top 100 rows are much less likely to be observed than in the bottom 100 rows. Columns on the right are also more densely sampled than columns on the left. This artificial sampling process mimics a situation where some objects are more popular than others (and hence more often rated) and some raters are more active than others.

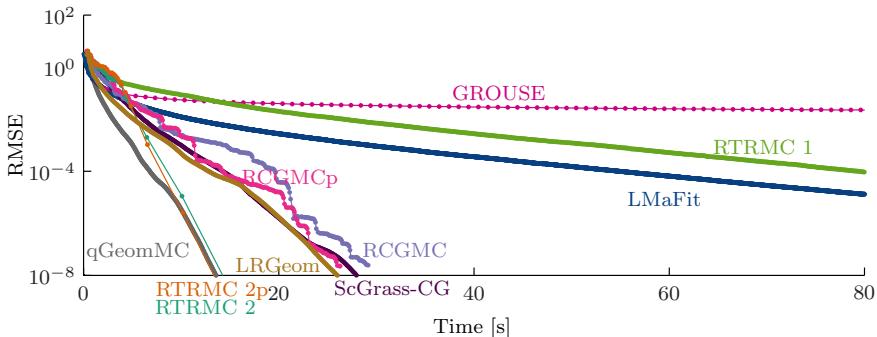


Figure 4.6: Scenario 4: completion task on a rectangular 1000×10000 matrix or rank 10, with 3.2% (OS = 2.9) of the entries revealed following a nonuniform sampling as depicted in Figure 4.5. It appears most methods can withstand some non-uniformity GROUSE is slowed down by the non-uniformity, possibly because it operates one column at a time.

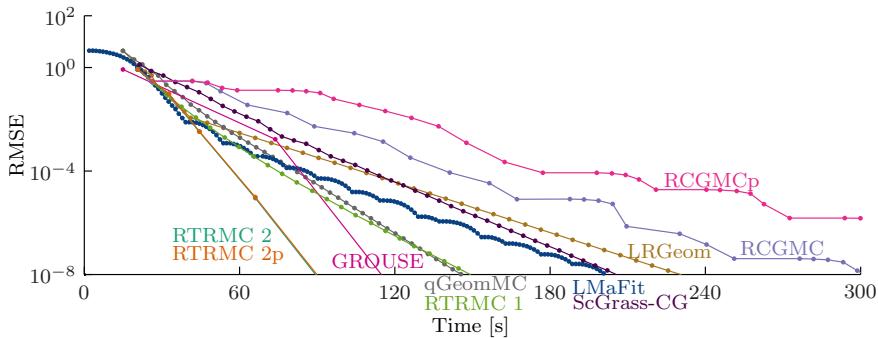


Figure 4.7: Scenario 5: completion task on a larger $10\,000 \times 100\,000$ matrix of rank 20 with an oversampling ratio of 5.

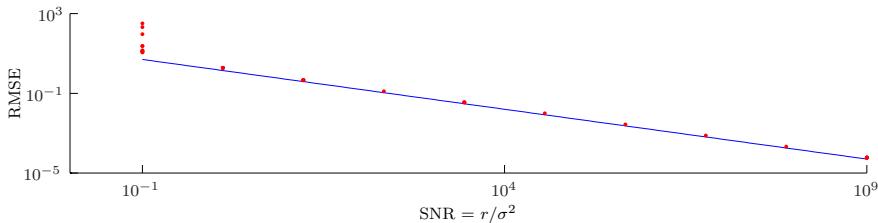


Figure 4.8: Scenario 6: RTRMC is well suited to solve matrix completion tasks under Gaussian noise, owing to its least-squares objective function ($m = n = 5\,000, r = 10, |\Omega|/(mn) = 1.6\%$). The straight blue line indicates the RMSE that an oracle who knows the column space of the target matrix X would reach (this is a lower bound on the performance of any practical algorithm). For different values of SNR, we generate 10 problem instances and solve them with RTRMC. The red dots report the RMSE's reached by RTRMC. For SNR's larger than 1, the dots are mostly indistinguishable and close to the oracle quality, which shows that Gaussian noise is easily filtered out.

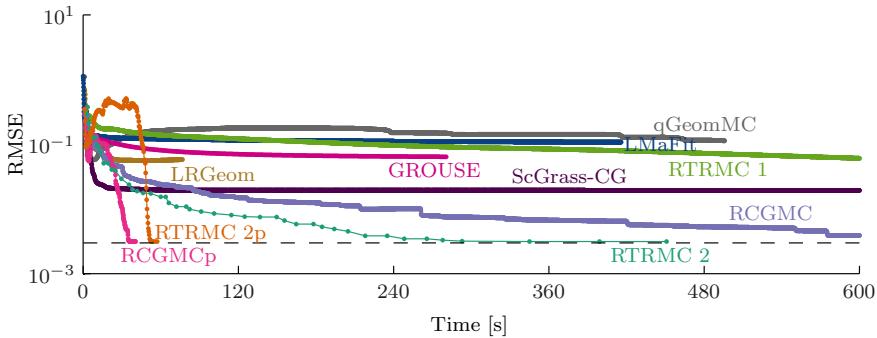


Figure 4.9: Scenario 7: in this challenging completion task, the target matrix is square, $5\,000 \times 5\,000$, and has rank 32 with singular values decaying exponentially from \sqrt{mn} to $\sqrt{mn} \cdot 10^{-10}$. The various algorithms attempt to construct a matrix of rank 8 best approximating the ill-conditioned target matrix, based on 3.2% of revealed entries ($OS = 10$). The best methods almost reach the oracle RMSE (dashed line), which corresponds to the RMSE reached by an SVD of the true matrix X truncated to rank 8. The outcome of this test is less constant than the others. Over many runs, the typical result is that RCGMCp and RTRMC 2p almost always converge as depicted, while qGeomMC and ScGrass-CG (the only two other preconditioned methods in this test) sometimes converge as well, but less often than not. RTRMC 2 and RCGMC sometimes achieve good reconstructions too (as depicted) but are always much slower than their preconditioned counterparts. Over many runs, we did not witness the other methods reach values close to the oracle bound.

4.5 Application: the Netflix prize

The competing algorithms are now tested on the Netflix data.² The complete matrix is $17\,770 \times 480\,189$ with 100 198 805 entries revealed for training and 281 702 additional entries reserved for validation. Each row corresponds to a movie and each column corresponds to a user. Entries are integer ratings between 1 and 5. The data is preprocessed to remove users who rated less than 10 movies and movies which were rated less than 100 times—more on this momentarily. The resulting matrix is $16\,777 \times 462\,746$ with $k = 100\,028\,462$ training entries (1.29%) and $k_{\text{test}} = 276\,279$ test entries. The training data is centered around the mean of the training entries (3.6047), since the regularized cost function (4.18) implicitly puts a prior on the value of each unknown entry being 0.

Figure 4.10 shows the behavior of the various algorithms we propose, with test RMSE as a function of time. The test RMSE is computed over the test entries only, not over the training entries. The rank is set to $r = 10$ (corresponding to an oversampling factor of 20.86). The regularization parameter λ of our algorithms is set to 0.1. The preconditioned algorithms seem to perform the best on this real dataset. We use RCGMCp to produce Table 4.2, where various values for the regularization λ as well as the reconstruction rank r are tested.

We tried our best to run the other algorithms on this same dataset, without success. As shown by Table 4.2, regularization is of prime importance on this dataset, which may explain why non-regularized competing algorithms fail (their test RMSE increases with iterations, well above 1). For ScGrass-CG, which supports regularization, we tested setting the regularization parameter to $\frac{k}{mn}10^{-4}$, as suggested by the authors (without guarantees on their part), as well as 0.01 (which essentially corresponds to the same regularization as what we use for our algorithms). For GROUSE, we tested multiple values of the step-size reduction speed. For LRGeom, we tried starting it with our initial guess $U_0W_{U_0}$, without any more success. We also tried all algorithms on the full dataset, without preprocessing, and all algorithms (including ours) fail. This explains why the rows and columns with too few known entries are removed in this test.

A baseline RMSE to compare with is the one reached by an algorithm which simply returns the average training rating. This RMSE is 1.128. The RMSE reached by RCGMCp in 12 minutes for rank 10, $\lambda = 0.1$, is 0.953. This corresponds to the RMSE reached by Cinematch, Netflix’s own algorithm, at the onset of the competition (Koren, 2009). Better scores can be reached, even using only low-rank matrix completion. But more importantly, from the literature on the Netflix competition, it is known

²<http://hazy.cs.wisc.edu/hazy/victor/download/>

that plain low-rank approximation is not sufficient to reach the best known scores, although it can provide an important basis for better predictors. For example, temporal information (how recent a rating is) should be taken into account. Perhaps most fundamentally, the least-squares criterion at the root of RTRMC and RCGMC, which necessarily leads to poor outlier rejection, is to blame for their humble performance. Nevertheless, the positive impact of preconditioning on this real dataset is interesting to note.

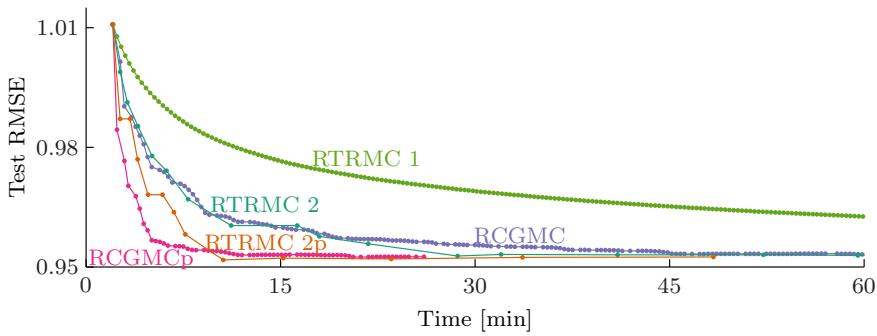


Figure 4.10: Convergence of our algorithms on a large fraction of the Netflix dataset: $16\,777 \times 462\,746$ with $k = 100 \cdot 10^6$ known ratings. The test RMSE is evaluated on 276 279 test ratings (not used for training). The algorithms aim for a rank 10 fitting of the data, with regularization parameter $\lambda = 0.1$. The preconditioned algorithms perform the best on this real dataset.

rank $r = 10$						
regul. λ	0.001	0.01	0.1	0.2	1	
RMSE (initial)	2.108	1.160	1.011	1.017	1.086	
RMSE	1.104	0.970	0.953	0.985	1.086	
Time [min]	75	58	12	7	2	

regularization $\lambda = 0.1$						
rank r	1	2	5	10	15	20
RMSE (initial)	1.086	1.047	1.023	1.011	1.009	1.014
RMSE	1.067	1.006	0.967	0.953	0.951	0.953
Time [min]	3	3	5	12	24	48

Table 4.2: Test RMSE of the initial guess, then best test RMSE reached by RCGMCp on a large fraction of the Netflix dataset, with various values of the regularization parameter λ and of the reconstruction rank r . Reported timings include the computation of the initial guess and of the RCGMCp iterations. A first conclusion is that regularization is necessary on this dataset. Another conclusion, looking at the increase in RMSE going from rank 15 to 20, is that aiming at large rank reconstruction “from scratch” may not be efficient. This suggests looking at incremental rank procedures such as the ones described in (Vandereycken, 2013). We did try incrementing the rank gradually as follows: obtain an SVD-based initial guess for rank $r = 1$ (similarly for $r = 10$), apply RCGMCp with $\lambda = 0.1$, then complement the obtained orthonormal basis U with a uniformly random column orthogonal to the column space of U . This is now a rank $r + 1$ basis which can be used as initial guess for RCGMCp. We iterate up to rank 20. The observation (not depicted) is that even though the cost value does steadily decrease, the RMSE’s reached after convergence stagnate close to a best value of 0.951. Hence, it is not clear that incremental rank procedures could boost the performance of the proposed methods in this setting.

4.6 Conclusions

Our contribution is a regularized cost function for low-rank matrix completion on a single Grassmann manifold, along with a set of efficient numerical methods to minimize it: RTRMC 2(p) and RCGMC(p). These are respectively second-order Riemannian trust-region methods and Riemannian conjugate gradient methods, with or without preconditioning. These algorithms compete with the state-of-the-art. The trust-region methods further enjoy global and local convergence to critical points, with a quadratic local convergence rate for RTRMC 2(p).

The methods we propose are particularly efficient on rectangular matrices. We believe this is because the dimension of the nonlinear search space, $\text{Gr}(m, r)$, grows as $\min(m, n)$, whereas for most competing methods the growth is in $m + n$. We also observed that second-order and preconditioned methods perform better than first-order methods when the matrix to complete is badly conditioned. We believe this is because the bad conditioning of the target matrix translates into an even worse conditioning of the Hessian of the cost function at the solution, as shown in the numerical experiments (4.34). Furthermore, the proposed algorithms can withstand low oversampling ratios or non-uniformity in the sampling process. RTRMC is effective against Gaussian noise, which is not surprising given its least-squares nature. Those combined strengths make RTRMC and RCGMC reasonably efficient on the Netflix dataset too.

A major drawback of the proposed algorithms is their strong reliance on the explicit solve of the inner least-squares problem in (4.5). This precludes simple adaptations of these algorithms to reach for better outlier rejection. Least-squares are indeed well-suited against Gaussian noise but perform poorly against wildly erroneous measurements. This explains at least in part the modest RMSE's reached on the Netflix dataset. Competing methods such as LRGeom, qGeomMC or Jellyfish may more easily accommodate better suited loss functions.

As future improvement, all the proposed methods could be parallelized to compete with very large scale implementations such as Jellyfish (Recht *et al.*, 2011) or the divide and conquer scheme of Mackey *et al.* (2011). Indeed, the expensive operations involved in computing the cost and its derivatives are inherently parallelizable over the columns.

Matlab code for RTRMC and RCGMC is available at:

<http://sites.uclouvain.be/absil/RTRMC/>.

Chapter 5

Synchronization of rotations

Synchronization of rotations is the problem of estimating rotation matrices $R_1, \dots, R_N \in \text{SO}(n)$ from noisy measurements of relative rotations $R_i R_j^\top$, where $\text{SO}(n)$ is the special orthogonal group:

$$\text{SO}(n) = \{R \in \mathbb{R}^{n \times n} : R^\top R = I_n, \det(R) = +1\}. \quad (5.1)$$

The set of available measurements gives rise to a graph structure, where the N nodes correspond to the rotations $\{R_i\}_{i=1\dots N}$ and an edge is present between two nodes i and j if a measurement of $R_i R_j^\top$ is given. Depending on the application, some rotations may be known in advance or not. The known rotations, if any, are called *anchors*. In the absence of anchors, it is only possible to recover the rotations up to a global rotation, since the measurements only reveal relative information.

Synchronization of rotations appears naturally in a number of important applications. Tron & Vidal (2009) for example consider a network of cameras. Each camera has a certain position in \mathbb{R}^3 and orientation in $\text{SO}(3)$. For some pairs of cameras, a calibration procedure produces a noisy measurement of relative position and relative orientation. The task of using all relative orientation measurements simultaneously to estimate the configuration of the individual cameras is a synchronization problem. An example of practical setup for this problem is the calibration of the Panoptic camera system: a golf ball-sized dome on which cameras are mounted, pointing outward, to acquire a representation of all of the surroundings simultaneously (Afshari *et al.*, 2013). See also the structure from motion problem (Arie-Nachimson *et al.*, 2012) and the global registration problem (Chaudhury *et al.*, 2013; Krishnan *et al.*, 2007). Cucuringu *et al.* (2012b) address sensor network

localization based on inter-node distance measurements. In their approach, they decompose the network in small, overlapping, rigid patches. Each patch is easily embedded in space owing to its rigidity, but the individual embeddings are noisy. These embeddings are then aggregated by aligning overlapping patches. For each pair of such patches, a measurement of relative orientation is produced. Synchronization permits the use all of these measurements simultaneously to prevent error propagation. In related work, Cucuringu *et al.* (2012a) apply a similar approach to the molecule problem. Tzveneva *et al.* (2011) and Wang & Singer (2013) apply synchronization to the construction of 3D models of objects based on scans of the objects under various unknown orientations (See Section 5.6). Singer & Shkolnisky (2011) study cryo-EM imaging. In this problem, the aim is to produce a 3D model of a macro-molecule based on many projections (pictures) of the molecule under various random and unknown orientations. A procedure specific to the cryo-EM imaging technique helps estimating the relative orientation between pairs of projections, but this process is very noisy. In fact, most measurements are outliers. The task is to use these noisy measurements of relative orientations of images to recover the true orientations under which the images were acquired. This naturally falls into the scope of synchronization of rotations, and calls for very robust algorithms. More recently, Sonday *et al.* (2013) use synchronization as a means to compute rotationally invariant distances between snapshots of trajectories of dynamical systems, as an important preprocessing stage before dimensionality reduction. In a different setting, Yu (Yu, 2009, 2012) applies synchronization of in-plane rotations (under the name of angular embedding) as a means to rank objects based on pairwise relative ranking measurements. This approach is in contrast with existing techniques which realize the embedding on the real line, but appears to provide unprecedented robustness. Hartley *et al.* (2013) address a broad class of rotation averaging problems, with a specific outlook for characterizations of the existence and uniqueness of global optimizers of the related optimization problems. Synchronization is addressed too under the name of multiple rotation averaging.

In some of these applications, a large subset of the measurements may be of poor quality. These applications call for robust synchronization algorithms, capable of withstanding outliers. Hartley *et al.* (2011) propose to estimate the rotations by minimizing an L1 norm of the disagreement between the model and the measurements, the Weiszfeld algorithm. The resulting algorithm is simple, fast and is shown to produce good results, but comes with little theoretical guarantees because of the nonconvexity and the nonsmoothness of the optimization problem they solve. Wang & Singer (2013) propose LUD, a convex relaxation of the synchronization problem we describe in Section 5.5.2. LUD achieves exact recovery when a given

portion of the measurements are exact, the other measurements being uniformly random. When the former measurements are slightly noisy rather than perfect, LUD remains stable.

Both the Weiszfeld algorithm and LUD address the synchronization problem by proposing a certain cost function at the onset. In contrast, we will address synchronization by first assuming a specific noise model on the measurements. This statistical approach to the problem has a number of advantages: (i) the underlying assumptions about the noise are clear and could be adapted to individual applications; (ii) Cramér-Rao bounds can be derived that provide a meaningful target to compare algorithms against, as we do in Chapter 8; and (iii) the definition of maximum likelihood estimator (MLE) naturally suggests an estimation algorithm.

In this chapter, we first propose a noise model for synchronization of rotations and define the associated MLE (Section 5.1). The MLE is the solution of an optimization problem on a manifold whose geometry is described in Section 5.2 for the anchored case. The latter optimization problem is nonconvex and we will need a good initial guess to (hope to) solve it. To that effect, Section 5.3 presents the eigenvector method, a spectral relaxation of the synchronization problem, together with an analysis due to Singer (2011) and slightly adapted for our purpose. Then we describe an algorithm to try to compute the MLE (Section 5.4) and we study, in Section 5.5, its performance against existing algorithms and the Cramér-Rao bounds (CRB) which we derive later, in Chapter 8. The conclusion will be that, in many scenarios, the computed proxy for the MLE seems to reach the CRB's. This, in turn, gives credit to the interpretations of the CRB.

5.1 Robust synchronization of rotations

In synchronization, the target quantities (the parameters) are the rotation matrices

$$R_1, \dots, R_N \in \mathrm{SO}(n).$$

In order to estimate these rotations, we are given measurements H_{ij} . Each H_{ij} is a noisy measurement of the relative rotation $R_i R_j^\top$. The available measurements define an undirected *measurement graph* or *synchronization graph*

$$\mathcal{G} = (V, \mathcal{E}), \tag{5.2}$$

with vertex set $V = \{1, \dots, N\}$ and edge set \mathcal{E} , where $\{i, j\} \in \mathcal{E}$ if a measurement H_{ij} is available. By symmetry, $H_{ij} = H_{ji}^\top$. The measurements

are modeled as follows:

$$H_{ij} = Z_{ij} R_i R_j^\top, \quad (5.3)$$

where $Z_{ij} \in \text{SO}(n)$ is a random variable. In order to model a measurement that is concentrated around the true relative rotation, one can give Z_{ij} a probability density function (pdf) that is concentrated around the identity matrix I_n .

A popular Gaussian-like distribution on $\text{SO}(n)$ is the *Langevin distribution*, which has the following pdf:

$$\ell_\kappa: \text{SO}(n) \rightarrow \mathbb{R}^+, \quad \ell_\kappa(Z) = \frac{1}{c_n(\kappa)} \exp(\kappa \text{trace}(Z)), \quad (5.4)$$

where $\kappa \geq 0$ is the *concentration parameter*. The pdf $\ell_\kappa(Z)$ attains its maximum at $Z = I_n$. The larger κ is, the more ℓ_κ is concentrated around the identity. As an extreme case, ℓ_0 is constant over $\text{SO}(n)$, i.e., it corresponds to a uniform distribution. If Z_{ij} is uniformly distributed, then so is H_{ij} and the measurement contains no information. On the other hand, ℓ_∞ is the point-mass function at the identity. If Z_{ij} is deterministically equal to I_n , then H_{ij} is a noiseless measurement. For $0 < \kappa < \infty$, the measurement H_{ij} is isotropically distributed around its mean $R_i R_j^\top$. The normalization constant $c_n(\kappa)$ ensures that ℓ_κ has unit integral over $\text{SO}(n)$ with respect to the Haar measure—see Section 8.3.

In order to model the fact that only a fraction $0 \leq p \leq 1$ of the measurements are of decent quality while the remaining measurements contain little or no information, we propose to consider the following pdf for the noise rotations Z_{ij} :

$$f: \text{SO}(n) \rightarrow \mathbb{R}^+, \quad f(Z) = p \ell_\kappa(Z) + (1 - p) \ell_{\kappa'}(Z). \quad (5.5)$$

This mixture of Langevin's indeed captures the presence of a fraction $1 - p$ of outliers if we let κ' be small compared to κ . With probability p , a measurement H_{ij} is distributed around $R_i R_j^\top$ with high concentration κ , and with probability $1 - p$, it is distributed around the same mean with low concentration κ' . In the sequel, we will often consider $\kappa' = 0$.

We make the following assumption on the noise rotations:

Assumption 5.1. *The Z_{ij} 's pertaining to different measurements are independent, identically distributed, with probability density function f (5.5).*

The assumption that the Z_{ij} 's are identically distributed is merely to allow for a cleaner exposition. All of what follows goes through if one assumes specific values of p, κ and κ' for each measurement individually (indeed, the Matlab code used in Section 5.5 allows for such freedom). Independence, on the other hand, is a central assumption in the present work and cannot be relaxed easily. For convenience we further assume connectivity:

Assumption 5.2. *The measurement graph (5.2) is connected.*

If the graph is not connected, all of what follows may be applied to each connected component separately. With a little more care, it is not even necessary to work on separate components.

Under Assumption 5.1, the *log-likelihood function* L for synchronization of rotations is as follows:

$$L: \text{SO}(n)^N \rightarrow \mathbb{R}, \quad L(\hat{\mathbf{R}}) = \sum_{i \sim j} \log f(H_{ij} \hat{R}_j \hat{R}_i^\top). \quad (5.6)$$

The summation is over the edges of the measurement graph.

Depending on the application, some of the rotations may be known in advance. They are called *anchors*. If no anchor is provided, synchronization can only be performed up to a global rotation. It is then acceptable, for the purpose of obtaining an estimator, to fix an arbitrary rotation to, say, the identity matrix. Let $A \subset \{1, \dots, N\}$ denote the set of indices of anchors. The parameter space, that is, the space of acceptable values for an estimator, is

$$\mathcal{P}_A = \{\hat{\mathbf{R}} \in \text{SO}(n)^N : \forall i \in A, \hat{R}_i = R_i\}. \quad (5.7)$$

The *maximum likelihood estimator* (MLE) $\hat{\mathbf{R}}_{\text{MLE}}$ is the parameter that maximizes the log-likelihood function L :

$$\hat{\mathbf{R}}_{\text{MLE}} = \underset{\mathbf{R} \in \mathcal{P}_A}{\operatorname{argmax}} L(\hat{\mathbf{R}}). \quad (5.8)$$

That it, $\hat{\mathbf{R}}_{\text{MLE}}$ is the assignment of rotations R_1, \dots, R_N that best explains the observations H_{ij} under the assumed noise model and in the absence of prior information on the rotations. Since L is a smooth function defined over the smooth and compact manifold \mathcal{P}_A , a global maximizer exists.

Remark 5.1 (least-squares case). *In particular, if we assume that there are no outliers, then we may set $p = 1$, set κ to some appropriate value, and κ' is irrelevant. The pdf of the Z_{ij} 's reduces to a simple Langevin prior: $f(Z) = \ell_\kappa(Z)$. The log-likelihood function then reads:*

$$L(\hat{\mathbf{R}}) = \sum_{i \sim j} \kappa \operatorname{trace}(H_{ij} \hat{R}_j \hat{R}_i^\top) + \text{constant}. \quad (5.9)$$

Since

$$\begin{aligned} \|H_{ij} \hat{R}_j - \hat{R}_i\|_{\text{F}}^2 &= \|H_{ij} \hat{R}_j\|_{\text{F}}^2 + \|\hat{R}_i\|_{\text{F}}^2 - 2 \operatorname{trace}(H_{ij} \hat{R}_j \hat{R}_i^\top) \\ &= 2(n - \operatorname{trace}(H_{ij} \hat{R}_j \hat{R}_i^\top)), \end{aligned}$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm, maximizing L is equivalent to minimizing

$$\sum_{i \sim j} \kappa \|H_{ij}\hat{R}_j - \hat{R}_i\|_{\text{F}}^2.$$

Hence, synchronization algorithms based on the minimization of the above least-squares criterion over \mathcal{P}_A are maximum likelihood estimators under a Langevin prior.

5.2 Geometry of the parameter space, with anchors

The MLE is defined as the global optimizer of an optimization problem over the Riemannian manifold \mathcal{P}_A (5.7). In order to apply the Riemannian optimization tools detailed in Chapter 3 to this situation, we need to describe the geometry of \mathcal{P}_A , which is the focus of this section. We start with a quick reminder of the geometry of $\text{SO}(n)$. This exposition relies on the differential geometric definitions from Chapter 2.

The group of rotations $\text{SO}(n)$ (5.1) is a connected, compact Lie group of dimension $d = n(n - 1)/2$. Being a Lie group, it is also a manifold and thus admits a tangent space $T_Q \text{SO}(n)$ at each point Q . The tangent space at the identity plays a special role. It is known as the *Lie algebra* of $\text{SO}(n)$ and is the set of skew-symmetric matrices:

$$T_I \text{SO}(n) = \mathfrak{so}(n) \triangleq \{\Omega \in \mathbb{R}^{n \times n} : \Omega + \Omega^\top = 0\}.$$

The other tangent spaces are easily obtained from $\mathfrak{so}(n)$:

$$T_Q \text{SO}(n) = Q \mathfrak{so}(n) = \{Q\Omega : \Omega \in \mathfrak{so}(n)\}.$$

Indeed, differentiating the constraint $Q^\top Q = I_n$ yields the condition $Q^\top \dot{Q} + \dot{Q}^\top Q = 0$ for a vector \dot{Q} to be a tangent vector at Q . We endow $\text{SO}(n)$ with the usual Riemannian metric by defining the following inner product on all tangent spaces:

$$\langle Q\Omega_1, Q\Omega_2 \rangle_Q = \text{trace}(\Omega_1^\top \Omega_2), \quad \|Q\Omega\|_Q^2 = \langle Q\Omega, Q\Omega \rangle_Q = \|\Omega\|_{\text{F}}^2.$$

Thus, $\text{SO}(n)$ is a Riemannian submanifold of $\mathbb{R}^{n \times n}$ with its standard metric. For better readability, we often omit the subscripts Q . The orthogonal projector from the embedding space $\mathbb{R}^{n \times n}$ onto the tangent space $T_Q \text{SO}(n)$ is:

$$\text{Proj}_Q(H) = Q \text{skew}(Q^\top H), \text{ with } \text{skew}(A) \triangleq (A - A^\top)/2.$$

It plays an important role in the computation of gradients of functions on $\mathrm{SO}(n)$, which will come up in optimization algorithms. The exponential map and the logarithmic map with respect to the Riemannian structure (Section 2.6) accept simple expressions in terms of matrix exponential and logarithm:

$$\begin{aligned}\mathrm{Exp}_Q : \mathrm{T}_Q \mathrm{SO}(n) &\rightarrow \mathrm{SO}(n), & \mathrm{Log}_Q : \mathrm{SO}(n) &\rightarrow \mathrm{T}_Q \mathrm{SO}(n) \\ \mathrm{Exp}_Q(Q\Omega) &= Q \exp(\Omega), & \mathrm{Log}_{Q_1}(Q_2) &= Q_1 \log(Q_1^\top Q_2).\end{aligned}\quad (5.10)$$

The mapping $t \mapsto \mathrm{Exp}_Q(tQ\Omega)$ defines a geodesic curve on $\mathrm{SO}(n)$, passing through Q with velocity $Q\Omega$ at time $t = 0$. Geodesic curves have zero acceleration and may be considered as the equivalent of straight lines on manifolds (Section 2.5). The logarithmic map Log_Q is (locally) the inverse of the exponential map Exp_Q . In the context of an estimation problem, $\mathrm{Log}_Q(\hat{Q})$ represents the estimation error of \hat{Q} for the parameter Q , that is, it is a notion of difference between Q and \hat{Q} . This will be useful in Chapter 8. The geodesic (or Riemannian) distance on $\mathrm{SO}(n)$ is the length of the shortest path (the geodesic arc) joining two points:

$$\mathrm{dist}(Q_1, Q_2) = \|\mathrm{Log}_{Q_1}(Q_2)\|_{Q_1} = \|\log(Q_1^\top Q_2)\|_{\mathrm{F}}.\quad (5.11)$$

In particular, for rotations in the plane ($n = 2$) and in space ($n = 3$), the geodesic distance between Q_1 and Q_2 is $\sqrt{2}\theta$, where $\theta \in [0, \pi]$ is the angle by which $Q_1^\top Q_2$ rotates.

Let $\tilde{f} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}$ be a differentiable function, and let $f = \tilde{f}|_{\mathrm{SO}(n)}$ be its restriction to $\mathrm{SO}(n)$. The gradient of f is a tangent vector field to $\mathrm{SO}(n)$ uniquely defined by:

$$\langle \mathrm{grad}f(Q), Q\Omega \rangle = \mathrm{D}f(Q)[Q\Omega] \quad \forall \Omega \in \mathfrak{so}(n),$$

with $\mathrm{grad}f(Q) \in \mathrm{T}_Q \mathrm{SO}(n)$ and $\mathrm{D}f(Q)[Q\Omega]$ the directional derivative of f at Q along $Q\Omega$. Let $\nabla \tilde{f}(Q)$ be the usual gradient of \tilde{f} in $\mathbb{R}^{n \times n}$. Then, the Riemannian gradient of f is easily computed as the orthogonal projection of $\nabla \tilde{f}(Q)$ on the tangent space at Q (Section 2.3). In the sequel, we often write ∇f to denote the gradient of f seen as a function in $\mathbb{R}^{n \times n}$, even if it is defined on $\mathrm{SO}(n)$, and express the Riemannian gradient simply as

$$\mathrm{grad}f(Q) = Q \operatorname{skew}(Q^\top \nabla f(Q)).$$

Similarly, from Section 2.4, an expression for the Riemannian Hessian of f at Q along $Q\Omega$ follows, in terms of the classical Hessian of f seen as a function in $\mathbb{R}^{n \times n}$ which we write $\nabla^2 f(Q)[Q\Omega]$:

$$\mathrm{Hess} f(Q)[Q\Omega] = Q \operatorname{skew}(Q^\top \nabla^2 f(Q)[Q\Omega] - \Omega \operatorname{sym}(Q^\top \nabla f(Q))).$$

Above, $\text{sym}(A) = (A + A^\top)/2$ extracts the symmetric part of a matrix. The Hessian comes up in second-order optimization algorithms.

The parent parameter space for synchronization is the product Lie group $\mathcal{P} = \text{SO}(n)^N$. Its geometry is trivially obtained by element-wise extension of the geometry of $\text{SO}(n)$ just described. In particular, tangent spaces and the Riemannian metric are given by:

$$\begin{aligned} T_{\mathbf{R}}\mathcal{P} &= \{\mathbf{R}\Omega = (R_1\Omega_1, \dots, R_N\Omega_N) : \Omega_1, \dots, \Omega_N \in \mathfrak{so}(n)\}, \\ \langle \mathbf{R}\Omega, \mathbf{R}\Omega' \rangle_{\mathbf{R}} &= \sum_{i=1}^N \text{trace}(\Omega_i^\top \Omega'_i). \end{aligned} \quad (5.12)$$

In the presence of anchors indexed in $A \subset \{1, \dots, N\}$, the parameter space is \mathcal{P}_A (5.7), a Riemannian submanifold of \mathcal{P} . The tangent space at $\hat{\mathbf{R}} \in \mathcal{P}_A$ is given by:

$$T_{\hat{\mathbf{R}}}\mathcal{P}_A = \{\mathbf{R}\Omega \in T_{\hat{\mathbf{R}}}\mathcal{P} : \forall i \in A, \Omega_i = 0\},$$

such that the orthogonal projector $\text{Proj}_{\hat{\mathbf{R}}} : T_{\hat{\mathbf{R}}}\mathcal{P} \rightarrow T_{\hat{\mathbf{R}}}\mathcal{P}_A$ simply sets to zero all components of a tangent vector that correspond to anchored rotations. All tools on \mathcal{P}_A (exponential and logarithmic map for example) are inherited in the obvious fashion from \mathcal{P} . In particular, the geodesic distance on \mathcal{P}_A is:

$$\text{dist}^2(\hat{\mathbf{R}}, \hat{\mathbf{R}}') = \sum_{i \notin A} \|\log(\hat{R}_i^\top \hat{R}'_i)\|_{\text{F}}^2. \quad (5.13)$$

5.3 The eigenvector method and its phase transition point

The synchronization problem, although not convex, admits surprisingly efficient tractable relaxations in the form of semidefinite programs (SDP's) or even, as we now discuss, in the form of a simple spectral problem. We refer to the solution based on the spectral relaxation as the *eigenvector method*. These relaxations were first addressed by Singer (2011) then further studied in a number of directions (Bandeira *et al.*, 2013b; Tzveneva *et al.*, 2011). This section is concerned with the simplest version of the eigenvector method, namely for a complete measurement graph and identical weights for all measurements. This is more amenable to analysis. In Section 5.4.1, a more general version of the eigenvector method is described, for practical use as an initial guess.

In his original paper, Singer (2011) focuses on in-plane rotations ($n = 2$) and studies a simple but revealing noise model where measurements are

either perfect (with probability p) or uniformly distributed (with probability $1 - p$). This allows to investigate the outlier-resilience of the eigenvector method. The analysis relies heavily on tools from random matrix theory, as we outline momentarily. Tzeneva *et al.* (2011) extends that analysis to the case of rotations in \mathbb{R}^3 , with the same noise model, and further considers the case of an incomplete measurement graph following an Erdős-Rényi model. In this section, we reproduce much of their analysis for context, and spell out as a minor contribution the application of this analysis to the large class of noise models satisfying Assumptions 8.1–8.3 from Chapter 8 for rotations in $\text{SO}(n)$, arbitrary n . It is good to keep this analysis in mind to compare with what the Cramér-Rao bounds from that same chapter teach us.

The rotations to estimate are denoted $R_1, \dots, R_N \in \text{SO}(n)$. Consider the slightly modified noise model

$$H_{ij} = R_i Z_{ij} R_j^\top,$$

where Z_{ij} is a random rotation. As per an argument similar to Remark 8.1, this noise model is equivalent to the standard model (5.3). It turns out to be more practical to work with for the present analysis.¹ The symmetry $H_{ij} = H_{ji}^\top$ implies $Z_{ij} = Z_{ji}^\top$. For simplicity, assume the noise matrices Z_{ij} are i.i.d. and such that $\mathbb{E}\{Z_{ij}\} = \beta I_n$, for some $0 < \beta \leq 1$ —we show below that this is always the case for the unbiased, isotropic noise models covered by Assumptions 8.1–8.3. Further assume that all measurements are acquired (the measurement graph is complete) and build the block matrices $H, Z \in \mathbb{R}^{nN \times nN}$ such that the off-diagonal blocks are given by the H_{ij} 's and the Z_{ij} 's respectively, and the diagonal blocks are defined as $H_{ii} = Z_{ii} = \beta I_n$. This definition of the diagonal blocks is a technical necessity for the analysis. In practice, β is unknown and the diagonal blocks are set to the identity. This small perturbation shifts all eigenvalues of H and Z by $1 - \beta$, which is negligible compared to their top eigenvalues which grow with N (see below). Now let $R \in \mathbb{R}^{nN \times n}$ be a tall block matrix with blocks R_i and let $D_R \in \mathbb{R}^{nN \times nN}$ be a block diagonal matrix with diagonal blocks R_i . The matrices H and Z are similar:

$$H = D_R Z D_R^\top$$

The basic observation underpinning the eigenvector method follows. The expectation of the measurement matrix, or *synchronization matrix*, H contains the sought information:

$$\begin{aligned}\mathbb{E}\{Z\} &= \beta(\mathbf{1}_{N \times N} \otimes I_n) = \beta(\mathbf{1}_N \otimes I_n)(\mathbf{1}_N \otimes I_n)^\top, \\ \mathbb{E}\{H\} &= D_R \mathbb{E}\{Z\} D_R^\top = \beta R R^\top,\end{aligned}$$

¹In hindsight, this form of the noise model would have been a convenient choice for Chapter 8 also, as it encodes some symmetries of the problem perhaps more explicitly than the convention $Z_{ij} R_i R_j^\top$.

where $\mathbf{1}_N$ and $\mathbf{1}_{N \times N}$ are the vector and the matrix of all ones and \otimes is the Kronecker product. Indeed, $\mathbb{E}\{H\}$ has rank n and the sought matrix R is an orthonormal basis of its dominant eigenspace ($\mathbb{E}\{H\}R = \beta NR$) with top eigenvalues βN repeated n times. Because H and Z are similar, they share the same spectrum. Separate Z into its mean and random components:

$$Z = \mathbb{E}\{Z\} + Y.$$

Thus, Y is a symmetric, random matrix with zero diagonal blocks $Y_{ii} = 0$ and i.i.d., zero-mean above-diagonal blocks

$$Y_{ij} = Z_{ij} - \beta I_n.$$

The intuition goes as follows: H is a random symmetric matrix perturbed by a rank- n matrix βRR^\top which is to be estimated. To succeed, the perturbation should dominate the noise, which suggests the following algorithm: compute the top n eigenvectors of H to form an orthonormal matrix \hat{R} . If there is no noise, then $\sqrt{N}\hat{R} = RQ$ for some orthogonal matrix Q . If there is noise, then the hope is that \hat{R} will still be correlated with R and rounding the blocks of \hat{R} to rotation matrices would provide a meaningful estimator. For the perturbation to dominate the noise, it is necessary that the top n eigenvalues of H be separate from the spectrum of the noise. Since H and Z are similar, they have the same spectrum and we may study Z instead of H . Work on small rank perturbations of random (Wigner) matrices suggests that the top eigenvalues of Z pop out of the noise spectrum as soon as

$$\beta N > \frac{1}{2}\lambda_{\max}(Y) \tag{5.14}$$

and concentrate at $\beta N + \lambda_{\max}^2(Y)/4\beta N > \lambda_{\max}(Y)$ (Capitaine *et al.*, 2009). The theorems in (Capitaine *et al.*, 2009) do not apply directly to the present situation because Y does not have all of its entries independent: the entries inside a block Y_{ij} are dependent because of the constraint $Z_{ij} \in \text{SO}(n)$. Nevertheless, as N grows, the (constant) size n of the blocks becomes relatively small and it is expected that the phase transition will occur at the same point. This is indeed observed numerically and confirmed by the accuracy of the phase transition point prediction in Section 5.5.

Thus, we expect to see a phase transition point in the performance of the eigenvector method for N and β such that $2\beta N = \lambda_{\max}(Y)$. Girko (1995) studies the limiting spectral distribution of random symmetric matrices with independent blocks, which we now leverage to evaluate $\lambda_{\max}(Y)$. Consider $\tilde{Y} = Y/\sqrt{N}$: its above-diagonal blocks are centered, i.i.d. with Frobenius norm deterministically bounded by $1/\sqrt{N}$ times a constant and its diagonal

blocks are deterministically zero. Then, Girko's theorem applies and states that \tilde{Y} has a limiting spectral distribution $F_N(x)$ whose Stieltjes transform obeys

$$\forall z \in \mathbb{C}, \Im(z) \neq 0, \quad \int_{\mathbb{R}} \frac{1}{x - z} dF_N(x) = \frac{1}{nN} \sum_{k=1}^N \text{trace}(C_k(z)), \quad (5.15)$$

where for $k = 1, \dots, N$, the $n \times n$ matrices $C_k(z)$ satisfy

$$C_k(z) = - \left[zI_n + \sum_{s \neq k} \mathbb{E} \left\{ \tilde{Y}_{ks} C_s(z) \tilde{Y}_{ks}^\top \right\} \right]^{-1}.$$

The matrices $C_k(z)$ exist and are unique if one further constrain them to be analytic and to satisfy $\Im(z)\Im(C_k(z)) > 0$ when $\Im(z) \neq 0$.

Given the uniqueness of the solutions $C_k(z)$, it is satisfactory to propose a solution and check its validity. Try solutions of the form $C_k(z) = C(z) = \alpha(z)I_n$. Then, since

$$\mathbb{E} \left\{ \tilde{Y}_{ks} \tilde{Y}_{ks}^\top \right\} = \frac{1}{N} \mathbb{E} \left\{ (Z_{ks} - \beta I_n)(Z_{ks} - \beta I_n)^\top \right\} = \frac{1}{N} (1 - \beta^2) I_n,$$

the analytic function $\alpha(z)$ must obey

$$\left(z + \frac{N-1}{N} (1 - \beta^2) \cdot \alpha(z) \right) \cdot \alpha(z) = -1.$$

This defines a quadratic in $\alpha(z)$. The condition $\Im(z)\Im(C_k(z)) > 0$ when $\Im(z) \neq 0$ singles out one solution, leading to

$$\alpha(z) = \frac{-z + \sqrt{z^2 - 4 \left[\frac{N-1}{N} (1 - \beta^2) \right]}}{2 \left[\frac{N-1}{N} (1 - \beta^2) \right]}.$$

This well-defined solution validates the hypothesized form of the $C_k(z)$'s. Equation (5.15) implies that the Stieltjes transform of the limiting distribution $F_N(x)$ is $\alpha(z)$. Inverting the transform reveals that the limiting spectral distribution of $Y = \sqrt{N}\tilde{Y}$ follows, unsurprisingly, a semicircle law of radius 2σ with $\sigma = \sqrt{(N-1)(1-\beta^2)}$. The largest eigenvalue of Y is expected to concentrate at or near the edge of this compactly supported distribution,² that is, for large N ,

$$\lambda_{\max}(Y) \approx 2\sigma = 2\sqrt{(N-1)(1-\beta^2)}.$$

²This is not immediate. See for example work by Bai & Yin (1988) for a confirmation when the random matrix has all of its entries independent (Wigner model). Again, we stretch such results to apply them to Y , whose blocks are independent.

Plugging this into the condition (5.14), the n top eigenvalues of the synchronization matrix H are expected to jump out of the (semicircular) spectrum of the noise for β and N satisfying

$$\beta > \sqrt{\frac{N-1}{N^2+N-1}} \approx \frac{1}{\sqrt{N}}. \quad (5.16)$$

The next step in the analysis is to verify that when condition (5.14) is fulfilled and the top eigenvalues of H are separated from the noise, then the associated top eigenvectors correlate better than randomly with R . Thus, assume $\beta N = s\sigma$ for some $s > 1$. Let $U \in \mathbb{R}^{nN \times n}$ be such that $U^\top U = NI_n$ and the columns of U are n dominant eigenvectors of Z . The eigenvector method computes the dominant eigenvectors of H , i.e., $\hat{R} = D_R U$. Since (5.14) holds, the top n eigenvalues of Z all concentrate around $s\sigma + \sigma/s$. Thus, for large N ,

$$\text{trace}(U^\top ZU) = N \sum_{i=1}^n \lambda_i(Z) \approx nN(s+1/s)\sigma. \quad (5.17)$$

On the other hand, $Z = \mathbb{E}\{Z\} + Y$. Hence,

$$\begin{aligned} \text{trace}(U^\top ZU) &= \beta \text{trace}(U^\top (\mathbb{1}_N \otimes I_n)(\mathbb{1}_N \otimes I_n)^\top U) \\ &\quad + \text{trace}(U^\top YU). \end{aligned} \quad (5.18)$$

On the right hand side, the first term evaluates to $\beta \|\sum_{i=1}^N U_i\|_F^2$ where U is partitioned into blocks U_1, \dots, U_N of size n . This is a quality criterion for the (unrounded) eigenvector method, since $U_i = R_i^\top \hat{R}_i$. Indeed, successful estimation leads to $\hat{R} \approx RQ$ for some orthogonal matrix Q , that is, to

$$\left\| \sum_{i=1}^N U_i \right\|_F^2 = \left\| \sum_{i=1}^N R_i^\top \hat{R}_i \right\|_F^2 \approx \|NQ\|_F^2 = nN^2.$$

In comparison, when estimation fails completely, then U is a random matrix such that $U^\top U = NI_n$. Letting the entries of U be i.i.d. Gaussian with mean zero and variance $1/n$ yields matrices such that $\mathbb{E}\{U^\top U\} = NI_n$. For large N , $U^\top U$ concentrates around this expectation and provides an acceptable model of random dominant eigenvectors. For such U , it holds that

$$\mathbb{E} \left\{ \left\| \sum_{i=1}^N U_i \right\|_F^2 \right\} = nN.$$

Indeed, it is the sum of n^2 expectations of the square of sums of N i.i.d. Gaussian variables with variance $1/n$.

To observe better than random correlation of \hat{R} with R with high probability, it is thus necessary that (using equality of (5.17) and (5.18))

$$\beta \left\| \sum_{i=1}^N U_i \right\|_F^2 \approx nN(s + 1/s)\sigma - \text{trace}(U^\top Y U) > n\beta N = ns\sigma. \quad (5.19)$$

The term involving Y is bounded with high probability since the top eigenvalues of Y concentrate around 2σ . Hence, $\text{trace}(U^\top Y U) < 2nN\sigma$ and a sufficient condition for (5.19) is:

$$N(s + 1/s) - 2N > s.$$

The right hand side is negligible for large N and the condition reads:

$$s^2 - 2s + 1 = (s + 1)(s - 1) > 0.$$

Therefore, as soon as $s > 1$ (which we already had to assume to let the dominant eigenvalues of H pop out of the semicircle), that is, as soon as (5.16) holds, we may expect the eigenvector method to return a better than random estimator of R .

It remains to show that under Assumptions 8.1–8.3 from Chapter 8, it indeed holds that $\mathbb{E}\{Z_{ij}\} = \beta I_n$. In doing so, we use tools which will be introduced in Section 8.3. Notably, μ denotes the Haar measure on the group of rotations.

To this end, let Z be a random rotation matrix in $\text{SO}(n)$ with probability density function f and let f be a spectral function, that is, $f(QZQ^\top) = f(Z)$ for all orthogonal Q . Then,

$$\mathbb{E}\{Z\} = \int_{\text{SO}(n)} Z f(Z) d\mu(Z) = \int_{\text{SO}(n)} \exp(\log(Z)) f(Z) d\mu(Z).$$

The second equality holds because, restricted to $\text{SO}(n)$, the matrix exponential and logarithm \exp and \log are smooth and inverse of each other. Expand the matrix exponential in Taylor series:

$$\mathbb{E}\{Z\} = \sum_{k=0}^{\infty} \frac{1}{k!} \int_{\text{SO}(n)} \log^k(Z) f(Z) d\mu(Z) := \sum_{k=0}^{\infty} \frac{1}{k!} A_k,$$

with $A_k = \mathbb{E}\{\log^k(Z)\}$. Since $\log(Z)$ is skew-symmetric, for k odd A_k is skew-symmetric too. For any orthogonal Q , the change of variable $Z \mapsto QZQ^\top$ in the integral below shows that

$$A_k = \int_{\text{SO}(n)} \log^k(Z) f(Z) d\mu(Z) = \int_{\text{SO}(n)} \log^k(QZQ^\top) f(Z) d\mu(Z) = Q A_k Q^\top.$$

This holds because $\text{SO}(n)$, f and $d\mu$ are invariant under the change of variable and $\log(QZQ^\top) = Q \log(Z) Q^\top$. Since A_k is skew-symmetric, it is a normal matrix and there exists an orthogonal matrix Q such that $QA_kQ^\top = A_k^\top$. Indeed, A_k and A_k^\top share the same spectrum. Therefore, $A_k = A_k^\top = -A_k = 0$. For k even, A_k is symmetric and it similarly holds that $A_k = QA_kQ^\top$ for all orthogonal Q . In particular, since A_k is symmetric, we may choose Q such that QA_kQ^\top is diagonal, showing that A_k has to be diagonal. Now let Q be a permutation matrix to see that the diagonal entries of A_k have to be equal, that is, $A_k = c_k I_n$ for some constant c_k . Finally, it holds as expected that

$$\mathbb{E}\{Z\} = \sum_{k=0}^{\infty} \frac{c_{2k}}{(2k)!} I_n = \beta I_n.$$

In practice, it is instructive to compute β for certain noise models. Since $\text{trace}(Z) = n\beta$, β may be obtained by evaluating this integral of a class function over $\text{SO}(n)$, as instructed in Appendix A:

$$\beta = \frac{1}{n} \int_{\text{SO}(n)} \text{trace}(Z) f(Z) d\mu(Z) = \frac{1}{n} \mathbb{E}\{\text{trace}(Z)\}.$$

In particular, for noise matrices Z_{ij} distributed following a Langevin (5.4) and for $n = 2, 3$, $\beta_n(\kappa)$ is given by:

$$\beta_2(\kappa) = \frac{I_1(2\kappa)}{I_0(2\kappa)}, \quad \beta_3(\kappa) = \frac{1}{3} \frac{I_1(2\kappa) - I_2(2\kappa)}{I_0(2\kappa) - I_1(2\kappa)}, \quad (5.20)$$

where $I_\nu(x)$ is the modified Bessel function of the first kind (A.4). It is easily checked that $\beta_n(\kappa)$ increases monotonically with κ and that $\beta_n(0) = 0$ and $\beta_n(\infty) = 1$. For the mixture of Langevin model (5.5), it holds that

$$\beta_n(\kappa, \kappa', p) = p\beta_n(\kappa) + (1-p)\beta_n(\kappa').$$

For the perfect-or-outlier noise model $\kappa = \infty, \kappa' = 0$ in (Singer, 2011), this evaluates to p and one recovers the phase transition point $p = 1/\sqrt{N}$.

5.4 An algorithm to compute the maximum likelihood estimator

We now propose an algorithm to compute $\hat{\mathbf{R}}_{\text{MLE}}$. Because the optimization problem (5.8) is nonconvex, we only guarantee the computation of a local maximizer, so that our “MLE” is really only a proxy for the true $\hat{\mathbf{R}}_{\text{MLE}}$.

Nevertheless, Section 5.5 shows that the algorithm performs well in practice, as compared to Cramér-Rao bounds.

The parameter space \mathcal{P}_A (5.7) is a Riemannian submanifold of $(\mathbb{R}^{n \times n})^N$. The log-likelihood function

$$L_A = L|_{\mathcal{P}_A}, \quad (5.21)$$

that is, the restriction of L (5.6) to \mathcal{P}_A , is a smooth objective function defined over that manifold. Maximizing L_A over \mathcal{P}_A is thus an instance of a smooth optimization problem on a manifold, as covered in Chapter 3.

In this section, we start by describing a procedure to obtain an initial guess (a first iterate). It is based on the eigenvector method presented in the previous section. We then go on to establish the gradient and the Hessian of the cost function L_A to be maximized. The second-order Riemannian trust-region method from Section 3.2 is then applied within the Manopt framework to improve on the initial guess, exploiting the gradient and Hessian information.

Notice that the parameters of the noise model (κ , κ' and p) are assumed known at first. In practice, these have to be estimated from the data. We propose one approach which we call MLE+ in Section 5.4.5. It is tested on real data in Section 5.6, with convincing performance.

5.4.1 An initial guess based on a spectral relaxation

Depending on the initial guess (the initial iterate) $\hat{\mathbf{R}}^{(0)}$, the iterative optimization algorithm used may converge to different critical points. Heuristically, to increase the chances of converging to a “good” critical point (ideally, the global optimizer), we want $\hat{\mathbf{R}}^{(0)}$ to be a decent estimator itself. For that purpose, convex relaxations of the synchronization problem, such as the max-cut-like relaxation for synchronization (Arie-Nachimson *et al.*, 2012; Singer, 2011) and the more robust LUD method (Wang & Singer, 2013), are prime candidates. Unfortunately, they tend to be costly to compute. On the other hand, the spectral relaxations of the synchronization problem developed in (Singer, 2011) for $\text{SO}(2)$ then (Singer & Shkolnisky, 2011) for $\text{SO}(3)$ and finally in (Bandeira *et al.*, 2013b) for the general case are suitable to produce cheap yet good solutions. Here, we show how (Bandeira *et al.*, 2013b, Algorithm 16) can be used to produce $\hat{\mathbf{R}}^{(0)}$ in \mathcal{P}_A from a structured eigenvalue problem. Algorithm 5 summarizes the procedure.

Let $D \in \mathbb{R}^{N \times N}$ be a diagonal matrix such that $D_{ii} = \sum_{i \sim j} \kappa$, i.e., κ times the degree of node i . Following notations in (Bandeira *et al.*, 2013b), define $D_1 = D \otimes I_n$ (Kronecker product). Let $W_1 \in \mathbb{R}^{nN \times nN}$ be a symmetric matrix composed of $n \times n$ blocks such that the (i, j) -block $(W_1)_{ij}$ is κH_{ij} if nodes i and j are connected, and zero otherwise.

Let $X \in \mathbb{R}^{nN \times n}$ be composed of N stacked $n \times n$ blocks X_1, \dots, X_N . Consider the following quadratic expressions:

$$X^\top D_1 X = \sum_{i=1}^N D_{ii} X_i^\top X_i, \quad (5.22)$$

$$X^\top W_1 X = \sum_{i \sim j} \kappa X_i^\top H_{ij} X_j + \kappa X_j^\top H_{ji} X_i. \quad (5.23)$$

Maximizing $\text{trace}(X^\top W_1 X)$ subject to $X_i \in \text{SO}(n)$ is equivalent to computing the maximum likelihood estimator for synchronization under a Langevin prior (see Remark 5.1, eq. (5.9)). This is difficult because of the nonconvexity of the constraints. Now observe that, under these same constraints, $X^\top D_1 X = \text{trace}(D) I_n$. If we relax and simply impose the latter, i.e., that the columns of X be D_1 -orthogonal, then maximizing $\text{trace}(X^\top W_1 X)$ becomes easy: it is a generalized eigenvector problem with pencil (W_1, D_1) .

This observation underpins (Bandeira *et al.*, 2013b, Algorithm 16). Compute the n dominant D_1 -orthonormal eigenvectors of W_1 , i.e., compute $X \in \mathbb{R}^{nN \times n}$ as the solution of (notice that the scaling of X is irrelevant as long as it is fixed):

$$\max_X \text{trace}(X^\top W_1 X) \text{ such that } X^\top D_1 X = I_n. \quad (5.24)$$

The global optimum of this problem can be computed efficiently, for example using `eigs` in Matlab.

In a noiseless scenario, the blocks X_i in the obtained solution will be orthogonal matrices (up to scaling). Because of noise in the measurements, this is, in general, not the case and one needs to project the X_i 's to construct a feasible solution for the original problem. The proposed rounding procedure is to project each block to $\text{SO}(n)$ as $R_i^{(a)} = \Pi_{\text{SO}(n)}(X_i)$, where $\Pi_{\text{SO}(n)} : \mathbb{R}^{n \times n} \rightarrow \text{SO}(n)$ assigns to $R_i^{(a)}$ the rotation matrix that is closest to X_i in the sense of the Frobenius norm in $\mathbb{R}^{n \times n}$. This may be computed via the SVD decomposition $S = U \Sigma V^\top$, $s = \det(UV^\top)$ (Sarlette & Sepulchre, 2009):

$$\Pi_{\text{SO}(n)}(S) = U \text{diag}(1, \dots, 1, s) V^\top, \quad (5.25)$$

where Σ is diagonal with decreasing entries and s is either 1 or -1 since U and V are orthogonal. As long as the smallest singular value of S has multiplicity one, this is uniquely defined (Sarlette & Sepulchre, 2009, Prop. 3.3).

The solution X of the eigenvalue problem (5.24) is defined up to an orthogonal transformation. This means that even in the noiseless case where the individual blocks X_i would be orthogonal (up to scaling), they could

Algorithm 5 EIG (anchored): Computes the initial guess $\hat{\mathbf{R}}^{(0)}$

- 1: Form the sparse matrices D_1 (5.22) and W_1 (5.23) ;
 - 2: Compute $X \in \mathbb{R}^{nN \times n}$, the dominant eigenvectors of the pencil (W_1, D_1) [Matlab: $[X, \sim] = \text{eigs}(W_1, D_1, n)$] ;
 - 3: **for all** $i \in 1 \dots N$ **do**
 - 4: $R_i^{(a)} = \Pi_{\text{SO}(n)}(X_i)$ and $R_i^{(b)} = \Pi_{\text{SO}(n)}(X_i J)$;
 - 5: **end for**
 - 6: $\tilde{\mathbf{R}} = \begin{cases} \mathbf{R}^{(a)} & \text{if } L(\mathbf{R}^{(a)}) \geq L(\mathbf{R}^{(b)}), \\ \mathbf{R}^{(b)} & \text{otherwise;} \end{cases}$
 - 7: Anchor alignment: $Q = \Pi_{\text{SO}(n)}\left(\sum_{i \in A} \tilde{R}_i^\top R_i\right)$;
 - 8: **for all** $i \in 1 \dots N$ **do**
 - 9: $\hat{R}_i^{(0)} = \begin{cases} R_i & \text{if } i \in A, \\ \tilde{R}_i Q & \text{otherwise;} \end{cases}$
 - 10: **end for**
-

turn out not to be rotation matrices, having negative determinant. To resolve this ambiguity, we also compute the projections of XJ , with $J = \text{diag}(1, \dots, 1, -1)$. Compute $R_i^{(b)} = \Pi_{\text{SO}(n)}(X_i J)$. Finally, keep either $\mathbf{R}^{(a)}$ or $\mathbf{R}^{(b)}$ depending on which is more likely (eq. (5.6)). That is, set $\tilde{\mathbf{R}} = \mathbf{R}^{(a)}$ if $L(\mathbf{R}^{(a)}) \geq L(\mathbf{R}^{(b)})$, and $\tilde{\mathbf{R}} = \mathbf{R}^{(b)}$ otherwise.

This procedure yields an initial guess of rotations $\tilde{\mathbf{R}}$ that does not, in general, comply with the anchor constraints. We thus further globally align \mathbf{R} with the anchors by computing (Sarlette & Sepulchre, 2009):

$$Q = \min_{Q \in \text{SO}(n)} \sum_{i \in A} \|R_i - \tilde{R}_i Q\|_{\text{F}}^2 = \Pi_{\text{SO}(n)}\left(\sum_{i \in A} \tilde{R}_i^\top R_i\right).$$

The initial guess for the optimization step is $\hat{\mathbf{R}}^{(0)}$, where $\hat{R}_i^{(0)}$ is set to R_i if node i is anchored and to $\tilde{R}_i Q$ otherwise.

5.4.2 Gradient of the log-likelihood L_A

The function L_A (5.21) is defined on \mathcal{P}_A (5.7), a Riemannian submanifold of $(\mathbb{R}^{n \times n})^N$ endowed with the usual inner product

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i=1}^N \text{trace}(X_i^\top Y_i). \quad (5.26)$$

The gradient of a L_A at $\hat{\mathbf{R}}$ is a tangent vector which we now compute.

Let \bar{L} be the function defined on $(\mathbb{R}^{n \times n})^N$ by the same analytic formula as L (5.6), such that L_A is merely the restriction of \bar{L} to \mathcal{P}_A :

$$\bar{L}: (\mathbb{R}^{n \times n})^N \rightarrow \mathbb{R}, \quad \bar{L}(\hat{\mathbf{R}}) = \sum_{i \sim j} \log f(\hat{R}_i^\top H_{ij} \hat{R}_j).$$

(We permuted the matrices in the argument to f , which is fine since f only depends on the trace of its input.) The gradient of \bar{L} can be computed in the usual way. Because \mathcal{P}_A is a Riemannian submanifold of $(\mathbb{R}^{n \times n})^N$, the gradient of L_A at a point $\hat{\mathbf{R}} \in \mathcal{P}_A$ is related to the gradient of \bar{L} by this simple equation (see Section 2.3):

$$\text{grad } L_A(\hat{\mathbf{R}}) = \text{Proj}_{\hat{\mathbf{R}}}(\text{grad } \bar{L}(\hat{\mathbf{R}})),$$

where $\text{Proj}_{\hat{\mathbf{R}}}$ is the orthogonal projector (w.r.t. the metric (5.26)) from the ambient space $(\mathbb{R}^{n \times n})^N$ to the tangent space to \mathcal{P}_A at $\hat{\mathbf{R}}$ (see Section 5.2). Explicitly, the i^{th} component of the gradient of L_A , that is, the gradient of L_A w.r.t. the i^{th} rotation \hat{R}_i , is given by:

$$\text{grad}_i L_A(\hat{\mathbf{R}}) = \begin{cases} \hat{R}_i \text{skew}(\hat{R}_i^\top \text{grad}_i \bar{L}(\hat{\mathbf{R}})) & \text{if } i \notin A, \\ 0 & \text{if } i \in A. \end{cases} \quad (5.27)$$

Gradient components pertaining to anchored rotations are forced to zero by the projector since these rotations cannot move. The other components are projected to a form $\hat{R}_i \Omega_i$ where Ω_i is skew-symmetric.

By definition, $\text{grad}_i \bar{L}(\hat{\mathbf{R}})$ is the unique matrix in $\mathbb{R}^{n \times n}$ satisfying, for all X in $\mathbb{R}^{n \times n}$,

$$\text{trace}(X^\top \text{grad}_i \bar{L}(\hat{\mathbf{R}})) = D_i \bar{L}(\hat{\mathbf{R}})[X], \quad (5.28)$$

where the right hand side is the directional derivative of \bar{L} at $\hat{\mathbf{R}}$ w.r.t. the i^{th} rotation \hat{R}_i along the direction X . In order to compute the gradient of \bar{L} , we thus compute its directional derivatives and proceed by identification in (5.28). Let us define

$$\hat{Z}_{ij} \triangleq \hat{R}_i^\top H_{ij} \hat{R}_j.$$

By the chain rule,

$$D_i \bar{L}(\hat{\mathbf{R}})[X] = \sum_{i \sim j} \frac{1}{f(\hat{Z}_{ij})} Df(\hat{Z}_{ij})[X^\top H_{ij} \hat{R}_j]. \quad (5.29)$$

The summation is over the nodes j that are neighbors of node i . The differential of f (5.5) is obtained as follows:

$$\begin{aligned} Df(Z)[Y] &= p D\ell_\kappa(Z)[Y] + (1 - p) D\ell_{\kappa'}(Z)[Y], \\ D\ell_\kappa(Z)[Y] &= \kappa \ell_\kappa(Z) \text{trace}(Y). \end{aligned} \quad (5.30)$$

Combining (5.29)–(5.30), we further obtain:

$$\begin{aligned} D_i \bar{L}(\hat{\mathbf{R}})[X] &= \sum_{i \sim j} g(\hat{Z}_{ij}) \operatorname{trace}(X^\top \hat{R}_i \hat{Z}_{ij}), \\ g(\hat{Z}_{ij}) &= \frac{p\kappa \ell_\kappa(\hat{Z}_{ij}) + (1-p)\kappa' \ell_{\kappa'}(\hat{Z}_{ij})}{f(\hat{Z}_{ij})}. \end{aligned} \quad (5.31)$$

By identification with (5.28) and in combination with (5.27), this establishes the gradient of L_A :

$$\operatorname{grad}_i L_A(\hat{\mathbf{R}}) = \begin{cases} \hat{R}_i \sum_{i \sim j} g(\hat{Z}_{ij}) \operatorname{skew}(\hat{Z}_{ij}) & \text{if } i \notin A, \\ 0 & \text{if } i \in A. \end{cases} \quad (5.32)$$

Notice that the i^{th} component of the gradient can be computed based solely on the information pertaining to node i and its neighbors. This hints toward gradient-based decentralized synchronization algorithms (which we do not discuss).

5.4.3 Hessian of the log-likelihood L_A

Second-order optimization algorithms on Riemannian manifolds require the computation of the Riemannian Hessian of the objective function. For the particular case of Riemannian submanifolds such as \mathcal{P}_A , the Hessian admits a simple formulation in terms of the differential of the gradient in the ambient space.

For unanchored nodes ($i \notin A$), introduce the functions $G_i: (\mathbb{R}^{n \times n})^N \rightarrow \mathbb{R}^{n \times n}$ (see (5.31) for g):

$$G_i(\hat{\mathbf{R}}) = \hat{R}_i \sum_{i \sim j} g(\hat{Z}_{ij}) \operatorname{skew}(\hat{Z}_{ij}).$$

From (5.32), we know that the restriction of G_i to \mathcal{P}_A yields the i^{th} gradient component of L_A . Then, following Section 2.4, the i^{th} component of the Hessian of L_A at $\hat{\mathbf{R}}$ applied to the tangent vector $\hat{\mathbf{R}}\Omega$ is given by (Ω is a tuple of skew-symmetric matrices):

$$\operatorname{Hess}_i L_A(\hat{\mathbf{R}})[\hat{\mathbf{R}}\Omega] = \hat{R}_i \operatorname{skew}(\hat{R}_i^\top D G_i(\hat{\mathbf{R}})[\hat{\mathbf{R}}\Omega]).$$

That is, it is sufficient to differentiate the gradient vector field in the ambient space and then to (orthogonally) project the resulting vector field to the

tangent spaces of \mathcal{P}_A . By the chain rule and the product rule:

$$\begin{aligned}\hat{R}_i^\top \mathrm{D}G_i(\hat{\mathbf{R}})[\hat{\mathbf{R}}\Omega] &= \sum_{i \sim j} \mathrm{D}g(\hat{Z}_{ij})[\hat{\Omega}_{ij}] \operatorname{skew}\left(\hat{Z}_{ij}\right) \\ &\quad + \Omega_i \sum_{i \sim j} g(\hat{Z}_{ij}) \operatorname{skew}\left(\hat{Z}_{ij}\right) + \sum_{i \sim j} g(\hat{Z}_{ij}) \operatorname{skew}\left(\hat{\Omega}_{ij}\right),\end{aligned}$$

where $\hat{\Omega}_{ij}$ is the directional derivative of \hat{Z}_{ij} when \hat{R}_i and \hat{R}_j are moved (infinitesimally) along $\hat{R}_i\Omega_i$ and $\hat{R}_j\Omega_j$:

$$\hat{\Omega}_{ij} = \Omega_i^\top \hat{R}_i^\top H_{ij} \hat{R}_j + \hat{R}_i^\top H_{ij} \hat{R}_j \Omega_j = \hat{Z}_{ij} \Omega_j - \Omega_i \hat{Z}_{ij}.$$

This is not, in general, a skew-symmetric matrix. Some algebra yields the following identity:

$$\mathrm{D}g(\hat{Z}_{ij})[\hat{\Omega}_{ij}] = \left(\frac{p\kappa^2 \ell_\kappa(\hat{Z}_{ij}) + (1-p)\kappa'^2 \ell_{\kappa'}(\hat{Z}_{ij})}{f(\hat{Z}_{ij})} - g^2(\hat{Z}_{ij}) \right) \operatorname{trace}(\hat{\Omega}_{ij}).$$

Combining equations in this subsection yields an explicit expression for the component $\operatorname{Hess}_i L_A(\hat{\mathbf{R}})[\hat{\mathbf{R}}\Omega]$ for non-anchored nodes. For anchored nodes, $\operatorname{Hess}_i L_A(\hat{\mathbf{R}})$ vanishes.

5.4.4 Maximizing the likelihood

We use the second-order Riemannian trust-region method described in Section 3.2 to maximize the likelihood over \mathcal{P}_A . This method converges globally (that is, from any initial guess) toward critical points (typically local optimizers) with quadratic local convergence. The initial guess is set as discussed previously, based on the eigenvector method.

The optimization algorithm is stopped once the norm of the gradient drops below $10^{-6}/|\mathcal{E}|$, where $|\mathcal{E}|$ is the number of measurements. The maximum trust-region radius is set to $\bar{\Delta} = \pi \sqrt{n(N - |A|)}$, which scales like the diameter of the compact manifold \mathcal{P}_A ; the initial radius is $\Delta_0 = \bar{\Delta}/8$. We allow up to 100 Hessian evaluations to solve each inner problem, but seldom if ever use that many. The other parameters are set to their default value.

5.4.5 MLE+: estimating both the noise distribution and the rotations

When the mixture parameters κ, κ' and p are unknown, which is the case in most if not all applications, it is desirable to estimate these parameters

from the data. Given an estimator $\hat{\mathbf{R}}$ for the rotations, an estimator for the noise matrices Z_{ij} is given by

$$\hat{Z}_{ij} = \hat{R}_i^\top H_{ij} \hat{R}_j.$$

(As only the trace of these will matter, the ordering is not important.) The rotations \hat{Z}_{ij} constitute an estimate of a sample of the noise distribution. Assuming the noise model is parametrized by κ, κ' and p , the log-likelihood of the \hat{Z}_{ij} 's is $\sum_{i \sim j} \log f(\hat{Z}_{ij})$ (see (5.5)). Then, a maximum likelihood estimator for κ, κ' and p can be obtained by maximizing the above quantity for fixed \hat{Z}_{ij} 's. This suggests estimating the parameters by minimizing the following function:

$$g(\kappa, \kappa', p) = -\frac{1}{M} \sum_{i \sim j} \log \left(p \ell_\kappa(\hat{Z}_{ij}) + (1-p) \ell_{\kappa'}(\hat{Z}_{ij}) \right). \quad (5.33)$$

It is easily seen that the derivative of the Langevin normalization coefficients $c_n(\kappa)$ (8.10) is given by $c'_n(\kappa) = n\beta_n(\kappa)c_n(\kappa)$, see (5.20). The derivative of ℓ_κ w.r.t. κ ensues:

$$\frac{\partial}{\partial \kappa} \ell_\kappa(Z) = (\text{trace}(Z) - n\beta_n(\kappa)) \ell_\kappa(Z).$$

Thus, the gradient of g follows:

$$\begin{aligned} \frac{\partial}{\partial \kappa} g(\kappa, \kappa', p) &= -\frac{1}{M} \sum_{i \sim j} \frac{p}{f(\hat{Z}_{ij})} \frac{\partial}{\partial \kappa} \ell_\kappa(\hat{Z}_{ij}), \\ \frac{\partial}{\partial \kappa'} g(\kappa, \kappa', p) &= -\frac{1}{M} \sum_{i \sim j} \frac{1-p}{f(\hat{Z}_{ij})} \frac{\partial}{\partial \kappa'} \ell_{\kappa'}(\hat{Z}_{ij}), \\ \frac{\partial}{\partial p} g(\kappa, \kappa', p) &= -\frac{1}{M} \sum_{i \sim j} \frac{1}{f(\hat{Z}_{ij})} \left(\ell_\kappa(\hat{Z}_{ij}) - \ell_{\kappa'}(\hat{Z}_{ij}) \right). \end{aligned}$$

The function g is to be minimized under the constraints that $\kappa, \kappa' > 0$ and that $0 \leq p \leq 1$. Furthermore, the concentration parameters scale logarithmically. This motivates the introduction of \tilde{g} , defined without constraints:

$$\tilde{g}(\gamma, \gamma', q) = g \left(\kappa = e^\gamma, \kappa' = e^{\gamma'}, p = \frac{1 + \cos q}{2} \right).$$

Algorithm 6 MLE+ : Alternate maximum likelihood estimation of $\hat{\mathbf{R}}$ and of κ, κ', p .

Require: Initial estimates: $\hat{\mathbf{R}}, \hat{\kappa}, \hat{\kappa}', \hat{p}$.

- 1: **for** $i = 1 \dots \text{max number of iterations}$ **do**
- 2: Estimate the noise: $\forall i \sim j, \hat{Z}_{ij} := \hat{R}_i^\top H_{ij} \hat{R}_j$;
- 3: Compute new values for $\hat{\kappa}, \hat{\kappa}', \hat{p}$ by minimizing g (5.33),
- 4: with the present values to build the initial guess (Section 5.4.5) ;
- 5: **if** the parameter estimate did not change significantly **then**
- 6: **Stop.**
- 7: **else**
- 8: Compute a new estimator $\hat{\mathbf{R}}$ using the proposed MLE method
- 9: (Section 5.4.4), with the present $\hat{\mathbf{R}}$ as initial guess ;
- 10: **end if**
- 11: **end for**

Its gradient is tied to that of g :

$$\begin{aligned}\frac{\partial}{\partial \gamma} \tilde{g}(\gamma, \gamma', q) &= \kappa \frac{\partial}{\partial \kappa} g(\kappa, \kappa', p), \\ \frac{\partial}{\partial \gamma'} \tilde{g}(\gamma, \gamma', q) &= \kappa' \frac{\partial}{\partial \kappa'} g(\kappa, \kappa', p), \\ \frac{\partial}{\partial q} \tilde{g}(\gamma, \gamma', q) &= -\frac{\sin q}{2} \frac{\partial}{\partial p} g(\kappa, \kappa', p).\end{aligned}$$

For $p \notin \{0, 1\}$, it holds that $\sin q \neq 0$ and the critical points of g are 1-to-1 with the critical points of \tilde{g} . For $p \in \{0, 1\}$, \tilde{g} might be at a critical points that does not correspond to a critical point of g , so that we never let $p \in \{0, 1\}$ in an initial guess. (An alternative change of variable for p is the sigmoid $p = \tanh(q)$, which is such that all critical points of \tilde{g} correspond to critical points of g but it excludes the values 0 and 1 for p altogether.)

A strategy to estimate the noise model parameters appears clearly now: given an estimator $\hat{\mathbf{R}}$ and an a priori guess of κ, κ', p , compute the matrices \hat{Z}_{ij} and apply the change of variables $\gamma = \log(\kappa), \gamma' = \log(\kappa'), q = \arccos(2p - 1)$. In practice, prior to the change of variables, we project p to $[0.01, 0.99]$ to avoid a spurious zero derivative along that direction, as per the discussion above. Furthermore, we project κ and κ' to $[10^{-6}, 10^6]$ to avoid numerical breakdown. Using any solver for smooth, unconstrained optimization problems (we use the conjugate gradient solver in Manopt on the Euclidean manifold \mathbb{R}^3 , see Section 3.1), find a critical point of \tilde{g} (hopefully a good minimizer) starting from (γ, γ', q) as initial guess. Apply the reverse change of variable to obtain a new estimate of the parameters κ, κ', p . Unfortunately, given that \tilde{g} is nonconvex, the initial estimate of the

parameters typically influences the outcome.

Table 5.1 reports on numerical experiments where the parameters of a known mixture of Langevin are estimated from a pseudorandom sample (see Remark 5.3), using the procedure described above with different initial guesses. The overall accuracy is excellent. On a desktop computer from 2010, the median estimation time is 0.10 second, 75% of the estimations run in under 0.22 second and the slowest estimation lasts 4.98 seconds.

Of course, the procedure using $\hat{\mathbf{R}}$ to estimate κ, κ', p can be iterated, as we now have new values for the mixture parameters which lead to a new estimator $\hat{\mathbf{R}}$. This suggests Algorithm 6, which we refer to as MLE+. There is no guarantee that this procedure always converges, but we observe excellent practical behavior in Section 5.6.

The initial guess for step 8 in Algorithm 6 is important. If κ is large and κ' is small, then poor estimators are located in almost flat regions of the likelihood function, essentially jamming the iteration. This also means that it may not be practical to estimate the parameters of the noise distribution for a given application once and for all: reaching the final estimator iteratively may be a necessary ingredient of Algorithm 6.

5.5 Numerical experiments

We now perform a few experiments on synthetic data to showcase properties of the proposed maximum likelihood estimator. Our main goal in this section is to study the performance of the MLE compared to the theoretical limits established in Chapter 8 in the form of Cramér-Rao bounds (CRB's). Hence, in all tests, the measurements are generated following the noise model proposed in Section 5.1 and the specific parameters (κ, κ' and p) are known to the algorithm. We will see that, under these favorable conditions, the MLE appears to reach the CRB in many cases. This suggests two appreciable conclusions: (i) the proposed estimator appears to be asymptotically efficient in spite of the nonconvex nature of the maximum likelihood problem, and (ii) if the CRB's are tight, then their interpretation gives valuable insight into the synchronization problem (Section 8.7). We further observe that the MLE tends to concentrate the estimation error on a few rotations and suggest a PageRank-like procedure to detect these.

5.5.1 Performance criterion and Cramér-Rao bounds

For a given estimator $\hat{\mathbf{R}}$ of the true rotations \mathbf{R} with anchors indexed by A , assuming there is at least one anchor, the performance criterion we choose is

the mean squared error (MSE) based on the geodesic distance on \mathcal{P}_A (5.13):

$$\text{MSE}(\mathbf{R}, \hat{\mathbf{R}}) = \frac{1}{N - |A|} \sum_{i \notin A} \|\log(R_i^\top \hat{R}_i)\|_F^2.$$

For rotations in the plane or in space ($n = 2$ or 3), $\|\log(R_i^\top \hat{R}_i)\|_F / \sqrt{2}$ is the angle in radians of the rotation $R_i^\top \hat{R}_i$. For small errors, $\|\log(R_i^\top \hat{R}_i)\|_F \approx \|R_i - \hat{R}_i\|_F$.

In the absence of anchors, this performance criterion is unsuitable because the sought rotations can only be recovered up to a global rotation—see Section 8.5.2 for an alternative.

Because measurements are noisy, there is no hope to reduce the MSE to zero all the time. Chapter 8 establishes that the expected MSE for synchronization is lower-bounded by some number which heavily relies on the topology of the measurement graph. The relevant features of the topology of the graph are captured by the spectrum of the *graph Laplacian*. We give an executive overview of these bounds here, as a motivation for Chapter 8.

Define the *information weight* of a measurement $Z R_i R_j^\top$ as in (8.22):

$$w = \mathbb{E} \left\{ \|\text{grad} \log f(Z)\|^2 \right\}, \quad (5.34)$$

where the expectation is taken w.r.t. Z distributed with pdf f (5.5). In the extreme case, if Z is uniformly distributed over $\text{SO}(n)$, then f is constant and w is zero, i.e., the measurement contains no information. The more f is concentrated (that is, the less uncertainty there is), the larger the gradient of $\log f$ and thus the larger w is. A formula for $w = w_n(\kappa, \kappa', p)$ is derived in Appendix A.2 using Weyl's integration formula. Numerically computable formulas for the special case $\kappa' = 0$ are given explicitly in Example 8.5.

Let us weigh each edge of the measurement graph with w . The Laplacian of the resulting graph is the symmetric, positive semidefinite matrix $\mathcal{L} \in \mathbb{R}^{N \times N}$ defined by:

$$\mathcal{L}_{ij} = \begin{cases} wd_i & \text{if } i = j, \\ -w & \text{if } i \sim j, \\ 0 & \text{otherwise,} \end{cases}$$

where d_i is the degree of node i . Further define the masked Laplacian \mathcal{L}_A which is obtained by forcing to zero the rows and columns of \mathcal{L} that correspond to anchored rotations:

$$(\mathcal{L}_A)_{ij} = \begin{cases} \mathcal{L}_{ij} & \text{if } i, j \notin A, \\ 0 & \text{otherwise.} \end{cases}$$

Then, the CRB on the expected MSE of any unbiased estimator $\hat{\mathbf{R}}$ for the anchored synchronization problem is lower-bounded as follows:

$$\mathbb{E} \left\{ \text{MSE}(\mathbf{R}, \hat{\mathbf{R}}) \right\} \geq \frac{(n(n-1)/2)^2}{N - |A|} \text{trace}(\mathcal{L}_A^\dagger) + \text{curvature terms},$$

where \dagger denotes Moore-Penrose pseudoinversion. This bound is valid in a large signal-to-noise ratio (SNR) regime. The curvature terms vanish for $n = 2$ and are negligible at large SNR for $n \geq 3$.

For complete graphs with one anchor, at large SNR, the CRB for rotations in $\text{SO}(3)$ reduces to:

$$\mathbb{E} \left\{ \text{MSE}(\mathbf{R}, \hat{\mathbf{R}}) \right\} \geq \frac{18}{wN} \left(1 - \frac{1}{wN} \right),$$

where the term $-1/wN$ accounts for the curvature terms. The larger the SNR (that is, the larger κ, κ' and p), the larger w (5.34) and the lower the CRB.

Remark 5.2 (The unbiasedness assumption). *The CRB's constrain the variance of unbiased estimators (Definition 6.5). It is hence not entirely clear that the CRB's are applicable to the estimators at hand before these are shown to be unbiased. Intuitively, one expects this to be the case, given the strong symmetries of the problem. Unfortunately, there is currently no proof supporting this statement.*

Because the parameter space \mathcal{P}_A is compact, even estimators that disregard measurements completely and return a random estimator would have finite MSE. Any reasonable estimator should perform at least as well as a random estimator. Hence, an upperbound on the MSE for rotations in $\text{SO}(3)$ is given in Section 8.7.1:

$$\mathbb{E} \left\{ \text{MSE}(\mathbf{R}, \hat{\mathbf{R}}) \right\} \leq \frac{2\pi^2}{3} + 4.$$

5.5.2 The least unsquared deviation algorithm (LUD)

We here describe the LUD algorithm introduced by Wang & Singer (2013) and against which we compare in the experiments below. Consider this formulation of anchor-free (or single-anchor) synchronization:

$$\min_{\hat{R}_1, \dots, \hat{R}_N \in \text{SO}(n)} \sum_{i \sim j} \|\hat{R}_i \hat{R}_j^\top - H_{ij}\|_F^q. \quad (5.35)$$

The constraints are not convex and, unsurprisingly, this is a difficult problem to solve to global optimality. As has been observed in (Arie-Nachimson

(*et al.*, 2012; Singer, 2011), letting $q = 2$ makes it possible to relax (5.35) to an SDP which can be solved globally in polynomial time (up to some precision). Unfortunately, even though it can be proven that these relaxations perform well, it remains true that least-squares loss functions are not adequate to cope with outliers. Furthermore, empirically, these SDP's appear to not perform significantly better than the eigenvector method, which is both faster and simpler.

As a reaction, Wang & Singer (2013) suggest letting $q = 1$, hence the name LUD for their method. The unsquared loss does not unduly penalize large errors and thus better accommodates outliers. The relaxation goes as follows. Let

$$\hat{R} = (\hat{R}_1^\top \quad \dots \quad \hat{R}_N^\top)^\top, \quad G = \hat{R}\hat{R}^\top.$$

Then, relaxing (5.35) a first time to only enforce $\hat{R}_i \in O(n)$, that is, simply dropping the determinant constraints, yields this program:

$$\begin{aligned} \min_{G \in \mathbb{R}^{nN \times nN}} & \sum_{i \sim j} \|G_{ij} - H_{ij}\|_F, \\ \text{s.t. } & G = G^\top \succeq 0, \quad G_{ii} = I_n \text{ for } i = 1, \dots, N, \\ & \text{rank}(G) = n. \end{aligned}$$

This formulation only references G (not the \hat{R}_i 's). The LUD algorithm consists in relaxing (ignoring) the rank constraint. The resulting program is convex but is not an SDP because its cost function is not linear in G (it can be made linear for $q = 2$). LUD solves the convex program up to some precision using ADM (an alternating direction augmented Lagrangian method) (Wen *et al.*, 2010), adapted to this non-SDP scenario. ADM returns a matrix G which can be thought of as a denoised version of the measurement matrix $W_1 \in \mathbb{R}^{nN \times nN}$ which appears in the eigenvector method, see Section 5.4.1. Applying the eigenvector method to G then yields an estimator for the rotations $\hat{R}_1, \dots, \hat{R}_N$.

Remarkably, for a complete measurement graph with i.i.d. noise distributed following a perfect-or-outlier model ($\kappa = \infty, \kappa' = 0$ and some value of p), LUD recovers the rotations *exactly* as soon as p exceeds some threshold (0.46 for $n = 2$ and 0.49 for $n = 3$). Furthermore, in case the good measurements are not perfect but somewhat noisy ($0 < \kappa < \infty$), the recovery is stable in the sense that the estimation error is proportioned to $1/\kappa$. These same results also apply (appropriately modified) for incomplete measurement graphs when the available measurements are selected independently, uniformly at random, that is, following an Erdős-Rényi model.

5.5.3 Synthetic experiments

Figures 5.2–5.6 show the expected MSE reached by the MLE for varying noise parameters, in comparison with the CRB and the expected MSE reached by the initial guess alone. The expected MSE of the LUD algorithm (Wang & Singer, 2013) is also displayed, computed with code supplied by its authors. LUD does not have perfect knowledge of the noise model but still performs excellently.

All tests are performed for synchronization of rotations in \mathbb{R}^3 ($n = 3$) with one anchor ($A = \{1\}$) on complete measurement graphs with $N = 400$ nodes, i.i.d. noise, $\kappa' = 0$. Setting $\kappa' = 0$ means measurements are either complete outliers (w.p. $1-p$) or concentrated around the true relative rotation they measure with concentration κ (w.p. p).

The performance plots display an estimate of the expected MSE of estimators by averaging the MSE's obtained over a number of realizations of the noise. As a means to interpret the experiments, we point out that Langevin measurements with concentration $\kappa = 0.1, 1, 5$ and 10 are, on average, off by $123^\circ, 81^\circ, 30^\circ$ and 21° , resp. See also Figure 5.1. Likewise, it is useful to understand how good or bad an MSE level is. For $n = 3$, assuming the error is spread over all the rotations equally so that each rotation is off by an angle θ , then θ and the MSE are related by $\theta = \frac{180}{\pi} \sqrt{\frac{\text{MSE}}{2}}$ (in degrees). MSE's of $10^{-2}, 10^{-1}$ and 10^0 correspond to average errors of, respectively, $4^\circ, 13^\circ$ and 40° on each rotation.

Remark 5.3 (Generating random rotations). *To perform the tests presented in this section, random realizations of the noise are generated. A number of algorithms exist to generate pseudo-random rotation matrices from the uniform distribution (Chikuse, 2003, §2.5.1) (Diaconis & Shahshahani, 1987). Possibly one of the easiest methods to implement is the following $\mathcal{O}(n^3)$ algorithm, adapted from (Diaconis & Shahshahani, 1987, Method A, p. 22) with implementation details as cautioned in (Mezzadri, 2007) (for large n , see the former paper for algorithms with better complexity):*

1. Generate $A \in \mathbb{R}^{n \times n}$, such that the entries $A_{ij} \sim \mathcal{N}(0, 1)$ are i.i.d. normal random variables;
2. Obtain a QR decomposition of A : $QR = A$;
3. Set $Z := Q \text{diag}(\text{sign}(\text{diag}(R)))$ (this ensures the mapping $A \mapsto Z$ is well-defined; see (Mezzadri, 2007));
4. Z is now uniform on $O(n)$. If $\det(Z) = -1$, permute columns 1 and 2 of Z . Return Z .

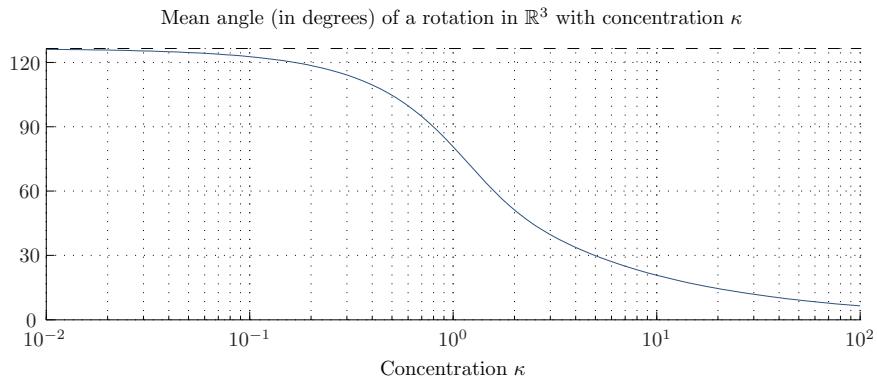


Figure 5.1: To understand the concentration parameter κ of the Langevin distribution (5.4), this plot shows the average error, in degrees, for a Langevin measurement of a rotation in $\text{SO}(3)$. More precisely, the curve has equation $\frac{180}{\pi} \mathbb{E} \{ \| \log(Z) \| / \sqrt{2} \}$, where the expectation is taken w.r.t. Z , distributed around the identity matrix with a Langevin of concentration κ . For $n = 2$ and $n = 3$, the quantity $\| \log(Z) \| / \sqrt{2}$ indeed corresponds to the angle θ (in radians) by which Z rotates around some axis. For example, at $\kappa = 5$, measurements are typically off by 30° .

Based on a uniform sampling algorithm on $\text{SO}(n)$, a simple acceptance-rejection scheme to sample from the Langevin distribution (Chikuse, 2003, §2.5.2) goes as follows:

1. Generate $Z \in \text{SO}(n)$, uniform;
2. Generate $t \in [0, e^{n\kappa}]$, uniform;
3. If $t \leq \exp(\kappa \text{trace}(Z))$, return Z (accept); Otherwise, try again (reject).

This is what we use. Not surprisingly, for large values of κ , this tends to be very inefficient. Chiuso et al. report using a Metropolis-Hastings-type algorithm instead (Chiuso et al., 2008, §7). Hoff describes an efficient Gibbs sampling method to sample from a more general family of distributions on the Stiefel manifold, which can be modified to work on $\text{SO}(n)$ (Hoff, 2009).

Figures 5.2–5.4 show nontrivial scenarios where the MLE rapidly reaches the CRB and solves the synchronization problem as well as possible, even at unfavorable SNR's. Figure 5.6 shows that for extremely low SNR's, the MLE may not reach the CRB. Even for such scenarios, most rotations are actually well estimated by the MLE: the error is mostly concentrated on

a few unlucky rotations. We describe an ad hoc method to detect these poorly estimated rotations now.

We propose a simple a posteriori criterion whose purpose is to rank the estimators \hat{R}_i from most likely to least likely to be accurate. This is heuristic and, admittedly, many approaches could be tested. For the sake of conciseness, we do not compare different ranking strategies here.

In the presence of many outliers, some rotations may be much harder to estimate than the others because too many of the measurements they are involved with happen to be outliers. We expect the maximum likelihood estimator to still be able to accurately estimate the other rotations. Comparatively, a least-squares based estimator such as the eigenvector method has a tendency to spread the error over all measurements, yielding overall poor synchronization in those cases.

For each measurement H_{ij} , compute a consistency score s_{ij} of agreement with the estimators \hat{R}_i and \hat{R}_j as follows, forming a symmetric matrix $S = (s_{ij})_{i,j=1\dots N}$:

$$s_{ij} = \ell_\kappa(\hat{R}_i^\top H_{ij} \hat{R}_j) = \frac{1}{c_n(\kappa)} \exp\left(\kappa \operatorname{trace}(\hat{R}_i^\top H_{ij} \hat{R}_j)\right). \quad (5.36)$$

Let $s_{ij} = 0$ if there is no measurement H_{ij} . Furthermore, let

$$D = \operatorname{diag}(d_1, \dots, d_N), \text{ with } d_i = \sum_{i \sim j} s_{ij}.$$

The summation is over the neighbors j of node i in the measurement graph. The consistency score of each estimator \hat{R}_i is then defined as s_i through a PageRank-like procedure:

$$s_i = \sum_{i \sim j} \frac{s_{ij}}{d_j} s_j, \quad (5.37)$$

that is, node i is given a large score if it is connected to nodes which have a large score themselves through measurements well explained by \hat{R} . By the Perron-Frobenius theorem, such scores exist, are positive and are uniquely defined if we further impose that they sum to N and require the measurement graph to be connected. Indeed, the vector s is simply the right eigenvector of the column-stochastic matrix SD^{-1} with eigenvalue 1. With proper normalization, it verifies $s = SD^{-1}s$.

As confirmed by Figure 5.7, a relatively low consistency score s_i indicates a higher chance that \hat{R}_i is a relatively bad estimator for the rotation R_i . The data in the latter figure comes from Figure 5.6, where it is seen that dropping a few of the worst \hat{R}_i 's, which may be acceptable in some applications, can decrease the MSE of the remaining estimators.

Finally, Figure 5.8 demonstrates a scenario where neither MLE nor LUD reach the CRB, but both seem to improve at the CRB rate.

	$\hat{\kappa}_0 = 20,$ $\hat{\kappa}'_0 = 0,$ $\hat{p}_0 = 1.00$	$\hat{\kappa}_0 = 20,$ $\hat{\kappa}'_0 = 0,$ $\hat{p}_0 = 0.50$	$\hat{\kappa}_0 = 20,$ $\hat{\kappa}'_0 = 5,$ $\hat{p}_0 = 1.00$
$\kappa = 3,$ $\kappa' = 0,$ $p = 1.00$	$\hat{\kappa} : (3.01, 0.04, 2.95, 3.12),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (1.00, 0.00, 1.00, 1.00)$	$\hat{\kappa} : (3.01, 0.04, 2.95, 3.12),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (1.00, 0.00, 1.00, 1.00)$	$\hat{\kappa} : (6.36, 8.90, 2.97, 54.01),$ $\hat{\kappa}' : (2.93, 0.17, 2.00, 3.07),$ $\hat{p} : (0.11, 0.18, 0.00, 0.93)$
$\kappa = 8,$ $\kappa' = 0,$ $p = 0.80$	$\hat{\kappa} : (7.98, 0.17, 7.61, 8.36),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (0.80, 0.01, 0.78, 0.82)$	$\hat{\kappa} : (7.98, 0.17, 7.61, 8.36),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (0.80, 0.01, 0.78, 0.82)$	$\hat{\kappa} : (8.00, 0.17, 7.61, 8.40),$ $\hat{\kappa}' : (0.02, 0.03, 0.00, 0.10),$ $\hat{p} : (0.80, 0.01, 0.77, 0.82)$
$\kappa = 8,$ $\kappa' = 0,$ $p = 0.40$	$\hat{\kappa} : (8.04, 0.22, 7.54, 8.50),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (0.40, 0.01, 0.36, 0.43)$	$\hat{\kappa} : (8.04, 0.22, 7.54, 8.50),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (0.40, 0.01, 0.36, 0.43)$	$\hat{\kappa} : (8.07, 0.22, 7.54, 8.50),$ $\hat{\kappa}' : (0.01, 0.02, 0.00, 0.08),$ $\hat{p} : (0.40, 0.01, 0.36, 0.43)$
$\kappa = 8,$ $\kappa' = 2,$ $p = 0.60$	$\hat{\kappa} : (4.20, 0.13, 3.92, 4.48),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (0.96, 0.01, 0.95, 0.97)$	$\hat{\kappa} : (4.20, 0.13, 3.92, 4.48),$ $\hat{\kappa}' : (0.00, 0.00, 0.00, 0.00),$ $\hat{p} : (0.96, 0.01, 0.95, 0.97)$	$\hat{\kappa} : (7.94, 0.38, 7.20, 8.87),$ $\hat{\kappa}' : (2.01, 0.09, 1.82, 2.20),$ $\hat{p} : (0.60, 0.03, 0.53, 0.66)$

Table 5.1: Estimation of the parameters of a mixture of Langevin on $\text{SO}(3)$. The parameters of the mixture are stated in the left column. The top row indicates different values for the initial guess used in optimizing g (5.33). For each mixture (row), we sample 2 000 rotations from the mixture and record the estimated $\hat{\kappa}, \hat{\kappa}', \hat{p}$ reached using each initial guess (column). This is repeated 50 times and each cell reports, in a tuple, the mean value reached, the standard deviation, the smallest and the largest observed value. The two first columns are identical, which shows some robustness w.r.t. the initial guess. The estimation is excellent, except for the last row, where κ' is nonzero. In that scenario, only the third initial guess succeeds, suggesting that to estimate a nonzero κ' the initial guess for the latter needs to be nonzero itself. Regarding the top-right cell, note that the models ($\kappa = 3, p = 1.00, \kappa'$ arbitrary) and ($\kappa' = 3, p = 0.00, \kappa$ arbitrary) are equivalent: the estimation mostly succeeds there too.

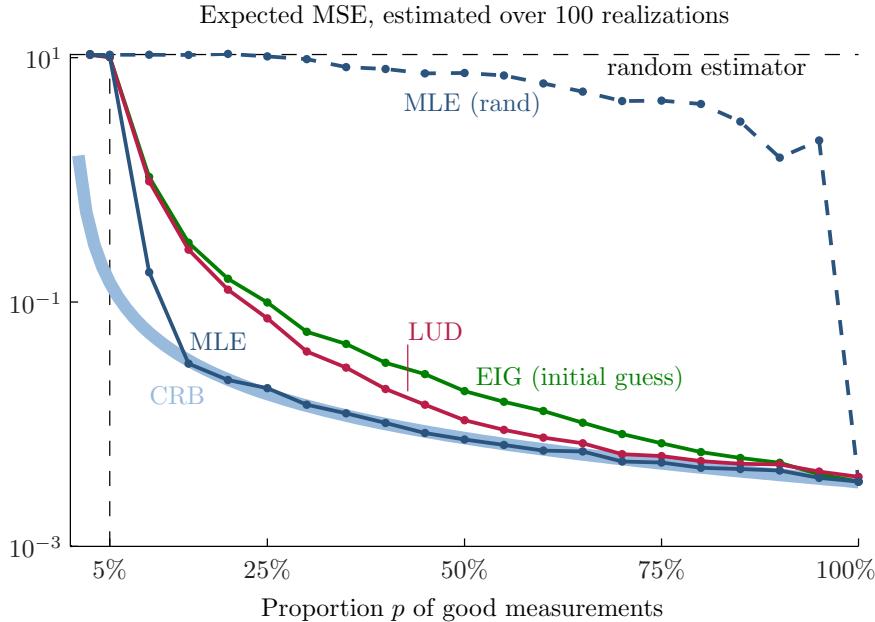


Figure 5.2: Synchronization of a complete graph of $N = 400$ rotations in $\text{SO}(3)$ with a variable proportion p of good measurements (concentration $\kappa = 5$). The remaining measurements are complete outliers ($\kappa' = 0$). As predicted in Section 5.3, a phase transition occurs for the eigenvector method when $\beta = 0.9p = 1/\sqrt{N}$, that is, at $p = 5.6\%$. For smaller p , the eigenvector method (and actually all estimators observed) perform as badly as a random estimator. For larger p , the MLE rapidly reaches the CRB and appears to be efficient. The initial guess, based on the eigenvector method, is much improved by the MLE at low SNR. The curve *MLE (rand)* uses a random initial estimator instead of the eigenvector method and refines this estimator with the MLE optimization approach. The results clearly demonstrate the importance of picking a good initial iterate. LUD is the method proposed in (Wang & Singer, 2013): it has no knowledge of the noise model but still performs well. The initial guess is computed in 4 to 6 seconds. The MLE needs 2 to 20 additional seconds for p larger than 15%. For smaller p (corresponding to harder problems), the MLE may need 4 to 6 minutes to converge.

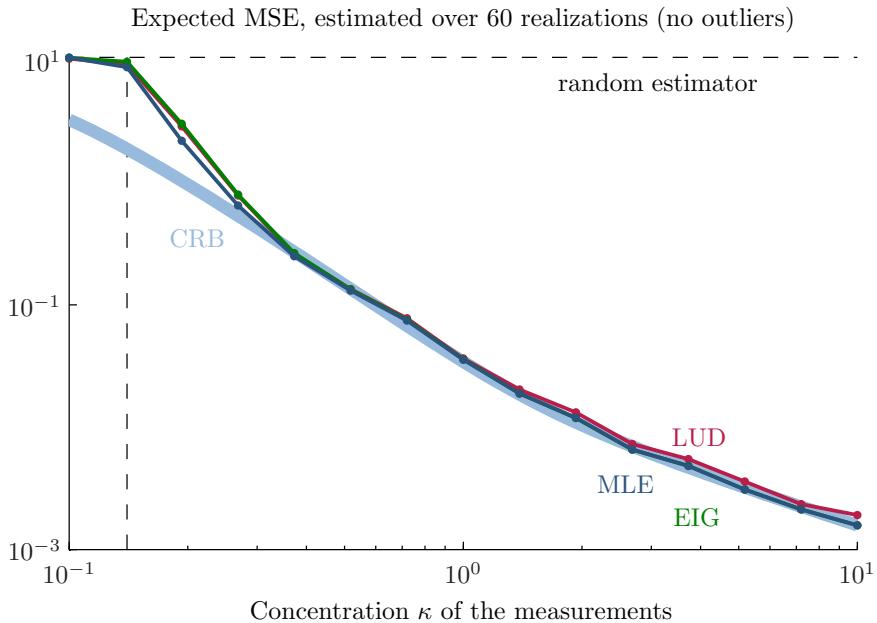


Figure 5.3: Synchronization of a complete graph of $N = 400$ rotations in $\text{SO}(3)$ without outliers ($p = 100\%$). The measurements are distributed following a Langevin with variable concentration κ . All estimators seem to rapidly reach the CRB as the SNR increases. The vertical dashed line indicates the predicted point at which the eigenvector method starts performing better than a random estimator (Section 5.3).

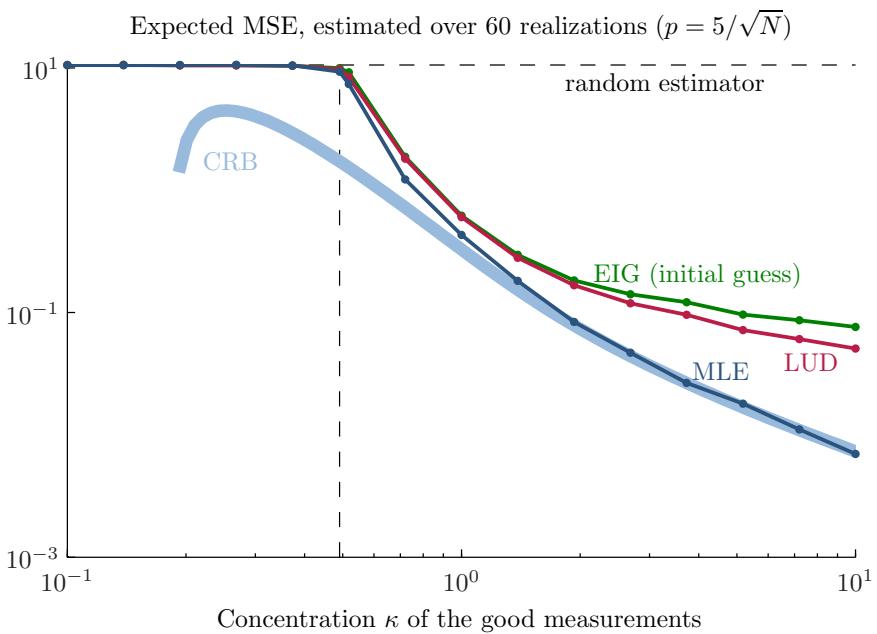


Figure 5.4: Same experiment as Figure 5.3, this time with outliers. A (comfortable) $p = 25\%$ of the measurements have variable concentration κ . The remaining 75% are complete outliers ($\kappa' = 0$). The proposed maximum likelihood estimator seems to rapidly reach the CRB as the SNR increases.

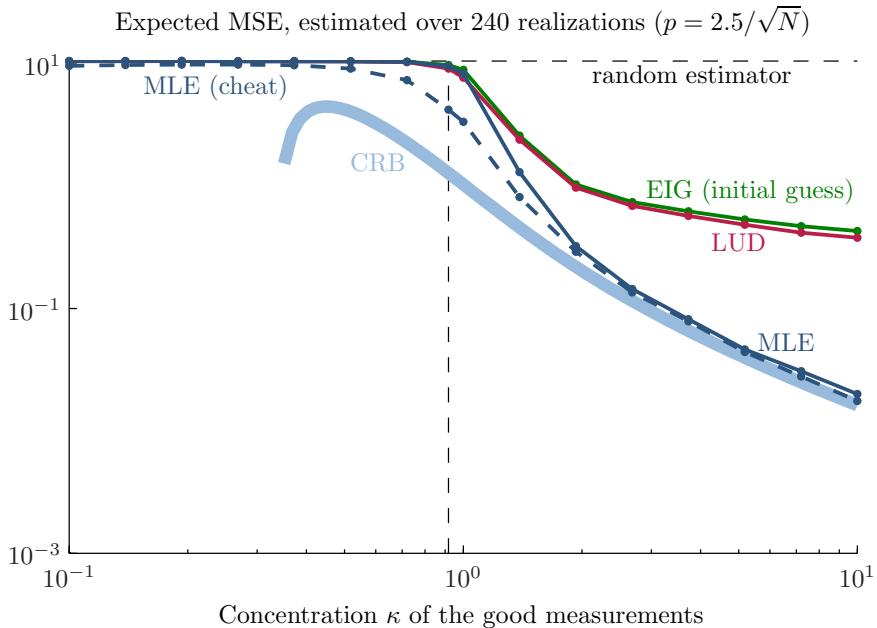


Figure 5.5: Same experiment as Figure 5.4, with $p = 2.5/\sqrt{N} = 12.5\%$ inliers. The *MLE (cheat)* dashed curve shows the expected MSE reached when using the true rotations as initial guess for the optimization stage. The comparison suggests that, at this noise level, the legitimate MLE method is still able to converge to optimizers of good quality.

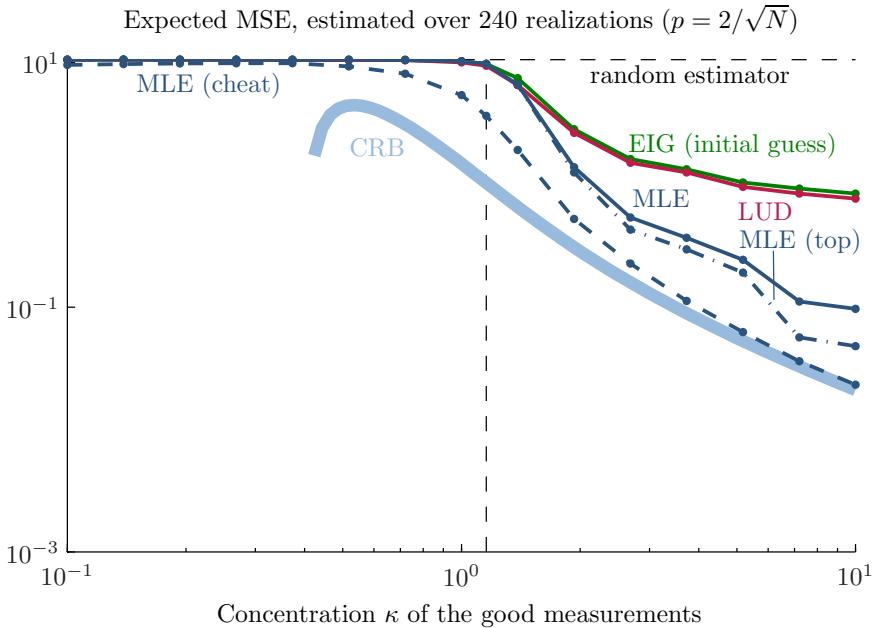


Figure 5.6: Same experiment as Figure 5.4, with a challenging fraction of outliers: $p = 10\%$ and the remaining 90% of the measurements bear no information. In this extreme scenario, the computed MLE does not seem to reach the CRB. This is in part due to non-global optimization of the likelihood function. Indeed, experimentally, for κ larger than the critical value (dashed vertical line), the *MLE (cheat)* algorithm (dashed curve, see Figure 5.5) reaches better critical points (according to L_A) than the legitimate MLE algorithm more than 9 out of 10 times, and indeed performs better. This suggests that the lesser performance of our proxy for the MLE is due to local optimizer traps. The *MLE (top)* curve (dash-dot) displays the MSE reached by the 395 best estimated rotations according to the score (5.37).

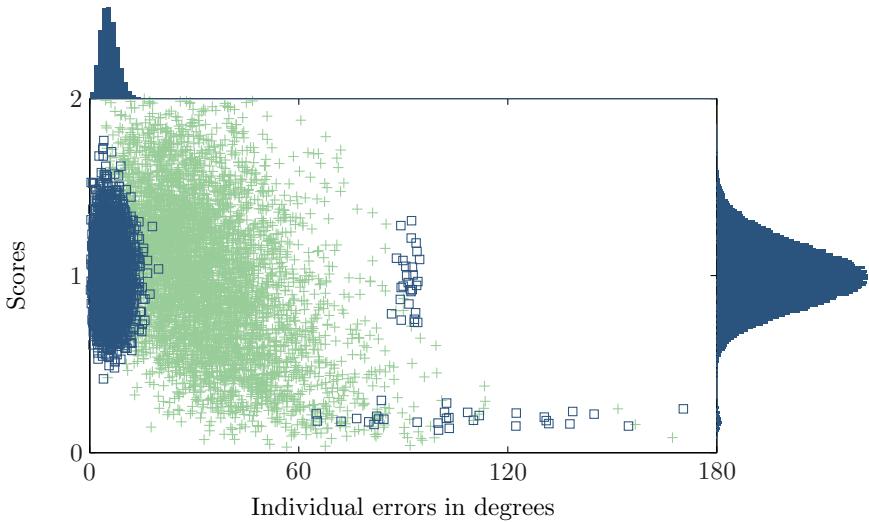


Figure 5.7: For each of the 240 repetitions of the experiment in Figure 5.6 at the largest concentration value ($\kappa = 10$), both for the eigenvector method (green '+') and for the MLE method (blue \square 's), we compute the 399 individual estimation errors $\|\log(R_i^\top \hat{R}_i)\|_F$ and plot them in degrees against the scores s_i (5.37) the corresponding \hat{R}_i 's obtained in that repetition. All $240 \times 399 = 95\,760$ points are used to produce the marginal distributions on the sides of the plot (only for MLE), but only 5% of the points are actually plotted, for legibility. Observe how, for the MLE method, (i) the error tends to be concentrated on just a few rotations, and (ii) the score is an excellent predictor to identify those poorly estimated rotations. The blue \square 's in the middle (large score even though large error) correspond to a repetition where the unique anchor was connected through too few good measurements, resulting in overall large absolute errors despite small relative errors.

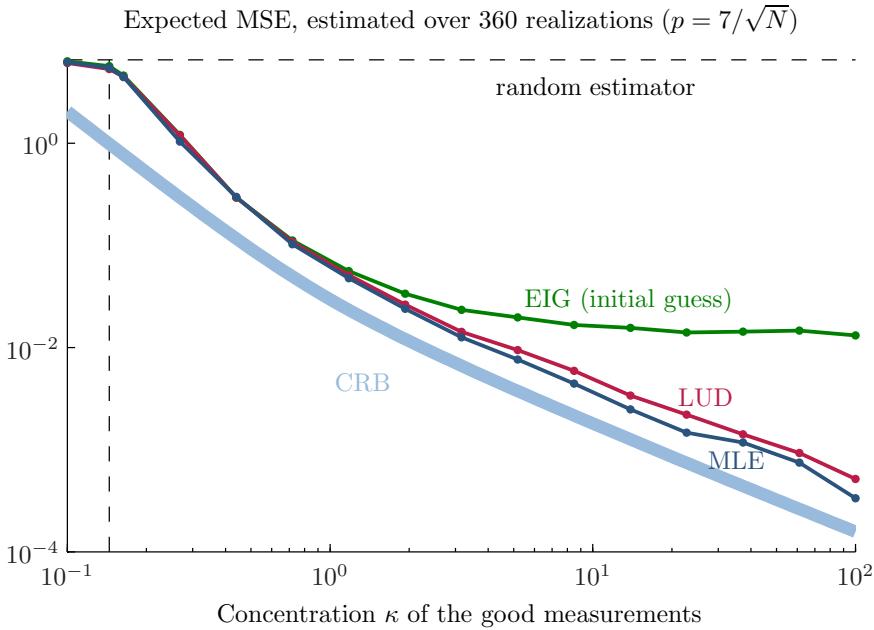


Figure 5.8: Synchronization of a complete graph of $N = 100$ rotations in $\text{SO}(2)$ (notice the $n = 2$ instead of 3 in the other figures) with 30% outliers ($p = 70\%$). The measurements are distributed following a Langevin with variable concentration κ . (This experiment is a rerun of the setup in (Wang & Singer, 2013, Fig. 8.3)) As the accuracy of the good measurements increases (larger κ), the MSE of the eigenvector method levels off. Interestingly, although neither the MLE nor the LUD method reach the CRB, their MSE decreases at about the same rate as the CRB. This is especially remarkable for LUD, which has no knowledge of the noise distribution. Unfortunately, LUD is also slower to compute than MLE (by a factor of 10 to 100).

5.6 Application: 3D scan registration

Following the experimental setup in (Tzveneva *et al.*, 2011; Wang & Singer, 2013), the three synchronization methods discussed (EIG, LUD and MLE+) are presented with data from an idealized 3D scan registration problem. The scans composing the Lucy statue 3D model (Figure 5.9) are downloaded from the Stanford 3D Scanning repository.³ We extract $N = 368$ of these scans which cover most of the model, totaling 3.5 million triangles out of 116 million. As noted on the repository’s webpage, this experimental setup is strongly idealized compared to true 3D scanning tasks. The Lucy dataset is heavily processed before it reaches the experiments in this section, hence the results should be taken with a grain of salt. Nevertheless, the noise affecting the measurements is largely out of our control, which gives some credit to the engaging performance of MLE+ reported below.



Figure 5.9: Left: virtual representation of Lucy provided by the Stanford 3D Scanning Repository. Right: representation of the subset of 368 scans of Lucy (each in a different color) with their reference alignment, using trimesh2.

Each scan is represented in its own reference frame. Let $(R_i, t_i) \in \text{SO}(3) \times \mathbb{R}^3$ represent the transformation from the local reference frame of scan i to the global reference frame, such that a point $p \in \mathbb{R}^3$ is transformed from the local to the global frame via $p \mapsto R_i p + t_i$. If two scans i and j contain two points $p_{i,k}$ and $p_{j,\ell}$ which correspond to the same physical point, then the following equation should hold, up to noise terms:

$$R_i p_{i,k} + t_i = R_j p_{j,\ell} + t_j.$$

³<http://graphics.stanford.edu/data/3Dscanrep/>



Figure 5.10: Alignments of the 368 patches of Lucy. Left to right: rotations are synchronized with EIG, LUD and MLE+. The fourth image depicts the reference alignment.

Thus, the rigid transformation of a point in frame i to a corresponding point in frame j is given by

$$p_{j,\ell} = R_j^\top R_i p_{i,k} + R_j^\top(t_i - t_j).$$

The iterative closest point (ICP) algorithm (Rusinkiewicz & Levoy, 2001), as implemented in the trimesh2 library,⁴ is applied to all 67 528 pairs of scans to produce estimates of the relative rigid transformation $(R_j^\top R_i, R_j^\top(t_i - t_j))$. Based on an initial guess of the relative alignment of two scans, ICP proceeds by matching their points according to a nearest neighbor criterion. Based on these matches, the scans are optimally aligned (this is a classical orthogonal Procrustes problem, solved by SVD). The new alignment is used to produce a new matching of the points and the procedure is iterated. If ICP finds sufficient overlap between the patches, it outputs a relative rigid transformation measurement. The expectation is that a correct overlap detection yields a good quality measurement whereas a false overlap detection yields an essentially random outlier, justifying the mixture of Langevin model. ICP identified 2 006 overlapping scans for Lucy.

The relative rotation measurements are readily processed by any of the synchronization algorithms discussed to obtain $\hat{\mathbf{R}}$. We then further synchronize the translational alignments based on the measurements

$$t_{ij} \approx R_j^\top(t_i - t_j) \approx \hat{R}_j^\top(t_i - t_j).$$

⁴Trimesh2 by S. Rusinkiewicz, see <http://gfx.cs.princeton.edu/proj/trimesh2/>.

A simple least-squares procedure to compute estimates $\hat{t}_1, \dots, \hat{t}_N$ consists in solving the following minimization problem:

$$\min_{\hat{t}_1, \dots, \hat{t}_N} \sum_{i \sim j} s_{ij} \|\hat{t}_i - \hat{R}_j t_{ij}\|^2.$$

The weights are set according to (5.36). The rationale is that if the rotation measurement between scans i and j appears to be poor (according to our current estimator $\hat{\mathbf{R}}$), then the translation measurement is probably poor too. And indeed, setting uniform weights in this step would lead to poor results in the sequel. Ordering the measurement edges arbitrarily, build the matrix $T_{\text{meas}} \in \mathbb{R}^{3 \times M}$ such that the columns of T_{meas} correspond to the rotated measurements $\hat{R}_j t_{ij}$, in order. Let $K \in \mathbb{R}^{N \times M}$ such that the column corresponding to edge (i, j) is zero except for a 1 on row i and a -1 on row j . Further let $S \in \mathbb{R}^{M \times M}$ be a diagonal matrix with diagonal entries s_{ij} , in order. Then, for $\hat{T} \in \mathbb{R}^{n \times N}$ a matrix whose columns are the estimated translations \hat{t}_i , the alignment problem reads

$$\min_{\hat{T} \in \mathbb{R}^{n \times N}} \|(\hat{T}K - T_{\text{meas}})S^{1/2}\|_F^2.$$

The solution follows easily:

$$\hat{T} = T_{\text{meas}}SK^\top(KSK^\top)^\dagger. \quad (5.38)$$

Notice that SK^\top is the Laplacian of the measurement graph, weighted by s_{ij} . The latter graph is theoretically connected, hence the above formulation makes \hat{T} the unique optimal solution centered at the origin (to check, multiply by $\mathbf{1}_N$ on both sides). Numerically though, when $\hat{\mathbf{R}}$ is obtained from MLE+, the graph has two isolated nodes (147 and 293). These are scans for which all measurements are trusted with s_{ij} close to zero (they have a score (5.37) of 10^{-17} compared to a median score of 0.8), so that we omit them in the plots. By default, the formulation above centers isolated scans at the origin.

We proceed as follows to generate Figure 5.10: for each of EIG, LUD and MLE+, obtain an estimate $\hat{\mathbf{R}}$ of the rotations. From this estimate, compute T_{meas} and S . Use these with (5.38) to compute \hat{T} , an estimate of the translations. Then apply the computed rigid transformations to the scans and render them using trimesh2. From the figure, it is clear that MLE+ attains the best reconstruction. Table 5.2 collects some statistics regarding the performance of all algorithms. Table 5.3 collects more statistics regarding the iterations of MLE+.

In Tables 5.2 and 5.3, the median data error (in degrees) is the median value of

$$\frac{180}{\pi} \sqrt{2} \text{dist}(H_{ij}, \hat{R}_i \hat{R}_j^\top).$$

Method	Total time (s)	median data error (deg)	median synch. error (deg)	MSE
EIG	0.6	5.37	9.99	$3.898 \cdot 10^{-2}$
LUD	2400	1.51	2.93	$8.251 \cdot 10^{-3}$
MLE+	27.6	0.81	1.70	$4.698 \cdot 10^{-3}$

Table 5.2: Performance metrics for EIG, LUD and MLE+ estimating rotations on the Lucy dataset. The large running time of LUD (40 minutes for almost 2000 iterations) is possibly due to its implementation not taking full advantage of the sparsity of the measurement graph (only 3% of the edges are present).

The mean squared error (MSE), in the context of this section, is given by

$$\min_{Q \in \text{SO}(n)} \sum_{i=1}^N \|R_i - \hat{R}_i Q\|_{\text{F}}^2 / N,$$

where \mathbf{R} are the reference rotations considered as ground truth. This is the metric used in (Singer, 2011; Tzveneva *et al.*, 2011; Wang & Singer, 2013). The rotation Q optimally aligns \mathbf{R} with $\hat{\mathbf{R}}$ in this sense. It is given by (5.25):

$$Q = \Pi_{\text{SO}(n)} \left(\sum_{i=1}^N \hat{R}_i^\top R_i \right).$$

The median synchronization error (in degrees) is the median value of

$$\frac{180}{\pi} \sqrt{2} \text{dist}(R_i, \hat{R}_i Q). \quad (5.39)$$

We note that Tzveneva *et al.* (2011) reach a reconstruction of almost the same quality as MLE+ using a simple outlier rejection iteration on top of the eigenvector method: from the estimation of the rotations, the quality of the measurements is assessed; based on this assessment, some measurements are discarded according to a user-supplied criterion and the procedure is iterated. In contrast, an advantage of MLE+ is that it never completely discards measurements and the distinction between inlier and outlier, which is not binary anymore, is automatically determined from the data. A possible combination of both ideas to accelerate MLE+ would be to replace the MLE estimation with an eigenvector estimation based on the weights s_{ij} (5.36).

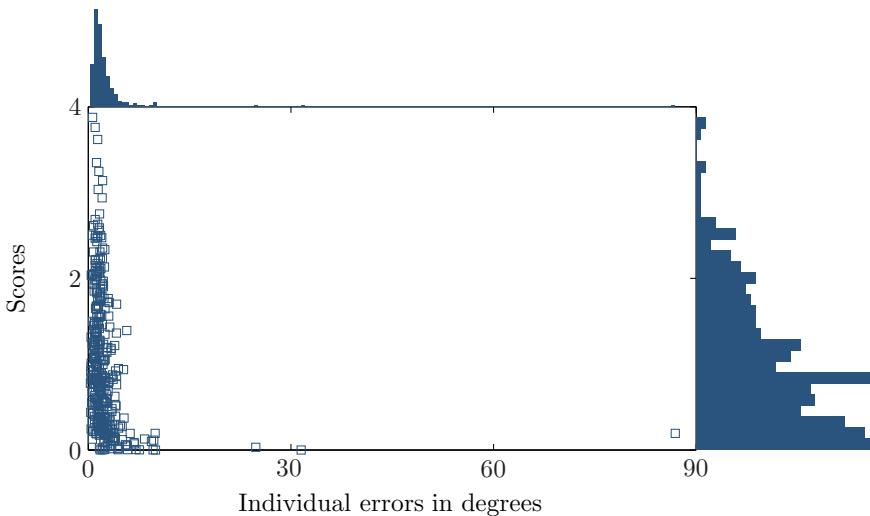


Figure 5.11: Scores (5.37) versus individual synchronization errors (5.39) for the MLE+ estimator of the rotations on the Lucy dataset. Notice the concentration of error: only a few scans are badly aligned.

5.7 Conclusions

This chapter framed synchronization of rotations as an estimation problem on a manifold. The maximum likelihood approach made it clear how tools from optimization on manifolds can be leveraged to perform the estimation. Many smooth Riemannian optimization algorithms guarantee convergence toward critical points, but in general there is no guarantee of global optimality. This called for two actions. First, a known spectral relaxation of the problem is used as initial guess, for which a known analysis establishes that it performs well even in the face of outliers. In practice, a good initial guess enhances the chance to converge toward a good local optimum. Second, it is necessary to develop tools to assess the quality of the computed estimator. To this end, we develop in Chapter 8 CRB's for the present estimation problem and saw in the synthetic experiments that the proposed estimator, MLE, appears to be efficient, at least asymptotically. Furthermore, we proposed one particular way of estimating the parameters of the noise model, leading to MLE+. The latter algorithm should prove useful for practical problems too, as indicated by the experiments on the Lucy dataset.

Code for both the estimators and the CRB's is available on my web page, currently hosted at <http://perso.uclouvain.be/nicolas.boumal/>.

iter	$\hat{\kappa}$	$\hat{\kappa}'$	\hat{p}	synch. time (s)	median data error (deg)	median synch. error (deg)	MSE	fitting time (s)
0				0.6	5.37	9.99	$3.898 \cdot 10^{-2}$	
1	7	0	1.00	1.1	6.04	12.60	$5.542 \cdot 10^{-2}$	0.2
2	249	0	0.93	2.3	1.21	1.97	$4.863 \cdot 10^{-3}$	0.1
3	5139	0	0.91	2.0	0.87	1.62	$4.281 \cdot 10^{-3}$	0.1
4	10025	0	0.90	2.1	0.83	1.72	$4.350 \cdot 10^{-3}$	0.1
5	11710	0	0.89	1.6	0.81	1.73	$4.356 \cdot 10^{-3}$	0.1
6	12425	0	0.89	1.9	0.81	1.71	$4.354 \cdot 10^{-3}$	0.0
7	12526	0	0.89	11.3	0.81	1.70	$4.712 \cdot 10^{-3}$	0.1
8	12593	0	0.89	3.9	0.81	1.70	$4.698 \cdot 10^{-3}$	0.1

Table 5.3: Progress of MLE+ synchronizing rotations on the Lucy dataset. The first row reports timing and error metrics for the initial guess (eigenvector method). After the eigenvector method, we manually set values for the parameters $\hat{\kappa}, \hat{\kappa}', \hat{p}$, run synchronization with MLE and fit a new mixture to the residuals. The parameters of this new mixture are used on the next row. The method appears to converge and suggests that 89% of the measurements are very accurate while 11% are outliers.

Part II

Estimation bounds

Chapter 6

Estimation on manifolds

In this part of the thesis, we study estimation problems on Riemannian manifolds. In such problems, one would like to estimate a deterministic but unknown parameter θ belonging to a manifold \mathcal{P} , given a measurement y belonging to a probability space \mathcal{M} . The measurement y is a random variable whose probability density function is shaped by θ . It is because y is distributed differently for different θ 's that sampling (observing) y reveals information about θ . In particular, we focus on developing Cramér-Rao bounds (CRB's), that is, lower bounds on the variance of estimators for certain tasks.

Estimation problems on manifolds arise naturally in camera network pose estimation (Tron & Vidal, 2009), angular synchronization (Singer, 2011), covariance matrix estimation and subspace estimation (Smith, 2005), the generalized Procrustes problem (Chaudhury *et al.*, 2013), Wahba's problem (Markley, 1988) and many other applications, see references therein. CRB's relate the covariance matrix of estimators to the Fisher information matrix (FIM) of an estimation problem through matrix inequalities. The classical results deal with estimation on Euclidean spaces (Rao, 1945). More recently, a number of authors have established similar bounds in the manifold setting, see (Smith, 2005; Xavier & Barroso, 2005) and the many references therein.

This chapter covers the main results established in (Smith, 2005) regarding intrinsic CRB's, with a focus on unbiased estimators. This chapter serves both as an introduction to intrinsic estimation theory and as a reference point for the next chapter, which hosts a useful adaptation of the CRB's for the special case of Riemannian submanifolds and Riemannian quotient manifolds. The contents of this chapter are attributable to Smith (2005), our contribution being a (somewhat) original exposition.

The CRB's presented in this chapter hold *at large SNR*. The origin of this

provision is double. Firstly, the definition of covariance on a manifold uses the logarithmic map on that manifold, which is only locally well-defined. It is thus necessary to require the noise level to be low enough so that the estimator of a parameter θ will, almost surely, belong to a neighborhood of θ where the logarithm is well-defined. This can be relevant even on flat manifolds such as the circle $\text{SO}(2)$ for example, which is compact. Secondly, on curved manifolds, the proof of Theorem 6.4 relies on truncated Taylor expansions. Those are legitimate only at large enough SNR so that typical errors are small compared to the scale at which curvature becomes a dominant feature.

6.1 Fisher information, bias and covariance

We consider the problem of estimating a (deterministic) parameter θ based on a measurement y . The parameter belongs to the *parameter space* \mathcal{P} , a Riemannian manifold. The measurement is a realization of a random variable Y defined over a probability space \mathcal{M} , the *measurement space*. Notice the somewhat different notation from usual: \mathcal{M} need not be a manifold. \mathcal{P} is equipped with a Riemannian metric $\langle \cdot, \cdot \rangle$ and a Riemannian connection ∇ (Theorem 2.2). \mathcal{M} is equipped with a probability measure μ such that $\mu(\mathcal{M}) = 1$.

Naturally, we need the realization y to convey information about θ . This is the case if and only if the distribution of Y is conditioned on θ :

$$Y \sim f(\cdot; \theta). \quad (6.1)$$

Sampling from Y reveals information about the distribution of Y . The more this distribution depends on θ , the more sampling from Y reveals about θ . This is the intuition we set out to quantify.

Assume the parameter space has dimension $\dim \mathcal{M} = d$ and let

$$e = (e_1, \dots, e_d)$$

be an orthonormal basis of the tangent space $T_\theta \mathcal{P}$ with respect to the Riemannian metric. The results derived in this section are intrinsic: they do not depend on the choice of basis e . Nevertheless, working “in coordinates” simplifies much of the algebra, warranting a sidestep from perfectly intrinsic notations. The intuition laid out above suggests that the information about θ in y is linked to how much the probability density function (pdf) of Y changes when θ changes. This motivates the following definitions:

Definition 6.1 (Log-likelihood function). *The log-likelihood function L is a random function over the parameter space defined by*

$$L: \mathcal{P} \rightarrow \mathbb{R}: \theta \mapsto L(\theta) = \log f(Y; \theta). \quad (6.2)$$

Definition 6.2 (Score). *The score vector $s = s(\theta) \in \mathbb{R}^d$ is a random coordinate vector defined w.r.t. the orthonormal basis e as*

$$s_i = DL(\theta)[e_i]. \quad (6.3)$$

The relevance of the log will become clear in the derivations.

We are especially interested in the amount of information an observation y reveals on average. This prompts us to take expectations with respect to Y —all expectations in this chapter are with respect to Y , the only source of randomness in the present setting.

The score vector has zero mean. The covariance of the score vector is an important quantity for our purpose, known as the *Fisher information matrix* (FIM).

Lemma 6.1. *The score vector has zero mean: $\mathbb{E}\{s\} = 0$.*

Proof. For each $i \in \{1, \dots, d\}$, with D_θ denoting the directional derivative w.r.t. θ , the expectation of s_i is given by:

$$\begin{aligned} \mathbb{E}\{s_i\} &= \mathbb{E}\{DL(\theta)[e_i]\} \\ &= \int_{\mathcal{M}} D_\theta \log f(y; \theta)[e_i] f(y; \theta) d\mu(y) \\ &= \int_{\mathcal{M}} D_\theta f(y; \theta)[e_i] d\mu(y) \\ &= D_\theta \left(\theta \mapsto \int_{\mathcal{M}} f(y; \theta) d\mu(y) \right) (\theta)[e_i] \\ &= D_\theta (\theta \mapsto 1)(\theta)[e_i] = 0. \end{aligned}$$

We commuted integration over \mathcal{M} with a derivative w.r.t. θ , which requires f to meet mild regularity conditions. \square

Definition 6.3 (Fisher information matrix (FIM)). *The Fisher information matrix $F = F(\theta)$ is the symmetric, positive semidefinite matrix of size d defined w.r.t. the basis e as*

$$F = \mathbb{E}\{ss^\top\}.$$

Thus, the entries of F are given by:

$$F_{ij} = F(\theta)_{ij} = \mathbb{E}\{DL(\theta)[e_i] \cdot DL(\theta)[e_j]\}.$$

When F is everywhere positive definite, it defines a Riemannian metric in its own right. The study of Riemannian manifolds equipped with the Fisher information metric is called *information geometry* (Amari & Nagaoka, 2007). This is not our focus.

The goal is to estimate θ , leading to the definition of estimators.

Definition 6.4 (Estimator). *An estimator $\hat{\theta}: \mathcal{M} \rightarrow \mathcal{P}$ is a deterministic mapping which to each realization y of the measurement associates a parameter $\hat{\theta}(y)$.*

The *estimation error* is classically defined to be the random variable $\hat{\theta}(Y) - \theta$. The difference between two points θ and $\hat{\theta}$ on a manifold is not defined intrinsically though. Remember from Section 2.6 that the logarithmic map (or inverse exponential map) is a good replacement for the difference between two points on a manifold:

$$X_\theta = \text{Log}_\theta(\hat{\theta}).$$

Thus, X is a random tangent vector field. For each realization y of the measurement and each value of the parameter θ , it generates $X_\theta(y)$, a tangent vector at θ . This vector “points toward” $\hat{\theta}$ and its length coincides with the geodesic distance $\text{dist}(\theta, \hat{\theta})$. In coordinates, we write

$$x_i = x(\theta)_i = \left\langle \text{Log}_\theta(\hat{\theta}), e_i \right\rangle_\theta. \quad (6.4)$$

Notice that the norm of x is the magnitude of the estimation error:

$$\|x\| = \sqrt{x^\top x} = \|\text{Log}_\theta(\hat{\theta})\|_\theta = \text{dist}(\theta, \hat{\theta}).$$

Definition 6.5 (Bias). *In coordinates w.r.t. the basis e , the bias of an estimator for a given parameter value $\theta \in \mathcal{P}$ is the average error vector $b = b(\theta) \in \mathbb{R}^d$:*

$$b = b(\theta) = \mathbb{E}\{x\}.$$

Thus, b_i quantifies the bias of $\hat{\theta}$ along the direction e_i :

$$b_i = \mathbb{E}\left\{\left\langle \text{Log}_\theta(\hat{\theta}), e_i \right\rangle_\theta\right\}.$$

An estimator is unbiased if its bias vector is zero everywhere on \mathcal{P} :

$$\forall \theta \in \mathcal{P}, \quad b(\theta) = 0.$$

We restrict our analysis to unbiased estimators. See the reference paper of this chapter for a treatment of biased estimators (Smith, 2005). The following definition quantifies the covariance of the estimation error.

Definition 6.6 (Covariance). *For an unbiased estimator $\hat{\theta}$, the covariance matrix $C = C(\theta) \in \mathbb{R}^{d \times d}$ w.r.t. the basis e of $T_\theta \mathcal{P}$ is a symmetric, positive semidefinite matrix defined by*

$$C = C(\theta) = \mathbb{E}\{xx^\top\}.$$

Thus, the entries of C are given by:

$$C_{ij} = C(\theta)_{ij} = \mathbb{E}\{x_i x_j\} = \mathbb{E}\left\{\left\langle \text{Log}_\theta(\hat{\theta}), e_i \right\rangle_\theta \cdot \left\langle \text{Log}_\theta(\hat{\theta}), e_j \right\rangle_\theta\right\}.$$

In particular, the variance of $\hat{\theta}$ at θ is

$$\text{trace}(C(\theta)) = \mathbb{E}\{x^\top x\} = \mathbb{E}\left\{\|\text{Log}_\theta(\hat{\theta})\|_\theta^2\right\} = \mathbb{E}\left\{\text{dist}^2(\theta, \hat{\theta})\right\},$$

with dist denoting the Riemannian distance on \mathcal{P} w.r.t. the chosen Riemannian metric.

6.2 Intrinsic Cramér-Rao bounds

In the previous section, we defined the covariance C of an estimator, which quantifies the average estimation error of that estimator, and the Fisher information matrix F of an estimation problem, which quantifies the average amount of information the random measurement y reveals about the sought parameter $\theta \in \mathcal{P}$. Necessarily, the smaller F is, the larger C has to be. This section quantifies that relationship in the form of a matrix inequality. For \mathcal{P} a Euclidean space, we will recover the celebrated result $C \succeq F^{-1}$. For \mathcal{P} a Riemannian manifold, additional terms are in order, related to the possible curvature of \mathcal{P} .

We first establish the following two lemmas about the cross-correlation between the score s and the estimation error x .

Lemma 6.2. *For all $u, v \in \mathbb{R}^d$ and for any tangent vector fields U, V such that $U_\theta = \sum_{i=1}^d u_i e_i$ and $V_\theta = \sum_{i=1}^d v_i e_i$,*

$$u^\top \mathbb{E}\{sx^\top\} v = -\mathbb{E}\{\langle (\nabla_U X)_\theta, V_\theta \rangle_\theta\}, \quad (6.5)$$

where $X_\theta = \text{Log}_\theta(\hat{\theta})$ defines the random error tangent vector field X and ∇ is the Riemannian connection on \mathcal{P} .

Proof. The no-bias assumption reads, for all $\theta \in \mathcal{P}$,

$$\mathbb{E}\{X_\theta\} = \int_{\mathcal{M}} X_\theta(y) f(y; \theta) d\mu(y) = 0 \in T_\theta \mathcal{P},$$

where $X_\theta(y) = \text{Log}_\theta(\hat{\theta}(y))$. This defines a zero vector field on \mathcal{P} . Taking covariant derivatives with respect to U on both sides of the equation, then taking inner products with V at θ on both sides too yields the scalar equation:

$$\int_{\mathcal{M}} \langle (\nabla_U(fX))_\theta, V_\theta \rangle_\theta d\mu(y) = 0.$$

Notice that the covariant derivative ∇_U commutes with the integral over \mathcal{M} . Apply the product rule for affine connections:

$$\int_{\mathcal{M}} \langle D_{\theta} f(y; \theta) [U_{\theta}] \cdot X_{\theta} + f(y; \theta) (\nabla_U X)_{\theta}, V_{\theta} \rangle_{\theta} d\mu(y) = 0,$$

where D_{θ} denotes a directional derivative w.r.t. θ . The following holds:

$$D_{\theta} \log f(y; \theta) [U_{\theta}] = \frac{1}{f(y; \theta)} D_{\theta} f(y; \theta) [U_{\theta}].$$

Inject the latter into the previous equation to obtain, with $L(\theta) = \log f(y; \theta)$:

$$\int_{\mathcal{M}} \left(DL(\theta) [U_{\theta}] \cdot \langle X_{\theta}, V_{\theta} \rangle_{\theta} + \langle (\nabla_U X)_{\theta}, V_{\theta} \rangle_{\theta} \right) f(y; \theta) d\mu(y) = 0.$$

From equations (6.3) and (6.4), we obtain respectively $DL(\theta) [U_{\theta}] = u^T s$ and $\langle X_{\theta}, V_{\theta} \rangle_{\theta} = x^T v$. In expectation notation:

$$u^T \mathbb{E} \{ s x^T \} v = -\mathbb{E} \{ \langle (\nabla_U X)_{\theta}, V_{\theta} \rangle_{\theta} \},$$

which concludes the proof. \square

The right hand side of equation (6.5) deserves a closer inspection. It involves infinitesimally thin geodesic triangles. Figure 6.1 depicts a regular triangle in a Euclidean space compared to a geodesic triangle on a curved manifold. To begin, assume that the parameter space \mathcal{P} is a Euclidean space. Then, the error vector is simply $X_{\theta} = \hat{\theta} - \theta$. Furthermore, the covariant derivative reduces to the classical directional derivative, so that

$$(\nabla_U X)_{\theta} = D(\theta \mapsto \hat{\theta} - \theta)(\theta) [U_{\theta}] = -U_{\theta}.$$

Hence, $u^T \mathbb{E} \{ s x^T \} v = u^T v$ for all $u, v \in \mathbb{R}^d$. In conclusion, for \mathcal{P} a Euclidean space, $\mathbb{E} \{ s x^T \} = I$ is the identity matrix. As Figure 6.1 suggests, this is no longer the case for a curved manifold \mathcal{P} . Lemma 6.3 quantifies the effects of curvature. Refer to Section 2.8 for a brief introduction to curvature.

Lemma 6.3. *(Continued from Lemma 6.2.) The matrix $\mathbb{E} \{ s x^T \}$ is symmetric and depends on the curvature of the manifold \mathcal{P} such that*

$$\mathbb{E} \{ s x^T \} = \mathbb{E} \{ x s^T \} = I - \frac{1}{3} R_m(C) + \mathcal{O} \left(\mathbb{E} \left\{ (\sqrt{K_{\max}} \|X_{\theta}\|)^3 \right\} \right),$$

where K_{\max} is an upper bound on the absolute value of the sectional curvatures of \mathcal{P} and $R_m: \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}$ is a linear operator expressed w.r.t. the

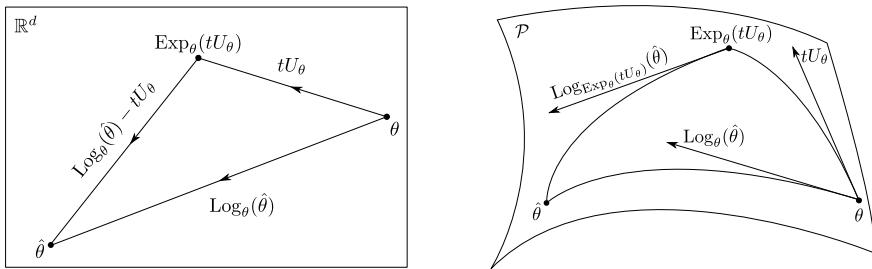


Figure 6.1: Comparison of a classical triangle and a geodesic triangle. Lemma 6.3 investigates how $\text{Log}_\theta(\hat{\theta})$ varies when the root point θ is moved infinitesimally along the direction tU_θ to $\text{Exp}_\theta(tU_\theta)$ (think of t very small). In a Euclidean space (left), $\text{Exp}_\theta(tU_\theta) = \theta + tU_\theta$ and $\text{Log}_\theta(\hat{\theta}) = \hat{\theta} - \theta$. Hence, the variation is simply $-tU_\theta$. On a curved Riemannian manifold (right), the difference between the logarithm at $\text{Exp}_\theta(tU_\theta)$ (parallel transported back to θ) and the logarithm at θ is not simply $-tU_\theta$ anymore: the curvature of \mathcal{P} is responsible for additional terms, elucidated by Lemma 6.3. As an extreme example of this, think of \mathcal{P} as the sphere and place $\hat{\theta}$ at a pole and θ on the equator. Moving θ along tU_θ parallel to the equator does not change the logarithm at all (up to parallel translation), meaning that the curvature terms are responsible for a large deviation from the normal (flat) behavior in this case. When θ and $\hat{\theta}$ are close-by compared to the scale at which curvature becomes a dominant feature, the curvature terms remain small.

basis e and in terms of the Riemannian curvature tensor \mathcal{R} on \mathcal{P} . It maps $d \times d$ symmetric matrices to $d \times d$ symmetric matrices as follows:¹

$$R_m(C)_{ij} = \mathbb{E} \{ \langle \mathcal{R}(X_\theta, e_i) e_j, X_\theta \rangle_\theta \}. \quad (6.6)$$

Since the right hand side is the expectation of a quadratic expression in X_θ , it is linear in the matrix C . Hence, this implicitly defines $R_m(A)$ for any symmetric A . If \mathcal{P} is flat, then $\mathcal{R} \equiv 0$ and similarly $R_m \equiv 0$ so that

$$\mathbb{E} \{ s x^\top \} = \mathbb{E} \{ x s^\top \} = I.$$

Proof. We give a proof for manifolds of constant sectional curvature $K \in \mathbb{R}$, that is, such that for all vector fields X, Y, Z ,

$$\mathcal{R}(X, Y)Z = K(\langle Y, Z \rangle X - \langle X, Z \rangle Y). \quad (6.7)$$

¹This definition differs from (Smith, 2005). In the latter paper, the notation R_m includes the higher-order terms and equation (6.6) only holds for sufficiently small errors. Comparatively, we define R_m via (6.6) and spell out the error terms where needed.

(See for example (Lee, 1997, Lemma 8.10).) For a general proof, we direct the reader to (Smith, 2005, Lemma 1). The proof in the latter reference is arguably difficult to follow, which prompted us to provide the present restricted but explicit argument. Our proof relies on a direct solve of the Jacobi equation, which may remain possible beyond manifolds of constant curvature provided they have additional structure (such as symmetry) but is impossible (analytically) for the general case.

We focus on the impact of curvature on the vector $(\nabla_U X)_\theta$. Assume θ and $\hat{\theta}$ are close enough such that there exists a unique minimizing geodesic γ with $\gamma(0) = \hat{\theta}$ and $\gamma(1) = \theta$:

$$\gamma(t) = \text{Exp}_{\hat{\theta}}(t \text{Log}_{\hat{\theta}}(\theta)).$$

Being a geodesic, γ has constant speed $\|\dot{\gamma}(t)\| = \text{dist}(\hat{\theta}, \theta) = \|X_\theta\|$. Our aim is to elucidate how γ is modified when the end-point θ is moved infinitesimally along U_θ . The language of Jacobi fields is dedicated to the study of such questions. Specifically, consider the vector fields J along γ which satisfy the (linear, differential) Jacobi equation (Lee, 1997, Thm. 10.2)

$$D_t^2 J + \mathcal{R}(J, \dot{\gamma})\dot{\gamma} = 0. \quad (6.8)$$

Above, D_t denotes covariant derivative along γ :

$$D_t J(t) \triangleq (\nabla_{\dot{\gamma}(t)} J)_{\gamma(t)}.$$

The solutions of (6.8) form a $2d$ -dimensional linear subspace (Lee, 1997, Cor. 10.5). Imposing two independent initial or boundary conditions singles out a unique solution. Impose $J(0) = 0$ and $J(1) = U_\theta$ to obtain the Jacobi field related to the perturbation of γ such that $\hat{\theta}$ remains fixed and θ is moved along U_θ . Then, Karcher (1977, App. C.3) relates J to $(\nabla_U X)_\theta$ via

$$-(\nabla_U X)_\theta = D_t J(1). \quad (6.9)$$

To leverage (6.9), we must first solve (6.8). We do so following mostly the method used in (Lee, 1997, Lemma 10.8). Plug the constant curvature assumption (6.7) in the Jacobi equation to obtain

$$D_t^2 J + K(\langle \dot{\gamma}, \dot{\gamma} \rangle J - \langle J, \dot{\gamma} \rangle \dot{\gamma}) = 0.$$

Since the solution exists and is unique, it is acceptable to “guess” J and check its validity. Decompose U_θ as

$$U_\theta = U_\theta^\perp + \alpha X_\theta, \text{ such that } \langle U_\theta^\perp, X_\theta \rangle = 0.$$

Let $E(t)$ be a parallel vector field along γ (that is, $D_t E(t) \equiv 0$) such that $E(1) = U_\theta^\perp$. Observe that both E and $\dot{\gamma}$ are parallel along γ and are orthogonal. Then, assume solutions of the form

$$J(t) = u_1(t)E(t) + u_2(t)\dot{\gamma}(t),$$

that is, J is the superposition of a normal and a tangent vector field to γ . The scalar functions u_1 and u_2 should satisfy the following ODE:

$$(u_1'' + K\|X_\theta\|^2 u_1)E + u_2''\dot{\gamma} = 0. \quad (6.10)$$

Since $\dot{\gamma}(1) = -X_\theta$, the boundary conditions are $u_1(0) = u_2(0) = 0$ and $u_1(1) = 1, u_2(1) = -\alpha$. Given that E and $\dot{\gamma}$ are orthogonal, equation (6.10) reduces to two separate ODE's for u_1 and u_2 . First, $u_2'' = 0$ and it is easy to see that

$$u_2(t) = -\alpha t.$$

Second, the linear, constant coefficient ODE $u_1'' + K\|X_\theta\|^2 u_1 = 0$ is readily solved:

$$u_1(t) = \begin{cases} t & \text{if } K = 0; \\ \frac{1}{\sin(\sqrt{K}\|X_\theta\|)} \sin(\sqrt{K}\|X_\theta\|t) & \text{if } K > 0; \\ \frac{1}{\sinh(\sqrt{-K}\|X_\theta\|)} \sinh(\sqrt{-K}\|X_\theta\|t) & \text{if } K < 0. \end{cases}$$

Hence, $J = u_1E + u_2\dot{\gamma}$ indeed solves the Jacobi equation. Motivated by (6.9), we now compute $D_t J(1)$:

$$\begin{aligned} D_t J(t) &= u_1'(t)E(t) - \alpha\dot{\gamma}(t), \\ D_t J(1) &= u_1'(1)U_\theta^\perp + \alpha X_\theta. \end{aligned} \quad (6.11)$$

The derivative $u_1'(1)$ is readily computed:

$$u_1'(1) = \begin{cases} 1 & \text{if } K = 0; \\ \sqrt{K}\|X_\theta\| \cot(\sqrt{K}\|X_\theta\|) & \text{if } K > 0; \\ \sqrt{-K}\|X_\theta\| \coth(\sqrt{-K}\|X_\theta\|) & \text{if } K < 0. \end{cases}$$

Remarkably, owing to the two Taylor expansions $x \cot(x) = 1 - x^2/3 + \mathcal{O}(x^4)$ and $x \coth(x) = 1 + x^2/3 + \mathcal{O}(x^4)$, it holds for any K that:

$$u_1'(1) = 1 - \frac{1}{3}K\|X_\theta\|^2 + \mathcal{O}(K^2\|X_\theta\|^4).$$

Plugging this into (6.11) yields, for all K ,

$$D_t J(1) = U_\theta - \frac{1}{3}K\|X_\theta\|^2 U_\theta^\perp + \mathcal{O}(K^2\|X_\theta\|^4)U_\theta^\perp.$$

We now aim at suppressing explicit references to K in favor of a more general-looking formulation involving \mathcal{R} . Owing to the skew-symmetry of the curvature tensor, $\mathcal{R}(X, X) = -\mathcal{R}(X, X) = 0$ and it follows that $\mathcal{R}(X, U) = \mathcal{R}(X, U^\perp)$, where U^\perp is the vector field obtained from U by suppressing the component parallel to X . Let V be any vector field on \mathcal{P} . Then, resorting to (6.7):

$$\langle \mathcal{R}(X, U)V, X \rangle_\theta = \langle \mathcal{R}(X, U^\perp)V, X \rangle_\theta = K\|X_\theta\|^2 \langle U_\theta^\perp, V_\theta \rangle.$$

Thus,

$$\langle D_t J(1), V_\theta \rangle = \langle U_\theta, V_\theta \rangle - \frac{1}{3} \langle \mathcal{R}(X, U)V, X \rangle_\theta + \mathcal{O}(K^2\|X_\theta\|^4) \langle U_\theta^\perp, V_\theta \rangle.$$

Owing to (6.9) and Lemma 6.2, taking expectations in the latter equation further shows

$$\begin{aligned} u^\top \mathbb{E}\{sx^\top\} v &= -\mathbb{E}\{\langle (\nabla_U X)_\theta, V_\theta \rangle_\theta\} \\ &= \langle U_\theta, V_\theta \rangle - \frac{1}{3} \mathbb{E}\{\langle \mathcal{R}(X, U)V, X \rangle_\theta\} \\ &\quad + \mathcal{O}(\mathbb{E}\{K^2\|X_\theta\|^4\}) \langle U_\theta^\perp, V_\theta \rangle. \end{aligned}$$

With the definition of $R_m(C)$ (6.6), in matrix notation this is equivalent to:

$$\mathbb{E}\{sx^\top\} = I - \frac{1}{3}R_m(C) + \mathcal{O}(\mathbb{E}\{K^2\|X_\theta\|^4\}).$$

Smith (2005, Lemma 1) argues that, even if \mathcal{P} does not have constant sectional curvature, the latter equation holds but the error term decays only cubically rather than quartically as $\mathcal{O}(\mathbb{E}\{(\sqrt{K_{\max}}\|X_\theta\|)^3\})$, with K_{\max} an upper bound on the maximum absolute value of sectional curvatures on \mathcal{P} . \square

In particular, the curvature terms are negligible when the dimensionless number $\sqrt{K_{\max}}\|X_\theta\|$ is small, that is, when estimation errors obey $\|X_\theta\| \ll 1/\sqrt{K_{\max}}$. The main theorem follows.

Theorem 6.4 (Intrinsic Cramér-Rao bound). *Let \mathcal{P} be a Riemannian manifold, let $\theta \in \mathcal{P}$ and let $e = (e_1, \dots, e_d)$ be an orthonormal basis of $T_\theta \mathcal{P}$. Consider an estimation problem on \mathcal{P} such that the FIM $F = F(\theta)$ (Definition 6.3) is invertible and $\lambda_{\max}(F^{-1})$ is small compared to $1/K_{\max}$. Then, for any unbiased estimator, the covariance matrix $C = C(\theta)$ (Definition 6.6) obeys the following matrix inequality, where both F and C are expressed w.r.t. the basis e :*

$$C \succeq F^{-1} - \frac{1}{3} (F^{-1}R_m(F^{-1}) + R_m(F^{-1})F^{-1}) + \mathcal{O}(\lambda_{\max}(F^{-1})^{2+1/2}),$$

with R_m as defined by equation (6.6). If the parameter space \mathcal{P} is flat, $R_m \equiv 0$ and the inequality simplifies to the celebrated

$$C \succeq F^{-1}.$$

Even for flat manifolds, these only hold for small enough errors so that the logarithm $\text{Log}_\theta(\hat{\theta})$ is well defined. For Euclidean spaces, there are no restrictions.

Proof. The main argument consists in building a well-chosen random vector $v \in \mathbb{R}^d$ such that the trivial matrix inequality $\mathbb{E}\{vv^\top\} \succeq 0$ leads to the sought result. Consider the following vector:²

$$v = x - F^{-1}s.$$

Notice that v has zero mean, since x and s have zero mean:

$$\mathbb{E}\{v\} = \mathbb{E}\{x\} - F^{-1}\mathbb{E}\{s\} = 0.$$

Now for the main argument:

$$\mathbb{E}\{vv^\top\} = \mathbb{E}\{xx^\top\} + F^{-1}\mathbb{E}\{ss^\top\}F^{-1} - F^{-1}\mathbb{E}\{sx^\top\} - \mathbb{E}\{xs^\top\}F^{-1} \succeq 0.$$

Inject $\mathbb{E}\{xx^\top\} = C$, $\mathbb{E}\{ss^\top\} = F$ and Lemma 6.3 for $\mathbb{E}\{sx^\top\} = \mathbb{E}\{xs^\top\}$:

$$\begin{aligned} C + F^{-1} - F^{-1} + \frac{1}{3}F^{-1}R_m(C) - F^{-1} + \frac{1}{3}R_m(C)F^{-1} \\ + \mathcal{O}(\mathbb{E}\{\|X_\theta\|^3\} \cdot \lambda_{\max}(F^{-1})) \succeq 0. \end{aligned}$$

Hence,

$$C + \frac{1}{3}(F^{-1}R_m(C) + R_m(C)F^{-1}) \succeq F^{-1} + \mathcal{O}(\mathbb{E}\{\|X_\theta\|^3\} \cdot \lambda_{\max}(F^{-1})).$$

The left hand side of this inequality is a linear function of the entries of the matrix C , with $\text{Id}: \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}$ the identity operator and $\Delta: \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}$ defined by

$$\Delta(C) = \frac{1}{3}(F^{-1}R_m(C) + R_m(C)F^{-1}),$$

so that the inequality reads

$$(\text{Id} + \Delta)(C) \succeq F^{-1} + \mathcal{O}(\mathbb{E}\{\|X_\theta\|^3\} \cdot \lambda_{\max}(F^{-1})).$$

²In (Smith, 2005), the vector $v = x - \mathbb{E}\{sx^\top\}F^{-1}s$ is considered instead, leading to the same result.

At large SNR, the operator Δ is small compared to Id , so that $\text{Id} + \Delta$ is positive definite and its inverse admits the Taylor expansion $(\text{Id} + \Delta)^{-1} = \text{Id} - \Delta + \Delta^2 - \dots$. Applying this to both sides of the inequality finally yields

$$\begin{aligned} C \succeq F^{-1} - \frac{1}{3} (F^{-1} R_m(F^{-1}) + R_m(F^{-1}) F^{-1}) + \\ \mathcal{O}(\mathbb{E} \{ \|X_\theta\|^3 \} \cdot \lambda_{\max}(F^{-1})). \end{aligned}$$

At large SNR, that is, for small F^{-1} , at best the average squared error $\|X_\theta\|^2$ is on the same order of magnitude as $\lambda_{\max}(F^{-1})$, so that the error terms scale as $\mathcal{O}(\lambda_{\max}(F^{-1})^{2+1/2})$, which is indeed higher order than the terms which are spelled out. Smith (2005) indicates an error term in $\lambda_{\max}(F^{-1})^3$ but we could not reproduce the argument. \square

Note that C and F are tied by an inequality even though C (as a tensor) depends on the chosen Riemannian metric whereas F (still as a tensor) does not. This apparent incompatibility is resolved by observing that the inverse of a tensor is defined with respect to the metric, so that F^{-1} (as a tensor) indeed depends on the metric too.

Chapter 7

Cramér-Rao bounds on submanifolds and quotient manifolds

In this chapter, we further consider estimation problems on Riemannian manifolds. Contrary to the previous chapter, we now focus on estimation problems such that the Fisher information matrix (FIM) F is *not necessarily* positive definite. Singularity of F typically arises when the measurements y are not sufficient to determine the parameter θ , that is, structural ambiguities remain. For example, locating a point $p = (x, y, z)$ in space based solely on information about the bearing $p/\|p\|$ is impossible, since nothing is known about the distance between p and the origin. The FIM of such a problem would only be positive *semidefinite*.

To resolve these ambiguities, one can proceed in at least two ways. Firstly, one can add constraints on θ , based on additional knowledge about the parameter. By restricting the parameter space to $\bar{\mathcal{P}} \subset \mathcal{P}$, a submanifold of \mathcal{P} , one may hope that the resulting estimation problem is well-posed. For example, if one knows beforehand that the distance between p and the origin is 1, one should perform the estimation on the sphere $\bar{\mathcal{P}} = \mathbb{S}^2 = \{(x, y, z) : x^2 + y^2 + z^2 = 1\}$ rather than on $\mathcal{P} = \mathbb{R}^3$. Alternatively, one can recognize that the parameter space is made of equivalence classes, that is, sets of parameters that are equally valid estimators for they give rise to the same measurement distribution. In this scenario, one ends up with an estimation problem on a quotient manifold $\bar{\mathcal{P}} = \mathcal{P}/\sim$, where \sim is an equivalence relation on \mathcal{P} stating that $\theta, \theta' \in \mathcal{P}$ are equivalent if they give rise to the same distribution of the measurements. Continuing with our example, all points p with the same bearing $p/\|p\|$ give rise to the

same measurement distribution, hence are indistinguishable and should be grouped into an equivalence class.

The treatment of submanifolds hereafter may also be useful when the FIM is invertible. In that scenario, one is interested in studying the Cramér-Rao bounds (CRB's) of the original problem, and the effect on those bounds caused by incorporating additional knowledge about θ .

The direct way to address ambiguities is to work on the smaller space $\bar{\mathcal{P}}$ directly, writing down Fisher information and covariance with respect to bases of the tangent spaces to $\bar{\mathcal{P}}$, leading to CRB's according to (Smith, 2005) (see the previous chapter). However, we argue that the tangent spaces of \mathcal{P} sometimes make more sense to the user: that is why the problem was defined on \mathcal{P} rather than $\bar{\mathcal{P}}$ to begin with. Furthermore, when $\bar{\mathcal{P}}$ is a quotient manifold, its tangent spaces are rather abstract objects to work with. It is hence desirable to have equivalent CRB's expressed as matrix inequalities w.r.t. bases of tangent spaces of \mathcal{P} instead. This is what the theorems in this chapter achieve. The present work derives the consequences of (Smith, 2005) for unbiased estimators in the presence of indeterminacies (ambiguities) or under additional constraints.

The case of constrained CRB's, that is, estimation on Riemannian submanifolds of \mathbb{R}^d , has been studied extensively (Ben-Haim & Eldar, 2009; Gorman & Hero, 1990; Stoica & Ng, 1998). Notably, in (Stoica & Ng, 1998), the authors describe $\bar{\mathcal{P}}$ through a set of equality constraints and they express the covariance in terms of distances in the embedding Euclidean space \mathbb{R}^d . In this chapter, we more generally consider Riemannian submanifolds of any Riemannian manifold \mathcal{P} . Furthermore, for the simple versions of the CRB's, only an orthogonal projector from the tangent spaces of \mathcal{P} to those of $\bar{\mathcal{P}}$ are required. More importantly, the covariance matrix in the proposed bounds is expressed in terms of the Riemannian, or geodesic, distance on $\bar{\mathcal{P}}$, which may be more natural for a number of applications.

The case of CRB's for estimation problems with singular FIM has also been investigated extensively (Ben-Haim & Eldar, 2009; Stoica & Marzetta, 2001; Xavier & Barroso, 2004). The classical remedy is to use the Moore-Penrose pseudoinverse, hereafter referred to as the pseudoinverse, of the FIM instead of the inverse in the CRB. We use the notation A^\dagger to denote the pseudoinverse of a matrix A . When the singularity is due to indeterminacies (a notion we make precise in Section 7.2), Xavier & Barroso (2004) showed a nice interpretation of the role of the pseudoinverse by recasting the estimation problem on a Riemannian quotient manifold $\bar{\mathcal{P}}$. In the latter reference, the authors give a geometric interpretation for the kernel of the FIM and propose a CRB-type bound they name IVLB (Xavier & Barroso, 2005) for the variance of unbiased estimators for such problems. In their bound, the possible curvature of $\bar{\mathcal{P}}$ is captured through a single number:

an upper bound on the sectional curvatures of $\bar{\mathcal{P}}$. In comparison, since the present results are based on (Smith, 2005), the proposed bounds concern the whole covariance matrix (the trace of which coincides with the variance). The pseudoinverse of the FIM appears naturally through the same manipulations as in (Xavier & Barroso, 2004). The additional curvature terms in the CRB (Section 7.3) take the whole Riemannian curvature tensor into account. This is especially useful when $\bar{\mathcal{P}}$ is flat or almost flat in most directions but has significant curvature in a few directions, which happens naturally for product spaces. In such scenarios, the IVLB tends to be overly optimistic, i.e., less restrictive—hence less informative—because it has to assume maximum curvature in all directions. In comparison, the bounds derived here based on (Smith, 2005) are able to capture complex curvature structures if need be.

Let $e = \{e_1, \dots, e_d\}$ be an orthonormal basis of $T_\theta \mathcal{P}$ w.r.t. the Riemannian metric $\langle \cdot, \cdot \rangle_\theta$. The FIM of the estimation problem on \mathcal{P} w.r.t. the basis e is a $d \times d$ symmetric, positive semidefinite matrix defined by (Definition 6.3):

$$(F_e)_{ij} = \mathbb{E} \{ \text{DL}(\theta)[e_i] \cdot \text{DL}(\theta)[e_j] \}, \quad (7.1)$$

where $L(\theta) = \log f(Y; \theta)$ is the log-likelihood function (Definition 6.1). The covariance matrix C_e w.r.t. the basis e is defined separately for the submanifold (Section 7.1) and the quotient manifold (Section 7.2) cases, then F_e and C_e are linked through matrix inequalities. At first, we neglect curvature terms that may appear due to the possible curvature of $\bar{\mathcal{P}}$. This results in simple statements (Theorems 7.2 and 7.3). These are practically useful because the curvature terms are often negligible at large SNR. Then, we establish the CRB's including curvature terms (Section 7.3). Finally, we illustrate the use of these theorems through an example (Section 7.4). The next chapter constitutes a more involved example of application for the theorems in this chapter.

7.1 Riemannian submanifolds

Consider the constrained estimation problem on the space $\bar{\mathcal{P}} \subset \mathcal{P}$, a Riemannian submanifold of \mathcal{P} , such that $\theta \in \bar{\mathcal{P}}$ and for which the log-likelihood function $\bar{L} = L|_{\bar{\mathcal{P}}}$ is the restriction of L to $\bar{\mathcal{P}}$. This situation arises when one adds supplementary constraints on the parameter θ . For example, some of the target parameters are known or deterministically related. We assume that the FIM for the estimation problem on $\bar{\mathcal{P}}$ is invertible, that is, the added constraints fix possible ambiguities in the estimation problem. Figure 7.1 depicts the situation.

Let $\hat{\theta}$ be any unbiased estimator for the estimation problem, that is, $\hat{\theta}: \mathcal{M} \rightarrow \bar{\mathcal{P}}$ maps every possible realization of the measurement y to a

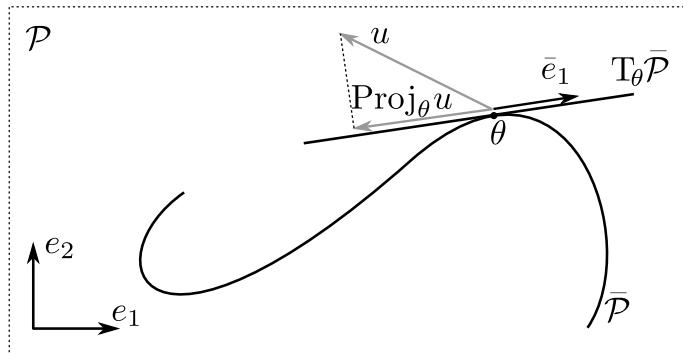


Figure 7.1: $\bar{\mathcal{P}}$ is a Riemannian submanifold of \mathcal{P} . We consider estimation problems for which the parameter to estimate is θ , a point of $\bar{\mathcal{P}}$. In this drawing, for simplicity, we chose $\mathcal{P} = \mathbb{R}^2$. The vectors $e = (e_1, e_2)$ form an orthonormal basis of $T_\theta \mathcal{P} \equiv \mathbb{R}^2$, while $\bar{e} = (\bar{e}_1)$ is an orthonormal basis of the tangent space $T_\theta \bar{\mathcal{P}}$. The operator Proj_θ projects vectors of $T_\theta \mathcal{P}$ orthogonally onto $T_\theta \bar{\mathcal{P}}$. We express the Cramér-Rao bounds for such problems in terms of the basis e , which at times may be more convenient than defining a basis \bar{e} for each point θ .

parameter $\hat{\theta}(y)$ and has zero bias (Definition 6.5):

$$\forall \theta \in \mathcal{P}, \quad b(\theta) = \mathbb{E} \left\{ \text{Log}_\theta(\hat{\theta}(y)) \right\} = 0,$$

where $\text{Log}_\theta: \bar{\mathcal{P}} \rightarrow T_\theta \bar{\mathcal{P}}$ is the logarithmic map at θ on $\bar{\mathcal{P}}$ (Section 2.6). For example, on a Euclidean space, $\text{Log}_\theta(\hat{\theta}(y)) = \hat{\theta}(y) - \theta$. For conciseness, we often write $\hat{\theta}$ to mean $\hat{\theta}(y)$. The covariance matrix of $\hat{\theta}$ w.r.t. the basis e is defined following Definition 6.6 as:

$$(C_e)_{ij} = \mathbb{E} \left\{ \left\langle \text{Log}_\theta(\hat{\theta}), e_i \right\rangle_\theta \cdot \left\langle \text{Log}_\theta(\hat{\theta}), e_j \right\rangle_\theta \right\}, \quad (7.2)$$

where, as always in this chapter, the expectation is taken w.r.t. the measurements $y \sim f(y; \theta)$. The goal is to link C_e and F_e through a matrix inequality.

Let $\bar{e} = \{\bar{e}_1, \dots, \bar{e}_{\bar{d}}\}$ be an orthonormal basis of $T_\theta \bar{\mathcal{P}} \subset T_\theta \mathcal{P}$ w.r.t. the Riemannian metric $\langle \cdot, \cdot \rangle_\theta$. Let E be the $\bar{d} \times d$ matrix such that $E_{ij} = \langle \bar{e}_i, e_j \rangle_\theta$. E is orthonormal: $EE^\top = I_{\bar{d}}$, but in general, $P_e \triangleq E^\top E \neq I_d$. Furthermore, let $\text{Proj}_\theta: T_\theta \mathcal{P} \rightarrow T_\theta \bar{\mathcal{P}}$ be the orthogonal projector onto $T_\theta \bar{\mathcal{P}}$. Clearly, P_e is the matrix representation of Proj_θ w.r.t. the basis e , that is: $\langle \text{Proj}_\theta e_i, e_j \rangle = (P_e)_{ij}$.

A direct application Theorem 6.4 to the estimation problem on $\bar{\mathcal{P}}$ would link the covariance matrix $C_{\bar{e}}$ of $\hat{\theta}$ and the inverse FIM $\bar{F}_{\bar{e}}^{-1}$ w.r.t. the basis \bar{e} . More precisely,

$$\begin{aligned} (C_{\bar{e}})_{ij} &= \mathbb{E} \left\{ \left\langle \text{Log}_{\theta}(\hat{\theta}), \bar{e}_i \right\rangle_{\theta} \cdot \left\langle \text{Log}_{\theta}(\hat{\theta}), \bar{e}_j \right\rangle_{\theta} \right\}, \\ (\bar{F}_{\bar{e}})_{ij} &= \mathbb{E} \left\{ \text{D}\bar{L}(\theta)[\bar{e}_i] \cdot \text{D}\bar{L}(\theta)[\bar{e}_j] \right\}, \\ C_{\bar{e}} &\succeq \bar{F}_{\bar{e}}^{-1} + \text{curvature terms}. \end{aligned} \quad (7.3)$$

We argue that it is sometimes convenient to work with C_e and F_e directly, to avoid the necessity to define and work with the basis \bar{e} . This is what the next theorem achieves, right after we establish a technical lemma.

Lemma 7.1. *Let $E \in \mathbb{R}^{\bar{d} \times d}$, $A \in \mathbb{R}^{d \times d}$, $B \in \mathbb{R}^{\bar{d} \times \bar{d}}$, with $\bar{d} \leq d$, $A = A^\top$, $B = B^\top$ and $EE^\top = I_{\bar{d}}$, i.e., E is orthonormal. Further assume that $\ker E \subset \ker A$. Then,*

$$EAE^\top \succeq B \Rightarrow A \succeq E^\top BE.$$

Proof. Since $\mathbb{R}^d = \text{im } E^\top \oplus \ker E$, for all $x \in \mathbb{R}^d$, there exist unique vectors $y \in \mathbb{R}^{\bar{d}}$ and $z \in \mathbb{R}^d$ such that $x = E^\top y + z$ and $Ez = 0$. It follows that:

$$\begin{aligned} x^\top Ax &= y^\top EAE^\top y + z^\top Az + 2y^\top EAz \\ &= y^\top EAE^\top y && (\text{since } Ez = 0 \Rightarrow Az = 0) \\ &\geq y^\top By && (\text{since } EAE^\top \succeq B) \\ &= x^\top E^\top BE x && (\text{since } Ex = EE^\top y + Ez = y.) \end{aligned}$$

This holds for all x , hence $A \succeq E^\top BE$. \square

Theorem 7.2 (CRB on submanifolds). *Given any unbiased estimator $\hat{\theta}$ for the estimation problem on the Riemannian submanifold $\bar{\mathcal{P}}$ with log-likelihood $\bar{L} = L|_{\bar{\mathcal{P}}}$ (6.2), at large SNR, the $d \times d$ covariance matrix C_e (7.2) and the $d \times d$ Fisher information matrix F_e (7.1) obey the matrix inequality (assuming $\text{rank}(P_e F_e P_e) = \bar{d}$):*

$$C_e \succeq (P_e F_e P_e)^\dagger + \text{curvature terms}, \quad (7.4)$$

where the $d \times d$ matrix $P_e = E^\top E$ represents the orthogonal projector from $T_\theta \mathcal{P}$ to $T_\theta \bar{\mathcal{P}}$ w.r.t. the basis e and \dagger denotes Moore-Penrose pseudoinversion. Furthermore, the spectrum of $(P_e F_e P_e)^\dagger$ is the spectrum of $\bar{F}_{\bar{e}}^{-1}$ with $d - \bar{d}$ additional zeroes. In particular, neglecting curvature terms:

$$\text{trace}(C_e) = \text{trace}(C_{\bar{e}}) \geq \text{trace}(\bar{F}_{\bar{e}}^{-1}) = \text{trace}((P_e F_e P_e)^\dagger).$$

Proof. Since $\hat{\theta} \in \bar{\mathcal{P}}$, $\text{Log}_\theta(\hat{\theta}) \in T_\theta \bar{\mathcal{P}}$. Consequently, for all $u \in T_\theta \mathcal{P}$,

$$\langle \text{Log}_\theta(\hat{\theta}), u \rangle_\theta = \langle \text{Log}_\theta(\hat{\theta}), \text{Proj}_\theta u \rangle_\theta,$$

where $\text{Proj}_\theta u$ is the orthogonal projection of u on $T_\theta \bar{\mathcal{P}}$. The orthogonal projection of the basis vector e_i on $T_\theta \bar{\mathcal{P}}$ expands in the basis \bar{e} as

$$\text{Proj}_\theta e_i = \sum_j \langle \bar{e}_j, e_i \rangle_\theta \bar{e}_j = \sum_j E_{ji} \bar{e}_j.$$

Then, by bilinearity, $(C_e)_{ij} = \sum_{k,\ell} E_{ki} E_{\ell j} (C_{\bar{e}})_{k\ell}$. In matrix form,

$$C_e = E^\top C_{\bar{e}} E.$$

Since $EE^\top = I_{\bar{d}}$, it also holds that $C_{\bar{e}} = EC_e E^\top$. The vectors of \bar{e} expand in the basis e as $\bar{e}_i = \sum_j \langle \bar{e}_i, e_j \rangle_\theta e_j = \sum_j E_{ij} e_j$. By bilinearity again, $(\bar{F}_{\bar{e}})_{ij} = \sum_{k,\ell} E_{ik} E_{j\ell} (F_e)_{k\ell}$. In matrix form,

$$\bar{F}_{\bar{e}} = EF_e E^\top.$$

Notice that the assumption $\text{rank}(P_e F_e P_e) = \bar{d}$ is equivalent to the assumption that $\bar{F}_{\bar{e}}$ is invertible. Then, substituting in (7.3), we find $EC_e E^\top \succeq (EF_e E^\top)^{-1}$. Since $\ker C_e = \ker(E^\top C_{\bar{e}} E) \supset \ker E$, Lemma 7.1 applies and it follows that (neglecting curvature terms):

$$C_e \succeq E^\top (EF_e E^\top)^{-1} E.$$

Finally, from the definition of pseudoinverse, it is easily checked that

$$E^\top (EF_e E^\top)^{-1} E = (E^\top EF_e E^\top E)^\dagger.$$

Since $P_e = E^\top E$, this concludes the proof of the main part.

We now establish the spectrum property. Since $\bar{F}_{\bar{e}}^{-1}$ is symmetric positive definite, there exist a diagonal matrix D and an orthogonal matrix U of size $\bar{d} \times \bar{d}$ such that $\bar{F}_{\bar{e}}^{-1} = UDU^\top$. Hence,

$$(P_e F_e P_e)^\dagger = E^\top UDU^\top E = V \begin{pmatrix} D & \\ & 0 \end{pmatrix} V^\top,$$

with $V = (E^\top U \quad (E^\top U)^\perp)$ a $d \times d$ orthogonal matrix. The trace property follows easily (neglecting curvature terms):

$$\text{trace}(C_e) = \text{trace}(E^\top C_{\bar{e}} E) = \text{trace}(C_{\bar{e}}) \geq \text{trace}(\bar{F}_{\bar{e}}^{-1}) = \text{trace}((P_e F_e P_e)^\dagger).$$

□

The trace property is especially interesting, as it bounds the variance of the estimator $\hat{\theta}$, expressed w.r.t. the Riemannian distance dist on $\bar{\mathcal{P}}$:

$$\text{trace}(C_e) = \text{trace}(C_{\bar{e}}) = \mathbb{E} \left\{ \|\text{Log}_{\theta}(\hat{\theta})\|^2 \right\} = \mathbb{E} \left\{ \text{dist}^2(\theta, \hat{\theta}) \right\}.$$

Here is one way of interpreting the bound (7.4). Expand the random error vector $\text{Log}_{\theta}(\hat{\theta}) = \sum_i x_i e_i$ with random coefficients x_i . From the definition, $(C_e)_{ii} = \mathbb{E} \{x_i^2\}$. Then, equation (7.4) implies $\mathbb{E} \{x_i^2\} \geq (P_e F_e P_e)_{ii}^{\dagger}$, which limits how well the i^{th} coordinate can be estimated. For example, when $\bar{\mathcal{P}}$ is Euclidean, $\text{Log}_{\theta}(\hat{\theta}) = \hat{\theta} - \theta$ and $\mathbb{E} \{x_i^2\} = \mathbb{E} \{(\hat{\theta}_i - \theta_i)^2\}$.

Notice that it is not necessary to explicitly construct a basis \bar{e} in order to use Theorem 7.2. Indeed, the orthogonal projector P_e is often easy to compute without requiring an explicit factorization as $E^{\top} E$. For example, the orthogonal projector from \mathbb{R}^3 onto the tangent space to the sphere \mathbb{S}^2 at θ , denoted $T_{\theta}\mathbb{S}^2$, w.r.t. the canonical basis of \mathbb{R}^3 is simply $P_e = I_3 - \theta\theta^{\top}$, where I_3 is the 3×3 identity matrix. This is fortunate since, because of the hairy ball theorem, it is impossible to define bases \bar{e} of $T_{\theta}\mathbb{S}^2$ for all θ in a smooth way, making it rather inconvenient to work with such bases.

7.2 Riemannian quotient manifolds

Whenever two parameters $\theta, \theta' \in \mathcal{P}$ give rise to the same measurement distribution, they are indistinguishable, in the sense that no argument based on the observed measurement can be used to favor one parameter over the other as estimator. This observation motivates the definition of the following equivalence relation (remember the definitions of the parameterized pdf of the measurements f (6.1) and of the log-likelihood function L (6.2)):

$$\theta \sim \theta' \Leftrightarrow f(\cdot, \theta) \equiv f(\cdot, \theta') \text{ almost everywhere on } \mathcal{M}. \quad (7.5)$$

The quotient space $\bar{\mathcal{P}} = \mathcal{P}/\sim$ —that is, the set of equivalence classes—then becomes the natural parameter space on which the estimation should be performed. Figures 7.2 and 7.3, courtesy of Xavier & Barroso (2004), depict the concept of quotient manifold and of the related basic objects we introduce hereafter, namely submersions and horizontal/vertical spaces. See also sections 2.2.2, 2.3.2 and 2.4.2.

We now consider the mapping π from \mathcal{P} to $\bar{\mathcal{P}}$, which maps each parameter θ to its equivalence class $[\theta]$,

$$\pi: \mathcal{P} \rightarrow \bar{\mathcal{P}}: \theta \mapsto \pi(\theta) = [\theta] \triangleq \{\theta' \in \mathcal{P} : \theta' \sim \theta\},$$

and concentrate on the case where π is a Riemannian submersion, see Absil *et al.* (2008); O'Neill (1983) or Section 2.3.2. That is, $\bar{\mathcal{P}}$ is a Riemannian

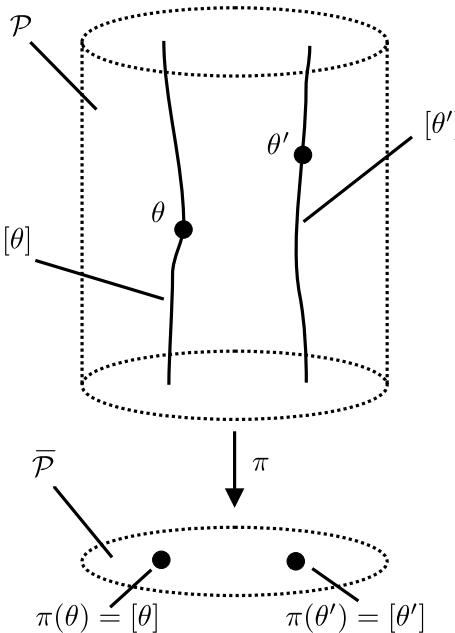


Figure 7.2: The parameter space \mathcal{P} is partitioned into equivalence classes, called fibers. The Riemannian submersion π maps each $\theta \in \mathcal{P}$ to its corresponding equivalence class $[\theta] \in \bar{\mathcal{P}}$. The space of equivalence classes is the quotient space $\bar{\mathcal{P}} = \mathcal{P}/\sim$, also a Riemannian manifold. *Figure courtesy of Xavier & Barroso (2004).*

quotient manifold of \mathcal{P} . In particular, $[\theta]$ is a Riemannian submanifold of \mathcal{P} (a *fiber*). The log-likelihood function $\bar{L}: \bar{\mathcal{P}} \rightarrow \mathbb{R}$ is well-defined by $\bar{L}([\theta]) \triangleq L(\theta)$.

The tangent space to $[\theta]$ at θ , named the vertical space V_θ , is a subspace of the tangent space $T_\theta \mathcal{P}$. The orthogonal complement of the vertical space, named the horizontal space H_θ , is such that $T_\theta \mathcal{P} = H_\theta \oplus V_\theta$. The pushforward $D\pi(\theta): T_\theta \mathcal{P} \rightarrow T_{[\theta]} \bar{\mathcal{P}}$ of a Riemannian submersion induces a metric on the abstract tangent space $T_{[\theta]} \bar{\mathcal{P}}$:

$$\forall u, v \in H_\theta, \quad \langle D\pi(\theta)[u], D\pi(\theta)[v] \rangle_{[\theta]} \triangleq \langle u, v \rangle_\theta.$$

The definition of Riemannian submersion ensures that this is well-defined, see (Absil *et al.*, 2008). We mention two key properties, with \ker denoting

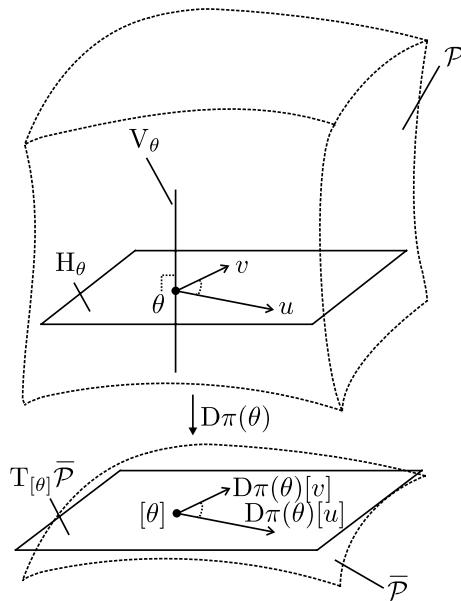


Figure 7.3: Each fiber $\pi(\theta) = [\theta]$ is a Riemannian submanifold of \mathcal{P} . The tangent space to a fiber at θ is the vertical space V_θ . The orthogonal complement of V_θ in $T_\theta \mathcal{P}$ is the horizontal space H_θ . The differential of π , noted $D\pi(\theta)$, is an isometry between H_θ and the abstract tangent space $T_{[\theta]} \bar{\mathcal{P}}$. This makes it convenient to represent abstract tangent vectors to $\bar{\mathcal{P}}$ as horizontal vectors. *Figure courtesy of Xavier & Barroso (2004).*

the kernel or null space:

$$\ker D\pi(\theta) = V_\theta, \quad \text{and} \\ D\pi(\theta)|_{H_\theta} : H_\theta \rightarrow T_{[\theta]} \bar{\mathcal{P}} \text{ is an isometry.}$$

Let $[\hat{\theta}] : \mathcal{M} \rightarrow \bar{\mathcal{P}}$ be any unbiased estimator for the present problem. Define the covariance matrix of $[\hat{\theta}]$ w.r.t. the basis e following Definition 6.6 as:

$$(C_e)_{ij} = \mathbb{E} \{ \langle \xi, e_i \rangle_\theta \cdot \langle \xi, e_j \rangle_\theta \}, \text{ with} \\ \xi = (D\pi(\theta)|_{H_\theta})^{-1} [\text{Log}_{[\theta]}([\hat{\theta}])]. \quad (7.6)$$

The error vector ξ is the shortest horizontal vector at θ such that $\text{Exp}_\theta(\xi) \in [\hat{\theta}]$. The exponential map is the inverse of the logarithmic map, see Section 2.6. On a Euclidean space, $\text{Exp}_\theta(\xi) = \theta + \xi$.

Let $\bar{e} = (\bar{e}_1, \dots, \bar{e}_{\bar{d}})$ be an orthonormal basis of $T_{[\theta]} \bar{\mathcal{P}}$. A direct application of Theorem 6.4 to the estimation problem on $\bar{\mathcal{P}}$ would link the covariance matrix $C_{\bar{e}}$ of $[\hat{\theta}]$ and the inverse FIM $\bar{F}_{\bar{e}}^{-1}$ w.r.t. the basis \bar{e} . More precisely,

$$\begin{aligned} (C_{\bar{e}})_{ij} &= \mathbb{E} \left\{ \left\langle \text{Log}_{[\theta]}([\hat{\theta}]), \bar{e}_i \right\rangle_{[\theta]} \cdot \left\langle \text{Log}_{[\theta]}([\hat{\theta}]), \bar{e}_j \right\rangle_{[\theta]} \right\}, \\ (\bar{F}_{\bar{e}})_{ij} &= \mathbb{E} \{ D\bar{L}([\theta])[\bar{e}_i] \cdot D\bar{L}([\theta])[\bar{e}_j] \}, \\ C_{\bar{e}} &\succeq \bar{F}_{\bar{e}}^{-1} + \text{curvature terms}. \end{aligned} \quad (7.7)$$

Since $T_{[\theta]} \bar{\mathcal{P}}$ is an abstract space, we argue that it is often convenient to work with the more concrete objects C_e and F_e instead.

Theorem 7.3 (CRB on quotient manifolds). *Given any unbiased estimator $[\hat{\theta}]$ for the estimation problem on the Riemannian quotient manifold $\bar{\mathcal{P}} = \mathcal{P}/\sim$ (7.5) with log-likelihood L (6.2), at large SNR, the $d \times d$ covariance matrix C_e (7.6) and the $d \times d$ Fisher information matrix F_e (7.1) obey the matrix inequality (assuming $\text{rank}(F_e) = \bar{d}$):*

$$C_e \succeq F_e^\dagger + \text{curvature terms},$$

where \dagger denotes Moore-Penrose pseudoinversion. Furthermore, the spectrum of F_e^\dagger is the spectrum of $\bar{F}_{\bar{e}}^{-1}$ with $d - \bar{d}$ additional zeroes. In particular, neglecting curvature terms:

$$\text{trace}(C_e) = \text{trace}(C_{\bar{e}}) \geq \text{trace}(\bar{F}_{\bar{e}}^{-1}) = \text{trace}(F_e^\dagger).$$

Proof. It is convenient to introduce the orthonormal basis of H_θ related to \bar{e} as $\tilde{e} = (\tilde{e}_1, \dots, \tilde{e}_{\bar{d}})$, with $\tilde{e}_i = D\pi(\theta)[\tilde{e}_i]$. The $\bar{d} \times d$ matrix E such that $E_{ij} = \langle \tilde{e}_i, e_j \rangle_\theta$ will prove useful. E is orthonormal: $EE^\top = I_{\bar{d}}$, but in general, $E^\top E \neq I_d$.

Let us denote the orthogonal projection of $u \in T_\theta \mathcal{P}$ onto the horizontal space H_θ as $\text{Proj}_\theta^h u$. Since ξ (7.6) is a horizontal vector, $\langle \xi, u \rangle_\theta = \langle \xi, \text{Proj}_\theta^h u \rangle$. Furthermore, $D\pi(\theta)[\text{Proj}_\theta^h u] = D\pi(\theta)[u]$. Then, using the fact that $D\pi(\theta)|_{H_\theta}$ is an isometry, it follows that

$$\begin{aligned} (C_e)_{ij} &= \mathbb{E} \{ \langle \xi, e_i \rangle_\theta \cdot \langle \xi, e_j \rangle_\theta \} \\ &= \mathbb{E} \left\{ \left\langle \xi, \text{Proj}_\theta^h e_i \right\rangle_\theta \cdot \left\langle \xi, \text{Proj}_\theta^h e_j \right\rangle_\theta \right\} \\ &= \mathbb{E} \left\{ \left\langle \text{Log}_{[\theta]}([\hat{\theta}]), D\pi(\theta)[e_i] \right\rangle_{[\theta]} \cdot \left\langle \text{Log}_{[\theta]}([\hat{\theta}]), D\pi(\theta)[e_j] \right\rangle_{[\theta]} \right\}. \end{aligned}$$

The vector $D\pi(\theta)[e_i] \in T_{[\theta]} \bar{\mathcal{P}}$ expands in the basis \bar{e} as $D\pi(\theta)[e_i] = \sum_j E_{ji} \bar{e}_j$. Indeed,

$$\langle D\pi(\theta)[e_i], \bar{e}_j \rangle_{[\theta]} = \langle D\pi(\theta)[e_i], D\pi(\theta)[\tilde{e}_j] \rangle_{[\theta]} = \langle e_i, \tilde{e}_j \rangle_\theta.$$

It follows that $(C_e)_{ij} = \sum_{k,\ell} E_{ki} E_{\ell j} (C_{\bar{e}})_{k\ell}$. In matrix form:

$$C_e = E^\top C_{\bar{e}} E.$$

Since $EE^\top = I_{\bar{d}}$, it also holds that $C_{\bar{e}} = EC_e E^\top$.

We now similarly link F_e and $\bar{F}_{\bar{e}}$. In doing so, we exploit the fact that the gradient $\text{grad } L(\theta)$ is a horizontal vector. This stems from the fact that the log-likelihood function L is constant over fibers (equivalence classes).

$$\begin{aligned} (F_e)_{ij} &= \mathbb{E} \{ DL(\theta)[e_i] \cdot DL(\theta)[e_j] \} \\ &= \mathbb{E} \{ \langle \text{grad } L(\theta), e_i \rangle_\theta \cdot \langle \text{grad } L(\theta), e_j \rangle_\theta \} \\ &= \mathbb{E} \left\{ \left\langle \text{grad } L(\theta), \text{Proj}_\theta^h e_i \right\rangle_\theta \cdot \left\langle \text{grad } L(\theta), \text{Proj}_\theta^h e_j \right\rangle_\theta \right\} \\ &\quad (\text{expand } \text{Proj}_\theta^h e_i \text{ and } \text{Proj}_\theta^h e_j \text{ in the basis } \tilde{e}) \\ &= \sum_{k,\ell} E_{ki} E_{\ell j} \mathbb{E} \{ \langle \text{grad } L(\theta), \tilde{e}_k \rangle_\theta \cdot \langle \text{grad } L(\theta), \tilde{e}_\ell \rangle_\theta \} \\ &= \sum_{k,\ell} E_{ki} E_{\ell j} \mathbb{E} \left\{ \langle D\pi(\theta)[\text{grad } L(\theta)], \bar{e}_k \rangle_{[\theta]} \cdot \langle D\pi(\theta)[\text{grad } L(\theta)], \bar{e}_\ell \rangle_{[\theta]} \right\} \\ &= \sum_{k,\ell} E_{ki} E_{\ell j} \mathbb{E} \left\{ \langle \text{grad } \bar{L}([\theta]), \bar{e}_k \rangle_{[\theta]} \cdot \langle \text{grad } \bar{L}([\theta]), \bar{e}_\ell \rangle_{[\theta]} \right\} \\ &= \sum_{k,\ell} E_{ki} E_{\ell j} \mathbb{E} \{ D\bar{L}([\theta])[\bar{e}_k] \cdot D\bar{L}([\theta])[\bar{e}_\ell] \} \\ &= \sum_{k,\ell} E_{ki} E_{\ell j} (\bar{F}_{\bar{e}})_{k\ell}. \end{aligned}$$

In matrix form,

$$F_e = E^\top \bar{F}_{\bar{e}} E.$$

Notice that the assumption $\text{rank}(F_e) = \bar{d}$ is equivalent to the assumption that $\bar{F}_{\bar{e}}$ is invertible. The latter equation thus highlights that $\ker F_e = \ker E$, which makes sense since $\ker E$ is the vertical space V_θ (more precisely, it is the space of coordinate vectors of vertical vectors w.r.t. the basis e). Again, by orthonormality of E , it also holds that $\bar{F}_{\bar{e}} = EF_e E^\top$. Combining these rules, it follows that:

$$F_e = E^\top E F_e E^\top E.$$

Applying Lemma 7.1 to the inequality (7.7) and using arguments similar to the proof of Theorem 7.2 finally yields:

$$C_e \succeq F_e^\dagger + \text{curvature terms},$$

since

$$E^\top (EF_e E^\top)^{-1} E = (E^\top EF_e E^\top E)^\dagger = F_e^\dagger.$$

The spectrum and trace properties follow directly, see proof of Theorem 7.2. \square

Again, there is no need to construct bases \tilde{e} or \bar{e} in order to use Theorem 7.3. Notice that it still holds that $\text{trace}(C_e) = \text{trace}(C_{\bar{e}}) = \mathbb{E}\{\|\xi\|_\theta^2\} = \mathbb{E}\{\text{dist}^2([\theta], [\hat{\theta}])\}$, where dist is the Riemannian distance on $\bar{\mathcal{P}}$, since the map $D\pi(\theta)|_{H_\theta}$ is an isometry.

7.3 Including curvature terms

The intrinsic CRB's developed in Chapter 6 include special terms accounting for the possible curvature of the parameter space $\bar{\mathcal{P}}$. The curvature terms vanish if $\bar{\mathcal{P}}$ is flat, that is, if it is locally isometric to a Euclidean space. In such cases, theorems 7.2 and 7.3 suffice. When $\bar{\mathcal{P}}$ is not flat, the curvature terms may nevertheless often be neglected for high enough SNR. The argument developed in the previous chapter to that end concludes that neglecting the curvature terms is legitimate as soon as estimation errors obey

$$\text{dist}(\theta, \hat{\theta}) \ll \frac{1}{\sqrt{K_{\max}}}, \quad (7.8)$$

where K_{\max} is an upper bound on the absolute value of the sectional curvatures of $\bar{\mathcal{P}}$ at θ . Intuitively, this is the scale at which curvature plays a minor role.

Condition (7.8) involves an upper bound on the sectional curvature of $\bar{\mathcal{P}}$. As a consequence, it may be overly restrictive for parameter spaces which have small curvature in most directions, and large curvature in a few. An important class of such spaces consists in all product manifolds.

As an example, let us consider the problem of estimating $(\theta_1, \dots, \theta_N) \in \bar{\mathcal{P}} = \mathbb{S}^2 \times \dots \times \mathbb{S}^2$, the product of N spheres. $\bar{\mathcal{P}}$ has unit curvature along tangent 2-planes (two-dimensional subspaces of the tangent spaces of $\bar{\mathcal{P}}$) pertaining to a single sphere, but zero curvature along all 2-planes spanning exactly two distinct spheres. Of course, $K_{\max} = 1$. If estimating θ_i and θ_j , $i \neq j$, are two independent but identical tasks, one should expect the distribution of $\text{dist}(\theta_i, \hat{\theta}_i)$ to be independent of i . Consequently, $\text{dist}(\theta, \hat{\theta})$ grows as \sqrt{N} , whereas K_{\max} remains constant. Hence, condition (7.8) becomes increasingly restrictive with growing N . Of course, since the N tasks are independent and can be considered separately, the negligibility of the

curvature terms should not depend on N , which brings the conclusion that simply describing the curvature of $\bar{\mathcal{P}}$ through K_{\max} may not be enough.

For such parameter spaces, it is necessary to explicitly compute the curvature terms in the intrinsic Cramér-Rao bounds, if only to show that they are indeed negligible at reasonable SNR. We now set out to give versions of theorems 7.2 and 7.3 including curvature terms, computable without constructing other bases than e , the basis of $T_\theta \mathcal{P}$. This will require the Riemannian curvature tensor of $\bar{\mathcal{P}}$. Useful references to look up/compute this tensor are (O’Neill, 1983, Lemma 3.39, Cor. 3.58, Thm 7.47, Cor. 11.10)(Lee, 1997)(Chavel, 1993).

7.3.1 Curvature terms for submanifolds

The random error vector $X_\theta \triangleq \text{Log}_\theta(\hat{\theta})$ expands in the basis \bar{e} as $X_\theta = \sum_i \bar{x}_i \bar{e}_i$, with $\bar{x}_1, \dots, \bar{x}_{\bar{d}}$ random variables. Notice that

$$(C_{\bar{e}})_{ij} = \mathbb{E} \{ \langle X_\theta, \bar{e}_i \rangle_\theta \langle X_\theta, \bar{e}_j \rangle_\theta \} = \mathbb{E} \{ \bar{x}_i \bar{x}_j \}.$$

Let $\bar{\mathcal{R}}$ be the Riemannian curvature tensor of $\bar{\mathcal{P}}$ (See Section 2.8 for a brief introduction to curvature). The mapping $(u, v, w, z) \in (T_\theta \bar{\mathcal{P}})^4 \mapsto \langle \bar{\mathcal{R}}(u, v)w, z \rangle_\theta$ is linear in its four arguments. Smith introduces the symmetric 2-form $\bar{\mathbf{R}}_m: T_\theta \bar{\mathcal{P}} \times T_\theta \bar{\mathcal{P}} \rightarrow \mathbb{R}$ defined by (Smith, 2005, eq.(34)):

$$\begin{aligned} \bar{\mathbf{R}}_m[\bar{e}_i, \bar{e}_j] &= \mathbb{E} \{ \langle \bar{\mathcal{R}}(X_\theta, \bar{e}_i)\bar{e}_j, X_\theta \rangle_\theta \} \\ &= \mathbb{E} \left\{ \sum_{k, \ell} \langle \bar{\mathcal{R}}(\bar{e}_k, \bar{e}_i)\bar{e}_j, \bar{e}_\ell \rangle_\theta \bar{x}_k \bar{x}_\ell \right\} \\ &= \sum_{k, \ell} \langle \bar{\mathcal{R}}(\bar{e}_k, \bar{e}_i)\bar{e}_j, \bar{e}_\ell \rangle_\theta (C_{\bar{e}})_{k\ell}. \end{aligned}$$

From the latter expression, it is apparent that the entries of the matrix associated to $\bar{\mathbf{R}}_m$ are linear combinations of the entries of $C_{\bar{e}}$. Generalizing this to any symmetric matrix, the following linear map is defined, as for (6.6) in the previous chapter:

$$\begin{aligned} \bar{R}_m: \mathbb{R}^{\bar{d} \times \bar{d}} &\rightarrow \mathbb{R}^{\bar{d} \times \bar{d}}: M \mapsto \bar{R}_m(M), \text{ with} \\ (\bar{R}_m(M))_{ij} &= \sum_{k, \ell} \langle \bar{\mathcal{R}}(\bar{e}_k, \bar{e}_i)\bar{e}_j, \bar{e}_\ell \rangle_\theta M_{k\ell}. \end{aligned}$$

At large SNR, the CRB with curvature terms is given by Theorem 6.4:

$$C_{\bar{e}} \succeq \bar{F}_{\bar{e}}^{-1} - \frac{1}{3} (\bar{R}_m(\bar{F}_{\bar{e}}^{-1})\bar{F}_{\bar{e}}^{-1} + \bar{F}_{\bar{e}}^{-1}\bar{R}_m(\bar{F}_{\bar{e}}^{-1})). \quad (7.9)$$

In order to provide an equivalent of (7.9) only referencing the basis e , we introduce the following symmetric 2-form on $T_\theta \bar{\mathcal{P}} \times T_\theta \bar{\mathcal{P}}$:

$$\mathbf{R}_m[e_i, e_j] \triangleq \bar{\mathbf{R}}_m[\text{Proj}_\theta e_i, \text{Proj}_\theta e_j].$$

Notice that, since $X_\theta \in T_\theta \bar{\mathcal{P}}$, we have $X_\theta = \text{Proj}_\theta X_\theta$. Expanding in the basis e , $X_\theta = \sum_i x_i e_i = \sum_i x_i \text{Proj}_\theta e_i$ with random variables x_1, \dots, x_d and $(C_e)_{ij} = \mathbb{E}\{x_i x_j\}$. It follows that:

$$\begin{aligned} \mathbf{R}_m[e_i, e_j] &= \mathbb{E}\{\langle \bar{\mathcal{R}}(X_\theta, \text{Proj}_\theta e_i) \text{Proj}_\theta e_j, X_\theta \rangle_\theta\} \\ &= \sum_{k, \ell} \langle \bar{\mathcal{R}}(\text{Proj}_\theta e_k, \text{Proj}_\theta e_i) \text{Proj}_\theta e_j, \text{Proj}_\theta e_\ell \rangle_\theta (C_e)_{k\ell}. \end{aligned}$$

From there, we introduce the following linear map:

$$\begin{aligned} R_m: \mathbb{R}^{d \times d} &\rightarrow \mathbb{R}^{d \times d}: M \mapsto R_m(M), \text{ with} \\ (R_m(M))_{ij} &= \sum_{k, \ell} \langle \bar{\mathcal{R}}(\text{Proj}_\theta e_k, \text{Proj}_\theta e_i) \text{Proj}_\theta e_j, \text{Proj}_\theta e_\ell \rangle_\theta M_{k\ell}. \end{aligned} \quad (7.10)$$

Riemannian curvature is often specified by a formula for $\langle \bar{\mathcal{R}}(u, v)v, u \rangle$. Hence, the standard polarization identity for symmetric bilinear forms may be useful to compute \mathbf{R}_m :

$$4\mathbf{R}_m[e_i, e_j] = \mathbf{R}_m[e_i + e_j, e_i + e_j] - \mathbf{R}_m[e_i - e_j, e_i - e_j].$$

The linear maps R_m and \bar{R}_m appear in the following theorem.

Theorem 7.4 (CRB on submanifolds, with curvature). (*Continued from Theorem 7.2*). Including terms due to the possible curvature of $\bar{\mathcal{P}}$, at large SNR, the covariance matrix C_e (7.2) of any unbiased estimator $\hat{\theta}: \mathcal{M} \rightarrow \bar{\mathcal{P}}$ and the Fisher information matrix F_e (7.1) w.r.t. the orthonormal basis e of $T_\theta \bar{\mathcal{P}}$ obey the following matrix inequality (assuming $\text{rank}(P_e F_e P_e) = \bar{d}$):

$$C_e \succeq \tilde{F}_e^\dagger - \frac{1}{3} \left(R_m(\tilde{F}_e^\dagger) \tilde{F}_e^\dagger + \tilde{F}_e^\dagger R_m(\tilde{F}_e^\dagger) \right),$$

where $\tilde{F}_e = P_e F_e P_e$ and $R_m: \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}$ is as defined by (7.10).

Proof. We start from the CRB w.r.t. the basis \bar{e} (7.9):

$$C_{\bar{e}} \succeq \bar{F}_{\bar{e}}^{-1} - \frac{1}{3} \left(\bar{R}_m(\bar{F}_{\bar{e}}^{-1}) \bar{F}_{\bar{e}}^{-1} + \bar{F}_{\bar{e}}^{-1} \bar{R}_m(\bar{F}_{\bar{e}}^{-1}) \right).$$

By expanding the projections $\text{Proj}_\theta e_i = \sum_j \langle \bar{e}_j, e_i \rangle \bar{e}_j = \sum_j E_{ji} \bar{e}_j$ and exploiting the linearity of $\langle \bar{\mathcal{R}}(u, v)w, z \rangle_\theta$ in its four arguments, the matrix relation below comes forth:

$$\forall M = M^\top \in \mathbb{R}^{d \times d}, \quad R_m(M) = E^\top \bar{R}_m(EME^\top) E. \quad (7.11)$$

From the proof of Theorem 7.2, recall that $C_{\bar{e}} = EC_e E^\top$ and

$$\bar{F}_{\bar{e}}^{-1} = E(P_e F_e P_e)^\dagger E^\top.$$

The relation (7.11) yields $\bar{R}_m(\bar{F}_{\bar{e}}^{-1}) = ER_m((P_e F_e P_e)^\dagger)E^\top$. Substituting in the CRB gives:

$$EC_e E^\top \succeq E \left(\tilde{F}_e^\dagger - \frac{1}{3} \left(R_m(\tilde{F}_e^\dagger) \tilde{F}_e^\dagger + \tilde{F}_e^\dagger R_m(\tilde{F}_e^\dagger) \right) \right) E^\top,$$

where we used the fact that $R_m(M)P_e = P_e R_m(M) = R_m(M)$, which is easily established from (7.11). Lemma 7.1 applies and concludes the proof, since $P_e(P_e F_e P_e)^\dagger P_e = (P_e F_e P_e)^\dagger$. \square

7.3.2 Curvature terms for quotient manifolds

We follow the same line of thought as for submanifolds. The random error vector $X_{[\theta]} \triangleq \text{Log}_{[\theta]}([\hat{\theta}])$ expands in the basis \bar{e} as $X_{[\theta]} = \sum_i \bar{x}_i \bar{e}_i$, with $\bar{x}_1, \dots, \bar{x}_{\bar{d}}$ random variables and $(C_{\bar{e}})_{ij} = \mathbb{E}\{\bar{x}_i \bar{x}_j\}$. Let $\bar{\mathcal{R}}$ be the Riemannian curvature tensor of $\bar{\mathcal{P}}$. We consider $\bar{\mathbf{R}}_m: T_{[\theta]}\bar{\mathcal{P}} \times T_{[\theta]}\bar{\mathcal{P}} \rightarrow \mathbb{R}$ defined by:

$$\begin{aligned} \bar{\mathbf{R}}_m[\bar{e}_i, \bar{e}_j] &= \mathbb{E} \left\{ \langle \bar{\mathcal{R}}(X_{[\theta]}, \bar{e}_i)\bar{e}_j, X_{[\theta]} \rangle_{[\theta]} \right\} \\ &= \sum_{k, \ell} \langle \bar{\mathcal{R}}(\bar{e}_k, \bar{e}_i)\bar{e}_j, \bar{e}_\ell \rangle_{[\theta]} (C_{\bar{e}})_{k\ell}. \end{aligned}$$

A linear map on $\bar{d} \times \bar{d}$ symmetric matrices follows, in agreement with (6.6):

$$\begin{aligned} \bar{R}_m: \mathbb{R}^{\bar{d} \times \bar{d}} &\rightarrow \mathbb{R}^{\bar{d} \times \bar{d}}: M \mapsto \bar{R}_m(M), \text{ with} \\ (\bar{R}_m(M))_{ij} &= \sum_{k, \ell} \langle \bar{\mathcal{R}}(\bar{e}_k, \bar{e}_i)\bar{e}_j, \bar{e}_\ell \rangle_{[\theta]} M_{k\ell}. \end{aligned} \tag{7.12}$$

Again, at large SNR, the CRB (7.9) holds. To express it only referencing the basis e , we introduce the following symmetric 2-form:

$$\mathbf{R}_m[e_i, e_j] \triangleq \bar{\mathbf{R}}_m[D\pi(\theta)[e_i], D\pi(\theta)[e_j]].$$

Let $\xi = (D\pi(\theta)|_{H_\theta})^{-1}[X_{[\theta]}]$ be the unique horizontal vector at θ such that $D\pi(\theta)[\xi] = X_{[\theta]}$ (the lift of the error vector). Expanding ξ in the basis e as $\xi = \sum_i x_i e_i$, we find $X_{[\theta]} = \sum_i x_i D\pi(\theta)[e_i]$ with random variables x_1, \dots, x_d and $(C_e)_{ij} = \mathbb{E}\{x_i x_j\}$. It follows that:

$$\mathbf{R}_m[e_i, e_j] = \sum_{k, \ell} \langle \bar{\mathcal{R}}(D\pi(\theta)[e_k], D\pi(\theta)[e_i]) D\pi(\theta)[e_j], D\pi(\theta)[e_\ell] \rangle_{[\theta]} (C_e)_{k\ell}.$$

From there, we introduce the following linear map from and to symmetric matrices:

$$R_m: \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}: M \mapsto R_m(M), \text{ with} \quad (7.13)$$

$$(R_m(M))_{ij} = \sum_{k,\ell} \langle \bar{\mathcal{R}}(\mathrm{D}\pi(\theta)[e_k], \mathrm{D}\pi(\theta)[e_i])\mathrm{D}\pi(\theta)[e_j], \mathrm{D}\pi(\theta)[e_\ell] \rangle_{[\theta]} M_{k\ell}.$$

Theorem 7.5 (CRB on quotient manifolds, with curvature). (*Continued from Theorem 7.3*). Including terms due to the possible curvature of $\bar{\mathcal{P}}$, at large SNR, the covariance matrix C_e (7.6) of any unbiased estimator $\hat{\theta}: \mathcal{M} \rightarrow \bar{\mathcal{P}}$ and the Fisher information matrix F_e (7.1) w.r.t. the orthonormal basis e of $T_\theta \mathcal{P}$ obey the following matrix inequality (assuming $\mathrm{rank}(F_e) = \bar{d}$):

$$C_e \succeq F_e^\dagger - \frac{1}{3} (R_m(F_e^\dagger)F_e^\dagger + F_e^\dagger R_m(F_e^\dagger)),$$

where $R_m: \mathbb{R}^{d \times d} \rightarrow \mathbb{R}^{d \times d}$ is as defined by (7.13).

Proof. The proof is very similar to that of Theorem 7.4. We start from the CRB w.r.t. the basis \bar{e} (7.9). Expanding

$$\mathrm{D}\pi(\theta)[e_i] = \mathrm{D}\pi(\theta)[\mathrm{Proj}_\theta^h e_i] = \sum_j \langle \tilde{e}_j, e_i \rangle \mathrm{D}\pi(\theta)[\tilde{e}_j] = \sum_j E_{ji} \bar{e}_j$$

and exploiting linearity of $\langle \bar{\mathcal{R}}(\cdot, \cdot) \cdot, \cdot \rangle_{[\theta]}$ in its four arguments, relation (7.11) is established for the operators \bar{R}_m (7.12) and R_m (7.13) too. From the proof of Theorem 7.3, recall that $C_{\bar{e}} = EC_e E^\top$ and $\bar{F}_{\bar{e}}^{-1} = EF_e^\dagger E^\top$. The relation (7.11) yields $\bar{R}_m(\bar{F}_{\bar{e}}^{-1}) = ER_m(F_e^\dagger)E^\top$. Substituting in the CRB gives:

$$EC_e E^\top \succeq E \left(F_e^\dagger - \frac{1}{3} (R_m(F_e^\dagger)F_e^\dagger + F_e^\dagger R_m(F_e^\dagger)) \right) E^\top,$$

where we used the fact that $R_m(M)P_e = P_e R_m(M) = R_m(M)$, which is easily established from (7.11). Lemma 7.1 applies and concludes the proof, since $P_e F_e^\dagger P_e = F_e^\dagger$. \square

7.4 Example

We take a look at an example of the family of synchronization problems. In such problems, one considers a group \mathcal{G} and a set of N group elements

$g_1, \dots, g_N \in \mathcal{G}$. The g_i 's are to be estimated based on noisy measurements of group element ratios $g_i g_j^{-1}$. When \mathcal{G} has a manifold structure, that is, when it is a Lie group, synchronization falls within the spectrum of estimation on manifolds. We investigate synchronization on the group of translations \mathbb{R}^n , which makes for a simple geometry and helps fix ideas. The next chapter is devoted to synchronization on $\text{SO}(n)$, the group of rotations in \mathbb{R}^n . Synchronization problems illustrate how both theorems for submanifolds and quotient manifolds can apply to the same setting, with rich interpretation.

Let $\boldsymbol{\theta} = (\theta_1, \dots, \theta_N)$ be a vector of N unknown but deterministic points in \mathbb{R}^n . Those can be thought of as positions, states, opinions, etc. of N agents. Let us consider an undirected graph on N nodes with edge set \mathcal{E} , such that for each edge $\{i, j\} \in \mathcal{E}$ we have a noisy measurement of the relative state $h_{ij} = \theta_j - \theta_i + n_{ij}$, where the $n_{ij} \sim \mathcal{N}(0, \Sigma)$ are i.i.d. normally distributed noise vectors. By symmetry, $h_{ij} = -h_{ji}$, so $n_{ij} = -n_{ji}$. While it is important to assume independence of noise on distinct edges to keep the derivation simple, it is easy to relax the assumption that they have identical distributions. We assume identical distributions to keep the exposition simple.

The task is to estimate the θ_i 's from the h_{ij} 's, thus $\mathcal{P} = (\mathbb{R}^n)^N$, and we set out to derive CRB's for this problem. An alternative way of obtaining this result can be found in (Howard *et al.*, 2010). Decentralized algorithms to execute this synchronization can be found there and in (Russell *et al.*, 2011).

The log-likelihood function $L: \mathcal{P} \rightarrow \mathbb{R}$ reads, with $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \dots, \hat{\theta}_N)$ and dropping additive constants:

$$L(\hat{\boldsymbol{\theta}}) = \frac{1}{2} \sum_{i=1}^N \sum_{i \sim j} -\frac{1}{2} (h_{ij} - \hat{\theta}_j + \hat{\theta}_i)^\top \Sigma^{-1} (h_{ij} - \hat{\theta}_j + \hat{\theta}_i).$$

The inner summation is over the neighbors j of node i . The coefficient $1/2$ accounts for the fact that the two sums cover each edge twice.

In order to compute the FIM for this problem, we need to pick an orthonormal basis of $T_{\boldsymbol{\theta}} \mathcal{P} \equiv \mathcal{P}$. We choose the basis such that the first n vectors correspond to the canonical basis for the first copy of \mathbb{R}^n in \mathcal{P} , the next n vectors correspond to the canonical basis for the second copy of \mathbb{R}^n in \mathcal{P} , etc., totaling nN orthonormal basis vectors. The gradient of $L(\hat{\boldsymbol{\theta}})$ w.r.t. $\hat{\theta}_i$ in this basis is the following vector in \mathbb{R}^n :

$$\text{grad}_i L(\hat{\boldsymbol{\theta}}) = \sum_{i \sim j} \Sigma^{-1} (h_{ij} - \hat{\theta}_j + \hat{\theta}_i).$$

Hence, $\text{grad}_i L(\boldsymbol{\theta}) = \sum_{j \in V_i} \Sigma^{-1} n_{ij}$. The FIM F (7.1) is formed of $N \times N$

blocks of size $n \times n$. Due to independence of the n_{ij} 's and $n_{ij} = -n_{ji}$,

$$\mathbb{E}\{(\Sigma^{-1}n_{ij})(\Sigma^{-1}n_{k\ell})^\top\} = \Sigma^{-1}\mathbb{E}\{n_{ij}n_{k\ell}^\top\}\Sigma^{-1} = \begin{cases} \Sigma^{-1} & \text{if } (i,j) = (k,\ell), \\ -\Sigma^{-1} & \text{if } (i,j) = (\ell,k), \\ 0 & \text{otherwise.} \end{cases}$$

Hence, the $(i,j)^{\text{th}}$ block of F is given by (with d_i the degree of node i):

$$F_{ij} = \mathbb{E}\{\text{grad}_i L(\boldsymbol{\theta}) \cdot \text{grad}_j L(\boldsymbol{\theta})^\top\} = \begin{cases} d_i \Sigma^{-1} & \text{if } i = j, \\ -\Sigma^{-1} & \text{if } i \sim j, \\ 0 & \text{otherwise.} \end{cases}$$

The structure of the graph Laplacian is apparent. Let $D = \text{diag}(d_1, \dots, d_N)$ be the degree matrix and let A be the adjacency matrix of the measurement graph. The Laplacian $\mathcal{L} = D - A$ is tied to the FIM via:

$$F = \mathcal{L} \otimes \Sigma^{-1},$$

where \otimes denotes the Kronecker product.

Of course, since we only have relative measurements, we can only hope to recover the θ_i 's up to a global translation. And indeed, for every translation vector $t \in \mathbb{R}^n$, we have $L(\hat{\boldsymbol{\theta}}) = L(\hat{\boldsymbol{\theta}} + t)$, where $\hat{\boldsymbol{\theta}} + t \triangleq (\hat{\theta}_1 + t, \dots, \hat{\theta}_N + t)$. That is, all $\hat{\boldsymbol{\theta}} + t$ induce the same distribution of the measurements h_{ij} , and are thus indistinguishable. This is the root of the rank deficiency of the FIM. Surely, if the graph is connected, the all-ones vector $\mathbf{1}_N$ forms a basis of $\ker \mathcal{L}$. Consequently, $\ker F$ consists of all vectors of the form $\mathbf{1}_N \otimes t$, with arbitrary $t \in \mathbb{R}^n$. Naturally, these correspond to global translations by t .

To resolve this ambiguity, we can either add constraints, most naturally in the form of anchors, or work on the quotient space.

With anchors Let us consider $A \subset \{1, \dots, N\}$, $A \neq \emptyset$, such that all θ_i with $i \in A$ are known; these are anchors. The resulting parameter space $\bar{\mathcal{P}} = \{\hat{\boldsymbol{\theta}} \in \mathcal{P} : \hat{\theta}_i = \theta_i \forall i \in A\}$ is a Riemannian submanifold of \mathcal{P} . The orthogonal projector from $T_{\boldsymbol{\theta}}\mathcal{P}$ to $T_{\boldsymbol{\theta}}\bar{\mathcal{P}}$ simply sets all components of a tangent vector corresponding to anchored nodes to zero. Formally, $P = I_A \otimes I_n$, where I_A is a diagonal matrix of size N whose i^{th} diagonal entry is 1 if $i \notin A$ and 0 otherwise. It follows that $PFP = I_A \mathcal{L} I_A \otimes \Sigma^{-1} = \mathcal{L}_A \otimes \Sigma^{-1}$, with the obvious definition for \mathcal{L}_A : the Laplacian with rows and columns corresponding to anchored nodes forced to zero. $\bar{\mathcal{P}}$ is Euclidean, hence it is flat and its curvature tensor vanishes identically. Theorem 7.2 yields the anchored CRB for the covariance matrix C of an unbiased estimator on $\bar{\mathcal{P}}$:

$$\mathbb{E}\{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^\top\} \triangleq C \succeq \mathcal{L}_A^\dagger \otimes \Sigma. \quad (7.14)$$

We used the commutativity of the Kronecker product and pseudoinversion (Bernstein, 2009, Fact 7.4.32). This bound is easily interpreted in terms of individual nodes. Indeed, by definition, inequality (7.14) means that for all $x \in \mathbb{R}^{nN}$, $x^\top Cx \geq x^\top (\mathcal{L}_A^\dagger \otimes \Sigma)x$. In particular, setting $x = e_i \otimes e_k$ with e_i the i^{th} canonical basis vector of \mathbb{R}^N and e_k the k^{th} canonical basis vector of \mathbb{R}^n , we have:

$$\mathbb{E} \left\{ (\hat{\theta}_i - \theta_i)_k^2 \right\} \geq (\mathcal{L}_A^\dagger)_{ii} \cdot \Sigma_{kk}.$$

Summing over $k = 1 \dots n$, this translates into a lower bound on the variance for estimating the state of node i :

$$\mathbb{E} \left\{ \|\hat{\theta}_i - \theta_i\|^2 \right\} \geq (\mathcal{L}_A^\dagger)_{ii} \cdot \text{trace}(\Sigma).$$

This puts forward the importance of the diagonal of \mathcal{L}_A^\dagger , which captures the topology of the measurement graph and the anchor placement. Taking traces on both sides of (7.14), we obtain an inequality for the total variance:

$$\mathbb{E} \left\{ \text{dist}^2(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) \right\} = \mathbb{E} \left\{ \sum_{i \notin A} \|\hat{\theta}_i - \theta_i\|^2 \right\} \geq \text{trace}(\mathcal{L}_A^\dagger) \text{trace}(\Sigma).$$

Notice that it would have been simple to pick a new basis for $T_{\boldsymbol{\theta}} \bar{\mathcal{P}}$, but this would have required a renumbering of the rows and columns of the matrices appearing in the CRB. If the ambiguities are fixed not by adding anchors but, more generally, by adding one or more (for example) linear constraints of the form $a_1\theta_1 + \dots + a_N\theta_N = b$, it becomes less obvious how to pick a meaningful basis for $T_{\boldsymbol{\theta}} \bar{\mathcal{P}}$ without breaking symmetry. In comparison, the projection method used here will apply gracefully, preserving symmetry and row/column ordering in the CRB matrices.

Without anchors If there are no anchors, perhaps because there is no meaningful reference to begin with, we work on the quotient space $\bar{\mathcal{P}} = \mathcal{P}/\sim$, where $\boldsymbol{\theta} \sim \boldsymbol{\theta}'$ iff there exists a translation vector $t \in \mathbb{R}^n$ such that $\boldsymbol{\theta} = \boldsymbol{\theta}' + t$. The distance between the equivalence classes $[\boldsymbol{\theta}]$ and $[\boldsymbol{\theta}']$ on $\bar{\mathcal{P}}$ is the distance between their best aligned members, that is:

$$\text{dist}^2([\boldsymbol{\theta}], [\boldsymbol{\theta}']) = \min_{t \in \mathbb{R}^n} \sum_{i=1}^N \|\theta_i + t - \theta'_i\|^2.$$

The optimal t is easily seen to be $t = \frac{1}{N} \sum_{i=1}^N \theta'_i - \theta_i$, which amounts to aligning the centers of mass of $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$. Consequently, if we denote by $\boldsymbol{\theta}_c$

the centered version of $\boldsymbol{\theta}$ —i.e., $\boldsymbol{\theta}$ translated such that its center of mass is at the origin—we find that:

$$\text{dist}^2([\boldsymbol{\theta}], [\boldsymbol{\theta}']) = \text{dist}^2(\boldsymbol{\theta}_c, \boldsymbol{\theta}'_c) = \sum_{i=1}^N \|\boldsymbol{\theta}_{c,i} - \boldsymbol{\theta}'_{c,i}\|^2.$$

From the first equality, it follows that the mapping $[\boldsymbol{\theta}] \mapsto \boldsymbol{\theta}_c$ is an isometry between $\bar{\mathcal{P}}$ and a Euclidean space. We thus conclude that $\bar{\mathcal{P}}$ is a flat manifold and that its curvature tensor vanishes identically (Lee, 1997, Chap. 7). Theorem 7.3 and the fact that Kronecker product and pseudoinversion commute (Bernstein, 2009, Fact 7.4.32) then yield:

$$\begin{aligned} \mathbb{E} \left\{ (\hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c)(\hat{\boldsymbol{\theta}}_c - \boldsymbol{\theta}_c)^\top \right\} &\triangleq C \succeq \mathcal{L}^\dagger \otimes \Sigma, \text{ and} \\ \mathbb{E} \left\{ \sum_{i=1}^N \|\hat{\boldsymbol{\theta}}_{c,i} - \boldsymbol{\theta}_{c,i}\|^2 \right\} &\geq \text{trace}(\mathcal{L}^\dagger) \text{trace}(\Sigma). \end{aligned} \quad (7.15)$$

We now interpret the CRB (7.15). Because of the ambiguity in the anchor-free scenario, it does not make much sense to ask what the variance for estimating a specific state is going to be. Rather, one should establish bounds for the variance on estimating the relative state between two nodes, i and j . Let $x = (e_i - e_j) \otimes e_k$ with e_i, e_j the i^{th} and j^{th} canonical basis vectors of \mathbb{R}^N and e_k the k^{th} canonical basis vector of \mathbb{R}^n . Notice that x is a horizontal vector (its components sum to zero). Applying $x^\top \cdot x$ on both sides of (7.15) yields:

$$\mathbb{E} \left\{ ((\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_j) - (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j))_k^2 \right\} \geq (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j) \cdot \Sigma_{kk}.$$

Notice that there is no need to center $\hat{\boldsymbol{\theta}}$ nor $\boldsymbol{\theta}$ anymore, since the quantities involved are relative states. Summing over $k = 1 \dots n$ gives a lower bound on the variance for estimating the relative state between node i and node j :

$$\mathbb{E} \left\{ \|(\hat{\boldsymbol{\theta}}_i - \hat{\boldsymbol{\theta}}_j) - (\boldsymbol{\theta}_i - \boldsymbol{\theta}_j)\|^2 \right\} \geq (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j) \cdot \text{trace}(\Sigma).$$

A nice interpretation is now possible. Indeed, the quantity $(e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j)$ is well-known to correspond to the squared Euclidean commute time distance (ECTD) between nodes i and j (Saerens *et al.*, 2004). It is small if many short paths connect the two nodes and if those paths have edges with large weights which, in our case, means measurements of high quality. Furthermore, Saerens *et al.* (2004) show how one can produce an embedding of the nodes in, say, the plane such that two nodes are close-by if the ECTD separating them is small. This is done via a projection akin to PCA and is

an interesting visualization tool as it leads to a plot of the graph such that easily synchronizable nodes are clustered together. See also Section 8.7.2.

Notice that the bound without anchors has a very different interpretation than that of the bound one would obtain by artificially fixing an arbitrary node. Notice also that, since we did not need to switch to a different basis to obtain the bounds, regardless of which anchors we did or did not choose, it is always the same rows and columns of the matrices in the CRB's that refer to a specific node, which is rather convenient.

The maximum likelihood estimator in the absence of anchors is easily obtained as the minimum-norm solution to the problem $\max L(\hat{\theta})$ (which is concave, quadratic). This estimator is centered and we state without proof that it is efficient, i.e., its covariance is exactly $\mathcal{L}^\dagger \otimes \Sigma$. In the anchored case, the maximum likelihood estimator is conveniently obtained via quadratic programming.

For the sake of simplicity, we considered a connected graph. In general, the graph might be disconnected, and there would then be more ambiguity. It is obvious that, in general, there is an \mathbb{R}^n ambiguity for each connected component that does not include an anchor. The CRB's presented here can easily be derived to take care of this more general situation: one simply needs to redefine the equivalence relation \sim accordingly. This in turn leads to a new quotient space with an appropriate notion of distance and covariance. The theorems established in this paper apply seamlessly to this more general scenario.

7.5 Conclusions

We proposed four theorems that are meant to ease the use of the intrinsic CRB's developed in the previous chapter when the actual parameter space is a Riemannian submanifold or a Riemannian quotient manifold of a (usually more natural) parent space. We showed on a simple example how these theorems provide meaningful bounds for estimation problems with indeterminacies, whether these are dealt with by including prior knowledge in the form of constraints or by acknowledging the quotient nature of the parameter space. We also observed on these same examples that fixing indeterminacies by adding constraints results in different CRB's than if the quotient nature is acknowledged. In the next chapter, we derive CRB's for synchronization of rotations. The non-commutativity of rotations and the curvature of the space of rotations calls for a more delicate analysis. The CRB's will again be structured by the Laplacian of the measurement graph, calling for rich interpretations.

Chapter 8

Cramér-Rao bounds for synchronization of rotations

In this chapter, the intrinsic estimation theory tools developed so far are applied to synchronization of rotations, which was addressed in Chapter 5. Recall that this is the problem of estimating rotation matrices R_1, \dots, R_N from noisy measurements of relative rotations $R_i R_j^\top$. Motivated by its pervasiveness in applications, we propose a derivation and analysis of Cramér-Rao bounds for this estimation problem. Our results hold for rotations in the special orthogonal group (5.1) for arbitrary n and for a large family of practically useful noise models, of which the mixture of Langevin model used in Chapter 5 is a particular case. We will see that the topology of the measurement graph plays a key role in the CRB's, via its Laplacian.

Previous work

As discussed in Section 5.3 about the eigenvector method, Singer (2011) studies synchronization of phases, that is, rotations in the plane, and reflects upon the generic nature of synchronization as the task of estimating group elements g_1, \dots, g_N based on measurements of their ratios $g_i g_j^{-1}$. In that work, the author focuses on synchronization in the presence of many outliers and establishes that the eigenvector method is remarkably robust: for a complete measurement graph, if a fraction p of the measurements are perfect and the remaining measurements are random outliers, then it is sufficient to have $p > 1/\sqrt{N}$ to provide better-than-random estimators. Furthermore, as $p^2 N \rightarrow \infty$, the estimation error goes to zero. In further work, Bandeira

et al. (2013b) derive Cheeger-type inequalities for synchronization on the orthogonal group under adversarial noise and generalize the eigenvector method to rotations in \mathbb{R}^n , as we leveraged in Section 5.4.1.

Wang & Singer (2013) propose the robust algorithm for synchronization called LUD (for *least unsquared deviation*) which we described and compared against in Section 5.5. It is based on a convex relaxation of an L1 formulation of the synchronization problem and comes with exact and stable recovery guarantees under a large set of scenarios. In particular, the authors show that for the same perfect-or-outlier scenario as in (Singer, 2011), there exists a critical value $p_{\text{critical}}^{(n)}$ (less than 50% for $n = 2$ or 3) such that if the fraction of perfect measurements p exceeds $p_{\text{critical}}^{(n)}$, then LUD achieves *exact* recovery of the rotations. This remarkable feat can be put in perspective with the famous approximation results of the SDP relaxation of max-cut (Goemans & Williamson, 1995) and, indeed, the LUD relaxation bears some resemblance with the latter. The authors furthermore establish that if the good measurements are affected by noise, then the recovery is stable.

The analyses about both the eigenvector method and the LUD algorithm provide statements about the performance of two specific algorithms for synchronization. As such, they can be regarded as upper bounds on the estimation error one is entitled to expect from competing estimation algorithms. More fundamentally, they give insight into the complexity of synchronization tasks.

In comparison, the present chapter focuses on providing lower bounds on estimation error for synchronization or rotations. Such bounds constitute a benchmark for estimation algorithms, but more importantly provide further insight into the decisive features that make a synchronization task more or less difficult to solve. In particular, because we allow for arbitrary (but deterministic) measurement graph structures, our analysis sheds light on the role of the topology of said graph. The original analyses in (Singer, 2011) and (Wang & Singer, 2013) are limited to complete or random Erdős-Rényi graphs. Analyses by Bandeira *et al.* (2013b) and Demanet & Jugnon (2013) provide bounds for the eigenvector method with fixed graphs too, but under adversarial noise (worst-case analysis).

Barooah & Hespanha (2007) study the covariance of the BLUE estimator for synchronization on the group of translations \mathbb{R}^n , with anchors. This covariance coincides with the CRB under Gaussian noise and involves the Laplacian of the measurement graph as in Section 7.4. The authors give interpretations of the covariance in terms of the resistance distance on the measurement graph, similar to the interpretations in this chapter for the anchored case.

Howard *et al.* (2010) study synchronization on the group of translations

\mathbb{R}^n and on the group of phases $\text{SO}(2)$. They establish CRB's for synchronization in the presence of Gaussian-like noise on these groups and provide decentralized algorithms to solve synchronization. Their derivation of the CRB's is limited to Gaussian-like noise and seems to rely heavily on the commutativity (and thus flatness) of \mathbb{R}^n and $\text{SO}(2)$, and hence does not apply to synchronization on $\text{SO}(n)$ in general. The present chapter can be considered a broad generalization of that work, using different tools.

Other authors have established CRB's for the related sensor network localization problem (SNL). Ash & Moses (2007) and Chang & Sahai (2006) among others study SNL based on inter-agent distance measurements, and notably give an interpretation of the CRB in the absence of anchors. A remarkable fact is that, for all these problems of estimation on graphs, the pseudoinverse of the graph Laplacian plays a fundamental role in the CRB—although not all authors explicitly reflect on this. As we shall see, this special structure is rich in interpretations, many of which exceed the context of synchronization of rotations specifically.

Contributions and outline

In this chapter, we first restate the problem of synchronization of rotations similarly to the presentation of Chapter 5, but with an emphasis on accommodating a large family of noise models rather than the specific mixture-of-Langevin model—Section 8.1. This estimation problem is stated on a manifold. In the presence of anchors, this manifold has a Riemannian submanifold geometry which was described in Section 5.2. When no anchors are known, the parameter space has a Riemannian quotient manifold geometry which we describe in Section 8.2.

We then spend some time studying probability density functions (pdf) on $\text{SO}(n)$ and exploring the family of noise models concerned by our analysis in Section 8.3. We show that this family is both useful for applications (it essentially contains zero-mean, isotropic noise models) and practical to work with (the expectations one is led to compute via integrals on $\text{SO}(n)$ are easily converted into classical integrals on \mathbb{R}^n). In particular, this family includes heavy-tailed distributions on $\text{SO}(n)$ which can prove useful generically for estimation problems on $\text{SO}(n)$ with outliers.

In Section 8.4, we derive the Fisher information matrix (FIM) for synchronization and establish that it is structured by the Laplacian of the measurement graph, where edge weights are proportional to the quality of their respective measurements. The FIM plays a central role in the CRB's we establish for anchored and anchor-free synchronization in Section 8.5. The main tools used to that effect are intrinsic versions of the CRB's, as developed in Chapter 7. The CRB's are structured by the pseudoinverse of

the Laplacian of the measurement graph. We derive clear interpretations of these bounds in terms of random walks, both with and without anchors.

As a main result for anchored synchronization, we show that for any unbiased estimator \hat{R}_i of the rotation R_i , asymptotically for small errors,

$$\mathbb{E} \left\{ \text{dist}^2(R_i, \hat{R}_i) \right\} \geq d^2 (\mathcal{L}_A^\dagger)_{ii},$$

where $\text{dist}(R_i, \hat{R}_i) = \|\log(R_i^\top \hat{R}_i)\|_F$ is the geodesic distance on $\text{SO}(n)$, $d = n(n - 1)/2$, \mathcal{L}_A is the Laplacian of the weighted measurement graph with rows and columns corresponding to anchors set to zero and \dagger denotes the Moore-Penrose pseudoinverse—see (8.30). The better a measurement is, the larger the weight on the associated edge is—see (8.22). This bound holds in a small-error regime under the assumption that noise on different measurements is independent, that the measurements are isotropically distributed around the true relative rotations and that there is at least one anchor in each connected component of the graph. The right-hand side of this inequality is zero if node i is an anchor, and is small if node i is strongly connected to anchors. More precisely, it is proportional to the ratio between the average number of times a random walker starting at node i will be at node i before hitting an anchored node and the total amount of information available in measurements involving node i .

As a main result for anchor-free synchronization, we show that for any unbiased estimator $\hat{R}_i \hat{R}_j^\top$ of the relative rotation $R_i R_j^\top$, asymptotically for small errors,

$$\mathbb{E} \left\{ \text{dist}^2(R_i R_j^\top, \hat{R}_i \hat{R}_j^\top) \right\} \geq d^2 (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j),$$

where \mathcal{L} is the Laplacian of the weighted measurement graph and e_i is the i^{th} column of the $N \times N$ identity matrix—see (8.35). This bound holds in a small-error regime under the assumption that noise on different measurements is independent, that the measurements are isotropically distributed around the true relative rotations and that the measurement graph is connected. The right-hand side of this inequality is proportional to the squared Euclidean commute time distance (ECTD) (Saerens *et al.*, 2004) on the weighted graph. It measures how strongly nodes i and j are connected. More explicitly, it is proportional to the average time a random walker starting at node i walks before hitting node j and then node i again.

Section 8.7 hosts a few comments on the CRB's. In particular, a PCA-like visualization tool is detailed, a link with the Fiedler value of the graph is described and the robustness of synchronization versus outliers is confirmed, via arguments that differ from those in (Singer, 2011).

8.1 A family of noise models

The target quantities (the parameters) are the rotation matrices R_1, \dots, R_N in $\text{SO}(n)$. The natural parameter space is thus:

$$\mathcal{P} = \text{SO}(n) \times \cdots \times \text{SO}(n) \quad (\text{N copies}). \quad (8.1)$$

For each edge $\{i, j\}$ in the measurement graph (5.2), a measurement (5.3)

$$H_{ij} = Z_{ij} R_i R_j^\top \quad (8.2)$$

is available, where Z_{ij} is a random variable distributed over $\text{SO}(n)$ following a probability density function (pdf)

$$f_{ij}: \text{SO}(n) \rightarrow \mathbb{R}^+$$

with respect to the Haar measure μ on $\text{SO}(n)$ —see Section 8.3. We say that the measurement is *unbiased*, or that the noise has *zero-mean*, if H_{ij} is an unbiased estimator of $R_i R_j^\top$, that is, the expectation of $\log(Z_{ij})$ is zero. We also say that noise is *isotropic* if its probability density function is only a function of distance to the identity. Different notions of distance on $\text{SO}(n)$ yield different notions of isotropy. In Section 8.3 we give a few examples of useful zero-mean, isotropic distributions on $\text{SO}(n)$.

By symmetry, define $H_{ji} = Z_{ji} R_j R_i^\top = H_{ij}^\top$ and the random variable Z_{ji} and its density f_{ji} are defined accordingly in terms of f_{ij} and Z_{ij} . In particular,

$$Z_{ji} = R_j R_i^\top Z_{ij}^\top R_i R_j^\top, \text{ and} \quad f_{ij}(Z_{ij}) = f_{ji}(Z_{ji}).$$

The pdf's f_{ij} and f_{ji} are linked as such because the Haar measure μ is invariant under the change of variable relating Z_{ij} and Z_{ji} .

In this work, we restrict our attention to noise models that fulfill the three following assumptions:

Assumption 8.1 (smoothness and support). *Each pdf f_{ij} is a smooth, positive function.*

Assumption 8.2 (independence). *The Z_{ij} 's associated to different edges of the measurement graph are independent random variables. That is, if $\{i, j\} \neq \{p, q\}$, then Z_{ij} and Z_{pq} are independent.*

Assumption 8.3 (invariance). *Each pdf f_{ij} is invariant under orthogonal conjugation, that is, $\forall Z \in \text{SO}(n), \forall Q \in \text{O}(n), f_{ij}(QZQ^\top) = f_{ij}(Z)$. We say f_{ij} is a spectral function, since it only depends on the eigenvalues of its argument. The eigenvalues of matrices in $\text{SO}(2k)$ have the form $e^{\pm i\theta_1}, \dots, e^{\pm i\theta_k}$, with $0 \leq \theta_1, \dots, \theta_k \leq \pi$. The eigenvalues of matrices in $\text{SO}(2k+1)$ have an additional eigenvalue 1.*

Assumption 8.1 is satisfied for all the noise models we consider; it could be relaxed to some extent but would make some of the proofs more technical. Assumption 8.2 is admittedly a strong restriction but is necessary to make the joint pdf of the whole estimation problem easy to derive, leading to an easy expression for the log-likelihood function. As we will see in Section 8.4, it is also at the heart of the nice Laplacian structure of the Fisher information matrix. Assumption 8.3 is a technical condition that will prove useful in many respects. One of them is the observation that pdf's which obey Assumption 8.3 are easy to integrate over $\text{SO}(n)$. We expand on this in Section 8.3, where we also show that a large family of interesting pdf's satisfy these assumptions, namely, zero-mean isotropic distributions.

Under Assumption 8.2, the *log-likelihood* of an estimator $\hat{\mathbf{R}} \in \mathcal{P}$, given the measurements H_{ij} , is given by:

$$L(\hat{\mathbf{R}}) = \sum_{i \sim j} \log f_{ij}(H_{ij} \hat{R}_j \hat{R}_i^\top) = \frac{1}{2} \sum_{i=1}^N \sum_{i \sim j} \log f_{ij}(H_{ij} \hat{R}_j \hat{R}_i^\top), \quad (8.3)$$

where the first $i \sim j$ summation is over all edges $\{i, j\}$ and the second is over the neighbors j of each node i . The coefficient $1/2$ reflects the fact that in the second form each measurement is counted twice. Under Assumption 8.1, L is a smooth function on the smooth manifold \mathcal{P} .

The log-likelihood function is invariant under a global rotation. Indeed,

$$\forall \hat{\mathbf{R}} \in \mathcal{P}, \forall Q \in \text{SO}(n), \quad L(\hat{\mathbf{R}}Q) = L(\hat{\mathbf{R}}),$$

where $\hat{\mathbf{R}}Q$ denotes $(\hat{R}_1 Q, \dots, \hat{R}_N Q) \in \mathcal{P}$. This invariance encodes the fact that all sets of rotations of the form $\hat{\mathbf{R}}Q$ yield the same distribution of the measurements H_{ij} , and are hence equally likely estimators.

To resolve the ambiguity, one can follow at least two courses of action. One is to include additional constraints, most naturally in the form of anchors, i.e., assume some of the rotations are known.¹ The other is to acknowledge the invariance by working on the associated quotient space. Following the first path, the parameter space becomes \mathcal{P}_A , a Riemannian submanifold of \mathcal{P} described in Section 5.2. Following the second path, the parameter space becomes \mathcal{P}_\emptyset , a Riemannian quotient manifold of \mathcal{P} described in the next section.

Remark 8.1 (A word about other noise models). *We show that measurements of the form $H_{ij} = Z_{ij,1} R_i R_j^\top Z_{ij,2}$, with $Z_{ij,1}$ and $Z_{ij,2}$ two random rotations with pdf's satisfying Assumptions 8.1 and 8.3, satisfy the*

¹If we only know that R_i is close to some matrix \bar{R} , and not necessarily equal to it, we may add a phony node R_{N+1} anchored at \bar{R} , and link that node and R_i with a high confidence measure $H_{i,N+1} = I_n$. This makes it possible to have “soft anchors”.

noise model considered in the present work. In doing so, we use some material from Section 8.3. For notational convenience, let us consider $H = Z_1 R Z_2$, with Z_1, Z_2 two random rotations with pdf's f_1, f_2 satisfying Assumptions 8.1 and 8.3, $R \in \mathrm{SO}(n)$ fixed. Then, the pdf of H is the function $h: \mathrm{SO}(n) \rightarrow \mathbb{R}^+$ given by (essentially) the convolution of f_1 and f_2 on $\mathrm{SO}(n)$:

$$h(H) = \int_{\mathrm{SO}(n)} f_1(Z) f_2(R^\top Z^\top H) d\mu(Z) = \int_{\mathrm{SO}(n)} f_1(Z) f_2(Z^\top H R^\top) d\mu(Z),$$

where we used that f_2 is spectral: $f_2(R^\top Z^\top H) = f_2(R R^\top Z^\top H R^\top)$. Let Z_{eq} be a random rotation with smooth pdf f_{eq} . We will shape f_{eq} such that the random rotation $Z_{\text{eq}} R$ has the same distribution as H . This condition can be written as follows: for all measurable subsets $S \subset \mathrm{SO}(n)$,

$$\int_S h(Z) d\mu(Z) = \int_{S R^\top} f_{\text{eq}}(Z) d\mu(Z) = \int_S f_{\text{eq}}(Z R^\top) d\mu(Z),$$

where, going from the second to the third integral, we used the change of variable $Z := Z R^\top$ and the bi-invariance of the Haar measure μ . In words: for all S , the probability that H belongs to S must be the same as the probability that $Z_{\text{eq}} R$ belongs to S . This must hold for all S , hence $f_{\text{eq}}(Z_{\text{eq}} R^\top) = h(Z_{\text{eq}})$, or equivalently:

$$f_{\text{eq}}(Z_{\text{eq}}) = h(Z_{\text{eq}} R) = \int_{\mathrm{SO}(n)} f_1(Z) f_2(Z^\top Z_{\text{eq}}) d\mu(Z).$$

This uniquely defines the pdf of Z_{eq} . It remains to show that f_{eq} is a spectral function. For all $Q \in \mathrm{O}(n)$,

$$\begin{aligned} f_{\text{eq}}(Q Z_{\text{eq}} Q^\top) &= \int_{\mathrm{SO}(n)} f_1(Z) f_2(Z^\top Q Z_{\text{eq}} Q^\top) d\mu(Z) \\ (\text{f_2 is spectral}) &= \int_{\mathrm{SO}(n)} f_1(Z) f_2(Q^\top Z^\top Q Z_{\text{eq}}) d\mu(Z) \\ (\text{change of variable: $Z := Q Z Q^\top$}) &= \int_{\mathrm{SO}(n)} f_1(Q Z Q^\top) f_2(Z^\top Z_{\text{eq}}) d\mu(Z) \\ (\text{f_1 is spectral}) &= \int_{\mathrm{SO}(n)} f_1(Z) f_2(Z^\top Z_{\text{eq}}) d\mu(Z) \\ &= f_{\text{eq}}(Z_{\text{eq}}). \end{aligned}$$

Hence, the noise model $H_{ij} = Z_{ij,1} R_i R_j^\top Z_{ij,2}$ can be replaced with the model $H_{ij} = Z_{ij,\text{eq}} R_i R_j^\top$ and the pdf of $Z_{ij,\text{eq}}$ is such that it falls within the scope of the present work.

In particular, if f_1 is a point mass at the identity, so that $H = RZ_2$ (noise multiplying the relative rotation on the right rather than on the left), $f_{\text{eq}} = f_2$, so that it does not matter whether we consider $H_{ij} = Z_{ij}R_iR_j^\top$ or $H_{ij} = R_iR_j^\top Z_{ij}$: they have the same distribution.

8.2 Geometry of the parameter space, without anchors

When no anchors are provided, the distribution of the measurements H_{ij} (8.2) is the same whether the true rotations are \mathbf{R} or $\mathbf{R}Q$, regardless of $Q \in \text{SO}(n)$. Consequently, the measurements contain no information as to which of those sets of rotations is the right one. This leads to the definition of the equivalence relation \sim over \mathcal{P} (8.1):

$$\mathbf{R} \sim \mathbf{R}' \Leftrightarrow \exists Q \in \text{SO}(n) : \mathbf{R} = \mathbf{R}'Q. \quad (8.4)$$

This equivalence relation partitions \mathcal{P} into equivalence classes, often called *fibers*. The quotient space (the set of equivalence classes)

$$\mathcal{P}_\emptyset \triangleq \mathcal{P} / \sim \quad (8.5)$$

is again a smooth manifold (in fact, \mathcal{P}_\emptyset is a *coset manifold* because it results from the quotient of the Lie group \mathcal{P} by a closed subgroup of \mathcal{P} (O'Neill, 1983, Prop. 11.12)). See Sections 2.2.2, 2.3.2 and 2.4.2 for background on quotient manifolds, which we use now. The notation \mathcal{P}_\emptyset reminds us that the set of anchors A is empty. Naturally, the log-likelihood function L (8.3) is constant over equivalence classes and hence descends as a well-defined function on \mathcal{P}_\emptyset .

Each fiber

$$[\mathbf{R}] = \{\mathbf{R}Q : Q \in \text{SO}(n)\} \in \mathcal{P}_\emptyset$$

is a Riemannian submanifold of the total space \mathcal{P} . As such, at each point \mathbf{R} , the fiber $[\mathbf{R}]$ admits a tangent space that is a subspace of $T_{\mathbf{R}}\mathcal{P}$. That tangent space to the fiber is called the *vertical space* at \mathbf{R} , noted $V_{\mathbf{R}}$. Vertical vectors point along directions that are parallel to the fibers. Vectors orthogonal, in the sense of the Riemannian metric (5.12), to all vertical vectors form the *horizontal space* $H_{\mathbf{R}} = (V_{\mathbf{R}})^\perp$, such that the tangent space $T_{\mathbf{R}}\mathcal{P}$ is equal to the direct sum $V_{\mathbf{R}} \oplus H_{\mathbf{R}}$. Horizontal vectors are orthogonal to the fibers, hence point toward the other fibers, i.e., the other points on the quotient space \mathcal{P}_\emptyset . See Figures 2.3 and 7.3 for an illustration.

Because \mathcal{P}_\emptyset is a coset manifold, the projection

$$\pi : \mathcal{P} \rightarrow \mathcal{P}_\emptyset : \mathbf{R} \mapsto \pi(\mathbf{R}) = [\mathbf{R}] \quad (8.6)$$

is a *submersion*. That is, the restricted differential $D\pi|_{H_{\mathbf{R}}}$ is a full-rank linear map between $H_{\mathbf{R}}$ and $T_{[\mathbf{R}]} \mathcal{P}_\emptyset$. Practically, this means that the horizontal space $H_{\mathbf{R}}$ is naturally identified to the (abstract) tangent space $T_{[\mathbf{R}]} \mathcal{P}_\emptyset$. This results in a practical means of representing abstract vectors of $T_{[\mathbf{R}]} \mathcal{P}_\emptyset$ simply as vectors of $H_{\mathbf{R}} \subset T_{\mathbf{R}} \mathcal{P}$, where \mathbf{R} is any arbitrarily chosen member of $[\mathbf{R}]$. Each horizontal vector $\xi_{\mathbf{R}}$ is unambiguously related to its abstract counterpart $\xi_{[\mathbf{R}]}$ in $T_{[\mathbf{R}]} \mathcal{P}_\emptyset$ via

$$\xi_{[\mathbf{R}]} = D\pi(\mathbf{R})[\xi_{\mathbf{R}}].$$

The representation $\xi_{\mathbf{R}}$ of $\xi_{[\mathbf{R}]}$ is called the *horizontal lift* of $\xi_{[\mathbf{R}]}$ at \mathbf{R} .

Consider $\xi_{[\mathbf{R}]}$ and $\eta_{[\mathbf{R}]}$, two tangent vectors at $[\mathbf{R}]$. Let $\xi_{\mathbf{R}}$ and $\eta_{\mathbf{R}}$ be their horizontal lifts at $\mathbf{R} \in [\mathbf{R}]$ and let $\xi_{\mathbf{R}'}$ and $\eta_{\mathbf{R}'}$ be their horizontal lifts at $\mathbf{R}' \in [\mathbf{R}]$. The Riemannian metric on \mathcal{P} (5.12) is such that $\langle \xi_{\mathbf{R}}, \eta_{\mathbf{R}} \rangle_{\mathbf{R}} = \langle \xi_{\mathbf{R}'}, \eta_{\mathbf{R}'} \rangle_{\mathbf{R}'}$. This motivates the definition of the metric

$$\langle \xi_{[\mathbf{R}]}, \eta_{[\mathbf{R}]} \rangle_{[\mathbf{R}]} = \langle \xi_{\mathbf{R}}, \eta_{\mathbf{R}} \rangle_{\mathbf{R}}$$

on \mathcal{P}_\emptyset , which is then well defined (it does not depend on the choice of \mathbf{R} in $[\mathbf{R}]$) and turns the restricted differential $D\pi(\mathbf{R}): H_{\mathbf{R}} \rightarrow T_{[\mathbf{R}]} \mathcal{P}_\emptyset$ into an isometry. This is a Riemannian metric and it is the only such metric such that π (8.6) is a *Riemannian submersion* from \mathcal{P} to \mathcal{P}_\emptyset (Gallot *et al.*, 2004, Prop. 2.28). Hence, \mathcal{P}_\emptyset is a *Riemannian quotient manifold* of \mathcal{P} .

We now describe the vertical and horizontal spaces of \mathcal{P} w.r.t. the equivalence relation (8.4). Let $\mathbf{R} \in \mathcal{P}$ and $Q : \mathbb{R} \rightarrow SO(n) : t \mapsto Q(t)$ such that Q is smooth and $Q(0) = I$. Then, the derivative $Q'(0) = \Omega$ is some skew-symmetric matrix in $\mathfrak{so}(n)$. Since $\mathbf{R}Q(t) \in [\mathbf{R}]$ for all t , it follows that $\frac{d}{dt} \mathbf{R}Q(t)|_{t=0} = \mathbf{R}\Omega$ is a tangent vector to the fiber $[\mathbf{R}]$ at \mathbf{R} , i.e., it is a vertical vector at \mathbf{R} . All vertical vectors have such form, hence:

$$V_{\mathbf{R}} = \{\mathbf{R}\Omega : \Omega \in \mathfrak{so}(n)\}.$$

A horizontal vector $\mathbf{R}\Omega = (R_1\Omega_1, \dots, R_N\Omega_N) \in H_{\mathbf{R}}$ is orthogonal to all vertical vectors, i.e., $\forall \Omega \in \mathfrak{so}(n)$,

$$0 = \langle \mathbf{R}\Omega, \mathbf{R}\Omega \rangle = \left\langle \sum_{i=1}^N \Omega_i, \Omega \right\rangle.$$

Since this is true for all skew-symmetric matrices Ω , we find that the horizontal space is defined as:

$$H_{\mathbf{R}} = \{\mathbf{R}\Omega : \Omega_1, \dots, \Omega_N \in \mathfrak{so}(n) \text{ and } \sum_{i=1}^N \Omega_i = 0\}.$$

This is not surprising: vertical vectors move all rotations in the same direction, remaining in the same equivalence class, whereas horizontal vectors move away toward other equivalence classes.

We now define the logarithmic map on \mathcal{P}_\emptyset , see Definition 2.26. Considering two points $[\mathbf{R}], [\hat{\mathbf{R}}] \in \mathcal{P}_\emptyset$, the logarithm $\text{Log}_{[\mathbf{R}]}([\hat{\mathbf{R}}])$ is the smallest tangent vector in $T_{[\mathbf{R}]} \mathcal{P}_\emptyset$ that brings us from the first equivalence class to the other through the exponential map. In other words: it is the error vector of $[\hat{\mathbf{R}}]$ in estimating $[\mathbf{R}]$. Working with the horizontal lift representation

$$D\pi(\mathbf{R})|_{H_{\mathbf{R}}}^{-1}[\text{Log}_{[\mathbf{R}]}([\hat{\mathbf{R}}])] = (R_1 \Omega_1, \dots, R_N \Omega_N) \in H_{\mathbf{R}}, \quad (8.7)$$

the Ω_i 's are skew-symmetric matrices solution of:

$$\min_{\Omega_i \in \mathfrak{so}(n), Q \in \text{SO}(n)} \|\Omega_1\|_F^2 + \dots + \|\Omega_N\|_F^2,$$

$$\text{such that } R_i \exp(\Omega_i) = \hat{R}_i Q, i = 1 \dots N, \text{ and}$$

$$\Omega_1 + \dots + \Omega_N = 0.$$

The rotation Q sweeps through all members of the equivalence class $[\hat{\mathbf{R}}]$ in search of the one closest to \mathbf{R} . By substituting $\Omega_i = \log(R_i^\top \hat{R}_i Q)$ in the objective function, we find that the objective value as a function of Q is

$$\sum_{i=1 \dots N} \|\log(R_i^\top \hat{R}_i Q)\|_F^2.$$

Critical points of this function w.r.t. Q verify $\sum_{i=1}^N \Omega_i = 0$, hence we need not enforce the last constraint: all candidate solutions are horizontal vectors. Summing up, we find that the squared geodesic distance on \mathcal{P}_\emptyset obeys:

$$\text{dist}^2([\mathbf{R}], [\hat{\mathbf{R}}]) = \min_{Q \in \text{SO}(n)} \sum_{i=1}^N \|\log(R_i^\top \hat{R}_i Q)\|_F^2. \quad (8.8)$$

Since $\text{SO}(n)$ is compact, this is a well-defined quantity. Let $Q \in \text{SO}(n)$ be one of the global minimizers. Then, an acceptable value for the logarithmic map is

$$D\pi(\mathbf{R})|_{H_{\mathbf{R}}}^{-1}[\text{Log}_{[\mathbf{R}]}([\hat{\mathbf{R}}])] = \left(R_1 \log(R_1^\top \hat{R}_1 Q), \dots, R_N \log(R_N^\top \hat{R}_N Q) \right).$$

Under reasonable proximity conditions on $[\mathbf{R}]$ and $[\hat{\mathbf{R}}]$, the global maximizer Q is uniquely defined, and hence so is the logarithmic map. An optimal Q is a *Karcher mean*—or *intrinsic mean* or *Riemannian center of mass*—of the rotation matrices $\hat{R}_1^\top R_1, \dots, \hat{R}_N^\top R_N$. Hartley *et al.* (2013), among others, give a thorough overview of algorithms to compute such means as well as uniqueness conditions.

8.3 Measures, integrals and distributions on $\mathrm{SO}(n)$

To define a noise model for the synchronization measurements (8.2), we now cover a notion of probability density function (pdf) over $\mathrm{SO}(n)$ and give a few examples of useful pdf's.

Being a compact Lie group, $\mathrm{SO}(n)$ admits a unique bi-invariant Haar measure μ such that $\mu(\mathrm{SO}(n)) = 1$ (Boothby, 1986, Thm 3.6, p. 247). Such a measure verifies, for all measurable subsets $S \subset \mathrm{SO}(n)$ and for all $L, R \in \mathrm{SO}(n)$, that $\mu(LSR) = \mu(S)$, where $LSR \triangleq \{LQR : Q \in S\} \subset \mathrm{SO}(n)$. That is, the measure of a portion of $\mathrm{SO}(n)$ is invariant under left and right actions of $\mathrm{SO}(n)$. We will need something slightly more general.

Lemma 8.1 (extended bi-invariance). $\forall L, R \in \mathrm{O}(n)$ such that $\det(LR) = 1$, $\forall S \subset \mathrm{SO}(n)$ measurable, $\mu(LSR) = \mu(S)$ holds.

Proof. LSR is still a measurable subset of $\mathrm{SO}(n)$. Let μ' denote the Haar measure on $\mathrm{O}(n) \supset \mathrm{SO}(n)$. The restriction of μ' to the measurables of $\mathrm{SO}(n)$ is still a Haar measure. By the uniqueness of the Haar measure up to multiplicative constant, there exists $\alpha > 0$ such that for all measurable subsets $T \subset \mathrm{SO}(n)$, we have $\mu(T) = \alpha\mu'(T)$. Then, $\mu(LSR) = \alpha\mu'(LSR) = \alpha\mu'(S) = \mu(S)$. \square

For the notion of Lebesgue integral associated with μ , Lemma 8.1 translates into the following statement, with $f: \mathrm{SO}(n) \rightarrow \mathbb{R}$ an integrable function:

$$\forall L, R \in \mathrm{O}(n) \text{ s.t. } \det(LR) = 1,$$

$$\int_{\mathrm{SO}(n)} f(LZR) d\mu(Z) = \int_{\mathrm{SO}(n)} f(Z) d\mu(Z). \quad (8.9)$$

This property will play an important role in the sequel.

A pdf on $\mathrm{SO}(n)$ is a nonnegative measurable function f on $\mathrm{SO}(n)$ such that

$$\mu(f) = \int_{\mathrm{SO}(n)} f(Z) d\mu(Z) = 1.$$

In this work, for convenience, we further assume pdf's are smooth and positive (Assumption 8.1) to make free use of the derivatives of their logarithm. Owing to Assumption 8.3, it further holds that pdf's in this work are *class functions* (Definition A.1). Appendix A details how this property helps reduce integrals over $\mathrm{SO}(n)$ to classical integrals, thus making them accessible analytically. This is done using the *Weyl integration formula*.

Example 8.1 (uniform). *The pdf associated with the uniform distribution is $f \equiv 1$.*

Example 8.2 (isotropic Langevin). *Recall the pdf for an isotropic Langevin distribution on $\text{SO}(n)$ with mean I_n and concentration $\kappa \geq 0$ (5.4):*

$$f(Z) = \ell_\kappa(Z) = \frac{1}{c_n(\kappa)} \exp(\kappa \text{trace}(Z)),$$

where $c_n(\kappa)$ is a normalization constant such that f has unit mass:

$$c_n(\kappa) = \int_{\text{SO}(n)} \exp(\kappa \text{trace}(Z)) d\mu(Z). \quad (8.10)$$

As per Weyl's integration formulas, the following can be derived:

$$c_2(\kappa) = I_0(2\kappa), \quad (8.11)$$

$$c_3(\kappa) = \exp(\kappa)(I_0(2\kappa) - I_1(2\kappa)), \quad (8.12)$$

$$c_4(\kappa) = I_0(2\kappa)^2 - 2I_1(2\kappa)^2 + I_0(2\kappa)I_2(2\kappa), \quad (8.13)$$

in terms of the modified Bessel functions of the first kind, I_ν (A.4). See Appendix A for details.

For $n = 2$, the Langevin distribution is also known as the von Mises or Fisher distribution on the circle (Mardia & Jupp, 2000). The Langevin distribution on $\text{SO}(n)$ also exists in anisotropic form (Chiuso et al., 2008). Unfortunately, the associated pdf is no longer a spectral function, which is an instrumental property in the present work. Consequently, we do not treat anisotropic distributions. Chikuse gives an in-depth treatment of statistics on the Grassmann and Stiefel manifolds (Chikuse, 2003), including a study of Langevin distributions on $\text{SO}(n)$ as a special case.

The set of pdf's is closed under convex combinations, as is the set of functions satisfying Assumptions 8.1 and 8.3. Thus, the mixture of Langevin model from Chapter 5 falls in the scope of the present chapter.

Example 8.3 (isotropic mixture of Langevin). *Recall the definition (5.5):*

$$f(Z) = p\ell_\kappa(Z) + (1-p)\ell_{\kappa'}(Z).$$

To conclude this section, we remark more broadly that all isotropic distributions around the identity matrix have a spectral pdf. Indeed, let $f: \text{SO}(n) \rightarrow \mathbb{R}$ be isotropic w.r.t. $\text{dist}(R_1, R_2) = \|\log(R_1^\top R_2)\|_{\text{F}}$ (5.11), the geodesic distance on $\text{SO}(n)$. That is, there is a function \tilde{f} such that $f(Z) = \tilde{f}(\text{dist}(I, Z)) = \tilde{f}(\|\log Z\|_{\text{F}})$. It is then obvious that $f(QZQ^\top) = f(Z)$ for all

$Q \in O(n)$ since $\log(QZQ^\top) = Q\log(Z)Q^\top$. The same holds for the embedded distance $\text{dist}(R_1, R_2) = \|R_1 - R_2\|_F$. This shows that the assumptions proposed in Section 8.1 include many interesting distributions.

Similarly we establish that all spectral pdf's have zero bias around the identity matrix I . The bias is the tangent vector (skew-symmetric matrix) $\Omega = \mathbb{E}\{\text{Log}_I(Z)\}$, with $Z \sim f$, f spectral. Since $\text{Log}_I(Z) = \log(Z)$ (5.10), we find, with a change of variable $Z := QZQ^\top$ going from the first to the second integral, that for all $Q \in O(n)$:

$$\Omega = \int_{SO(n)} \log(Z) f(Z) d\mu(Z) = \int_{SO(n)} \log(QZQ^\top) f(Z) d\mu(Z) = Q\Omega Q^\top.$$

Since skew-symmetric matrices are normal matrices and since Ω and $\Omega^\top = -\Omega$ have the same eigenvalues, we may choose $Q \in O(n)$ such that $Q\Omega Q^\top = -\Omega$. Therefore, $\Omega = -\Omega = 0$. As a consequence, it is only possible to treat unbiased measurements under the assumptions we make in this paper.

8.4 The Fisher information matrix

The relative rotation measurements $H_{ij} = Z_{ij}R_iR_j^\top$ (8.2) reveal information about the sought rotations R_1, \dots, R_N . The Fisher information matrix (FIM) encodes how much information these measurements contain on average. In other words, the FIM is an assessment of the quality of the measurements we have at our disposal for the purpose of estimating the sought parameters. The FIM will be instrumental in deriving CRB's in the next section.

Much of the technicalities involved in computing the FIM originate in the non-commutativity of rotations. It is helpful and informative to first go through this section with the special case $SO(2)$ in mind. Doing so, rotations commute and the space of rotations has dimension $d = 1$, so that one can reach the final result more directly.

Recall Definition 6.3 for the FIM. We first derive the gradient of the log-likelihood function L (8.3), $\text{grad } L(\hat{\mathbf{R}})$, a tangent vector in $T_{\hat{\mathbf{R}}} \mathcal{P}$. The i^{th} component of this gradient, that is, the gradient of the mapping $\hat{R}_i \mapsto L(\hat{\mathbf{R}})$ with $\hat{R}_{j \neq i}$ fixed, is a vector field on $SO(n)$ which can be written as:

$$\text{grad}_i L(\hat{\mathbf{R}}) = \sum_{i \sim j} \left[\text{grad} \log f_{ij}(H_{ij}\hat{R}_j\hat{R}_i^\top) \right]^\top H_{ij}\hat{R}_j.$$

Evaluated at the true rotations \mathbf{R} , this component becomes

$$\text{grad}_i L(\mathbf{R}) = \sum_{i \sim j} [\text{grad} \log f_{ij}(Z_{ij})]^\top Z_{ij}R_i.$$

The vector field $\text{grad } \log f_{ij}$ on $\text{SO}(n)$ may be factored into:

$$\text{grad } \log f_{ij}(Z) = ZG_{ij}^\top(Z), \quad (8.14)$$

where $G_{ij}: \text{SO}(n) \mapsto \mathfrak{so}(n)$ is a mapping that will play an important role in the sequel. In particular, the i^{th} gradient component now takes the short form:

$$\text{grad}_i L(\mathbf{R}) = \sum_{i \sim j} G_{ij}(Z_{ij}) R_i.$$

Let us consider a canonical orthonormal basis of $\mathfrak{so}(n)$: (E_1, \dots, E_d) , with $d = n(n - 1)/2$. For $n = 3$, we pick this one:

$$E_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, E_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, E_3 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}. \quad (8.15)$$

An obvious generalization yields similar bases for other values of n . We can transport this canonical basis into an orthonormal basis for the tangent space $T_{R_i} \text{SO}(n)$ as $(R_i E_1, \dots, R_i E_d)$. Let us also fix an orthonormal basis for the tangent space at $\mathbf{R} \in \mathcal{P}$, as

$$(\xi_{ik})_{i=1 \dots N, k=1 \dots d}, \text{ with } \xi_{ik} = (0, \dots, 0, R_i E_k, 0, \dots, 0), \\ \text{a zero vector except for the } i^{\text{th}} \text{ component equal to } R_i E_k. \quad (8.16)$$

The FIM w.r.t. this basis is composed of $N \times N$ blocks of size $d \times d$. Let us index the (k, ℓ) entry inside the (i, j) block as $F_{ij,k\ell}$. Accordingly, the matrix F at \mathbf{R} is defined by (see Definition 6.3):

$$F_{ij,k\ell} = \mathbb{E} \{ \langle \text{grad } L(\mathbf{R}), \xi_{ik} \rangle \cdot \langle \text{grad } L(\mathbf{R}), \xi_{j\ell} \rangle \} \\ = \mathbb{E} \{ \langle \text{grad}_i L(\mathbf{R}), R_i E_k \rangle \cdot \langle \text{grad}_j L(\mathbf{R}), R_j E_\ell \rangle \} \\ = \sum_{i \sim r} \sum_{j \sim s} \mathbb{E} \{ \langle G_{ir}(Z_{ir}), R_i E_k R_i^\top \rangle \cdot \langle G_{js}(Z_{js}), R_j E_\ell R_j^\top \rangle \}. \quad (8.17)$$

We prove that, in expectation, the mappings G_{ij} (8.14) are zero. This fact is directly related to the standard result from estimation theory stating that the average *score* for a given parameterized probability density function f is zero, Lemma 6.1.

Lemma 8.2. *Given a smooth probability density function $f: \text{SO}(n) \rightarrow \mathbb{R}^+$ and the mapping $G: \text{SO}(n) \rightarrow \mathfrak{so}(n)$ such that $\text{grad } \log f(Z) = ZG(Z)$, it holds that $\mathbb{E}\{G(Z)\} = 0$, where expectation is taken w.r.t. Z , distributed according to f .*

Proof. Define $h(Q) = \int_{\mathrm{SO}(n)} f(ZQ) d\mu(Z)$ for $Q \in \mathrm{SO}(n)$. Since f is a probability density function, bi-invariance of μ (8.9) yields $h(Q) \equiv 1$. Take gradients with respect to the parameter Q :

$$0 = \mathrm{grad} h(Q) = \int_{\mathrm{SO}(n)} \mathrm{grad}_Q f(ZQ) d\mu(Z) = \int_{\mathrm{SO}(n)} Z^\top \mathrm{grad} f(ZQ) d\mu(Z).$$

With a change of variable $Z := ZQ$, by bi-invariance of μ , we further obtain:

$$\int_{\mathrm{SO}(n)} Z^\top \mathrm{grad} f(Z) d\mu(Z) = 0.$$

Using this last result and the fact that $\mathrm{grad} \log f(Z) = \frac{1}{f(Z)} \mathrm{grad} f(Z)$ concludes:

$$\begin{aligned} \mathbb{E}\{G(Z)\} &= \int_{\mathrm{SO}(n)} Z^\top \mathrm{grad} \log f(Z) f(Z) d\mu(Z) \\ &= \int_{\mathrm{SO}(n)} Z^\top \mathrm{grad} f(Z) d\mu(Z) = 0. \end{aligned}$$

□

We now invoke Assumption 8.2 (independence). Independence of Z_{ij} and Z_{pq} for two distinct edges $\{i, j\}$ and $\{p, q\}$ implies that, for any two functions $\phi_1, \phi_2: \mathrm{SO}(n) \rightarrow \mathbb{R}$, it holds that

$$\mathbb{E}\{\phi_1(Z_{ij})\phi_2(Z_{pq})\} = \mathbb{E}\{\phi_1(Z_{ij})\}\mathbb{E}\{\phi_2(Z_{pq})\},$$

provided all involved expectations exist. Using both this and Lemma 8.2, most terms in (8.17) vanish and we obtain a simplified expression for the matrix F :

$$F_{ij,k\ell} = \begin{cases} \sum_{i \sim r} \mathbb{E}\{\langle G_{ir}(Z_{ir}), R_i E_k R_i^\top \rangle \cdot \langle G_{ir}(Z_{ir}), R_i E_\ell R_i^\top \rangle\}, & \text{if } i = j, \\ \mathbb{E}\{\langle G_{ij}(Z_{ij}), R_i E_k R_i^\top \rangle \cdot \langle G_{ji}(Z_{ji}), R_j E_\ell R_j^\top \rangle\}, & \text{if } i \neq j \text{ and } i \sim j, \\ 0, & \text{if } i \neq j \text{ and } i \not\sim j. \end{cases} \quad (8.18)$$

We further manipulate the second case, which involves both G_{ij} and G_{ji} , by noting that those are deterministically linked. Indeed, by symmetry of the measurements ($H_{ij} = H_{ji}^\top$), we have that (i) $Z_{ji} = R_j R_i^\top Z_{ij}^\top R_i R_j^\top$ and

(ii) $f_{ij}(Z_{ij}) = f_{ji}(Z_{ji})$. Invoking Assumption 8.3, since Z_{ij} and Z_{ji} have the same eigenvalues, it follows that $f_{ij}(Z) = f_{ji}(Z)$ for all $Z \in \text{SO}(n)$. As a by-product, it also holds that $G_{ij}(Z) = G_{ji}(Z)$ for all $Z \in \text{SO}(n)$. Still under Assumption 8.3, we show in the appendix Section B.1 that

$$\begin{aligned} \forall Q \in \text{O}(n), \quad G_{ij}(QZQ^\top) &= QG_{ij}(Z)Q^\top, \text{ and} \\ G_{ij}(Z^\top) &= -G_{ij}(Z). \end{aligned} \quad (8.19)$$

Combining these observations, we obtain:

$$G_{ji}(Z_{ji}) = G_{ij}(Z_{ji}) = G_{ij}(R_j R_i^\top Z_{ij}^\top R_i R_j^\top) = -R_j R_i^\top G_{ij}(Z_{ij}) R_i R_j^\top.$$

The minus sign, which plays an important role in the structure of the FIM, comes about via the skew-symmetry of G_{ij} . The following identity thus holds:

$$\langle G_{ji}(Z_{ji}), R_j E_\ell R_j^\top \rangle = -\langle G_{ij}(Z_{ij}), R_i E_\ell R_i^\top \rangle. \quad (8.20)$$

This can advantageously be plugged into (8.18).

Describing the expectations appearing in (8.18) takes us through a couple of lemmas. Let us, for a certain pair (i, j) , $i \sim j$, introduce the functions $h_k : \text{SO}(n) \rightarrow \mathbb{R}$, $k = 1 \dots d$:

$$h_k(Z) = \langle G_{ij}(Z), R_i E_k R_i^\top \rangle, \quad (8.21)$$

where we chose to not overload the notation h_k with an explicit reference to the pair (i, j) , as this will always be clear from the context. We may rewrite the FIM in terms of the functions h_k , starting from (8.18) and incorporating (8.20):

$$F_{ij,k\ell} = \begin{cases} \sum_{i \sim r} \mathbb{E} \{ h_k(Z_{ir}) \cdot h_\ell(Z_{ir}) \}, & \text{if } i = j, \\ -\mathbb{E} \{ h_k(Z_{ij}) \cdot h_\ell(Z_{ij}) \}, & \text{if } i \neq j \text{ and } i \sim j, \\ 0, & \text{if } i \neq j \text{ and } i \not\sim j. \end{cases}$$

Another consequence of Assumption 8.3 is that the functions $h_k(Z)$ and $h_\ell(Z)$ are uncorrelated for $k \neq \ell$, where Z is distributed according to the density f_{ij} . As a consequence, $F_{ij,k\ell} = 0$ for $k \neq \ell$, i.e., the $d \times d$ blocks of F are diagonal. We establish this fact in Lemma 8.4, right after a technical lemma.

Lemma 8.3. *Let $E, E' \in \mathfrak{so}(n)$ such that $E_{ij} = -E_{ji} = 1$ and $E'_{k\ell} = -E'_{\ell k} = 1$ (all other entries are zero), with $\langle E, E' \rangle = 0$, i.e., $\{i, j\} \neq \{k, \ell\}$. Then, there exists $P \in \text{O}(n)$ a signed permutation such that $P^\top EP = E'$ and $P^\top E'P = -E$.*

Proof. See the appendix Section B.2 for a proof and an explanation of why this is not direct. \square

Lemma 8.4. *Let $Z \in \mathrm{SO}(n)$ be a random variable distributed according to f_{ij} . The random variables $h_k(Z)$ and $h_\ell(Z)$, $k \neq \ell$, as defined in (8.21) have zero mean and are uncorrelated, i.e., $\mathbb{E}\{h_k(Z)\} = \mathbb{E}\{h_\ell(Z)\} = 0$ and $\mathbb{E}\{h_k(Z) \cdot h_\ell(Z)\} = 0$. Furthermore, it holds that $\mathbb{E}\{h_k^2(Z)\} = \mathbb{E}\{h_\ell^2(Z)\}$.*

Proof. The first part follows directly from Lemma 8.2. We show the second part. Consider a signed permutation matrix $P_{k\ell} \in \mathrm{O}(n)$ such that $P_{k\ell}^\top E_k P_{k\ell} = E_\ell$ and $P_{k\ell}^\top E_\ell P_{k\ell} = -E_k$. Such a matrix always exists according to Lemma 8.3. Then, identity (8.19) yields:

$$h_k(R_i P_{k\ell} R_i^\top Z R_i P_{k\ell}^\top R_i^\top) = \langle G_{ij}(Z), R_i P_{k\ell}^\top E_k P_{k\ell} R_i^\top \rangle = h_\ell(Z).$$

Likewise,

$$h_\ell(R_i P_{k\ell} R_i^\top Z R_i P_{k\ell}^\top R_i^\top) = -h_k(Z).$$

These identities as well as the (extended) bi-invariance (8.9) of the Haar measure μ on $\mathrm{SO}(n)$ and the fact that f_{ij} is a spectral function yield, using the change of variable $Z := R_i P_{k\ell} R_i^\top Z R_i P_{k\ell}^\top R_i^\top$ going from the first to the second integral:

$$\begin{aligned} \mathbb{E}\{h_k(Z) \cdot h_\ell(Z)\} &= \int_{\mathrm{SO}(n)} h_k(Z) h_\ell(Z) f_{ij}(Z) d\mu(Z) \\ &= \int_{\mathrm{SO}(n)} -h_\ell(Z) h_k(Z) f_{ij}(Z) d\mu(Z) = -\mathbb{E}\{h_k(Z) \cdot h_\ell(Z)\}. \end{aligned}$$

Hence, $\mathbb{E}\{h_k(Z) \cdot h_\ell(Z)\} = 0$. We prove the last statement using the same change of variable:

$$\begin{aligned} \mathbb{E}\{h_k^2(Z)\} &= \int_{\mathrm{SO}(n)} h_k^2(Z) f_{ij}(Z) d\mu(Z) \\ &= \int_{\mathrm{SO}(n)} h_\ell^2(Z) f_{ij}(Z) d\mu(Z) = \mathbb{E}\{h_\ell^2(Z)\}. \end{aligned}$$

We note that, more generally, it can be shown that the h_k 's are identically distributed. \square

The skew-symmetric matrices $(R_i E_1 R_i^\top, \dots, R_i E_d R_i^\top)$ form an orthonormal basis of the Lie algebra $\mathfrak{so}(n)$. Consequently, we may expand each mapping G_{ij} in this basis and express its squared norm as:

$$G_{ij}(Z) = \sum_{k=1}^d h_k(Z) \cdot R_i E_k R_i^\top, \quad \|G_{ij}(Z)\|^2 = \sum_{k=1}^d h_k^2(Z).$$

Since by Lemma 8.4 the quantity $\mathbb{E}\{h_k^2(Z_{ij})\}$ does not depend on k , it follows that:

$$\mathbb{E}\{h_k^2(Z_{ij})\} = \frac{1}{d}\mathbb{E}\{\|G_{ij}(Z_{ij})\|^2\}, \quad k = 1 \dots d.$$

This further shows that the $d \times d$ blocks that constitute the FIM have constant diagonal. Hence, F can be expressed as the Kronecker product (\otimes) of some matrix with the identity I_d . Let us define the following (positive) weights on the edges of the measurement graph:

$$w_{ij} = w_{ji} = \mathbb{E}\{\|G_{ij}(Z_{ij})\|^2\} \triangleq \mathbb{E}\{\|\text{grad} \log f_{ij}(Z_{ij})\|^2\}. \quad (8.22)$$

Also let $w_{ij} = w_{ji} = 0$ if i and j are not connected. Let $\mathcal{A} \in \mathbb{R}^{N \times N}$ be the adjacency matrix of the measurement graph with $\mathcal{A}_{ij} = w_{ij}$ and let $\mathcal{D} \in \mathbb{R}^{N \times N}$ be the diagonal degree matrix such that $\mathcal{D}_{ii} = \sum_{i \sim j} w_{ij}$. Then, the *weighted Laplacian matrix* $\mathcal{L} = \mathcal{D} - \mathcal{A}$, $\mathcal{L} = \mathcal{L}^\top \succeq 0$, is given by:

$$\mathcal{L}_{ij} = \begin{cases} \sum_{i \sim r} w_{ir}, & \text{if } i = j, \\ -w_{ij}, & \text{if } i \neq j \text{ and } i \sim j, \\ 0, & \text{if } i \neq j \text{ and } i \not\sim j. \end{cases} \quad (8.23)$$

It is now apparent that the matrix $F \in \mathbb{R}^{dN \times dN}$ is tightly related to \mathcal{L} . We summarize this in the following theorem.

Theorem 8.5 (FIM for synchronization). *Let $R_1, \dots, R_N \in \text{SO}(n)$ be unknown but fixed rotations and let $H_{ij} = Z_{ij}R_iR_j^\top$ for $i \sim j$, with the Z_{ij} 's random rotations which fulfill Assumptions 8.1–8.3. Consider the problem of estimating the R_i 's given a realization of the H_{ij} 's. The Fisher information matrix (Definition 6.3) of that estimation problem with respect to the basis (8.16) is given by*

$$F = \frac{1}{d}(\mathcal{L} \otimes I_d), \quad (8.24)$$

where \otimes denotes the Kronecker product, $d = \dim \text{SO}(n) = n(n-1)/2$, I_d is the $d \times d$ identity matrix and \mathcal{L} is the weighted Laplacian matrix (8.23) of the measurement graph.

The Laplacian matrix has a number of properties, some of which will yield nice interpretations when deriving the Cramér-Rao bounds. One remarkable fact is that this FIM does not depend on $\mathbf{R} = (R_1, \dots, R_N)$, the set of true rotations. This is an appreciable property seen as \mathbf{R} is unknown in practice. This stems from the strong symmetries in our problem.

Another important feature of this FIM is that it is rank deficient. Indeed, for a connected measurement graph, \mathcal{L} has exactly one zero eigenvalue

(and more if the graph is disconnected) associated to the vector of all ones, $\mathbf{1}_N$. The null space of the FIM is thus composed of all vectors of the form $\mathbf{1}_N \otimes t$, with $t \in \mathbb{R}^d$ arbitrary. This corresponds to the vertical spaces of \mathcal{P} w.r.t. the equivalence relation (8.4), i.e., the null space consists in all tangent vectors that move all rotations R_i in the same direction, leaving their relative positions unaffected. This makes perfect sense: the distribution of the measurements H_{ij} is also unaffected by such changes, hence the FIM, seen as a quadratic form, takes up a zero value when applied to the corresponding vectors. We will need the special tools developed in Chapter 7 to deal with this (structured) singularity when deriving the CRB's in the next section.

Notice how Assumption 8.2 (independence) gave F a block structure based on the sparsity pattern of the Laplacian matrix, while Assumption 8.3 (spectral pdf's) made each block proportional to the $d \times d$ identity matrix and made F independent of \mathbf{R} .

Example 8.4 (Langevin distributions). *(Continued from Example 8.2)* Considering the Langevin pdf f (5.4), $\text{grad} \log f(Z) = -\kappa Z \text{skew}(Z)$ and we find that the weight w associated to this noise distribution is a function given by:

$$w = w_n(\kappa) = \mathbb{E} \{ \|\text{grad} \log f(Z)\|^2 \} = \frac{\kappa^2}{4} \int_{\text{SO}(n)} \|Z - Z^\top\|^2 f(Z) d\mu(Z).$$

Since the integrand is again a class function, apply the tools from Appendix A to derive for $n = 2, 3$:

$$w_2(\kappa) = \kappa \frac{I_1(2\kappa)}{I_0(2\kappa)}, \quad w_3(\kappa) = \frac{\kappa}{2} \frac{(2-\kappa)I_1(2\kappa) + \kappa I_3(2\kappa)}{I_0(2\kappa) - I_1(2\kappa)}.$$

The functions $I_\nu(z)$ are the modified Bessel functions of the first kind (A.4). We used formulas for the normalization constants c_2 (8.11) and c_3 (8.12) as well as the identity $I_1(2\kappa) = \kappa(I_0(2\kappa) - I_2(2\kappa))$.

For the special case $n = 2$, taking the concentrations for all measurements to be equal, we find that the FIM is proportional to the unweighted Laplacian matrix $D - A$, with D the degree matrix and A the adjacency matrix of the measurement graph. This particular result was shown before via another method in (Howard et al., 2010). For the derivation in the latter work, commutativity of rotations in the plane is instrumental, and hence the proof method does not—at least in the proposed form—transfer to $\text{SO}(n)$ for $n \geq 3$.

Example 8.5 (Mixture of Langevin). *(Continued from Example 8.3)* The information weight $w = w_n(\kappa, \kappa', p)$ for this model is derived in the appendix

Section A.2. Because we need it in Section 8.7 to study the resilience of synchronization against outliers, we do give explicit formulas for the special case $\kappa' = 0$ here:

$$w_2(\kappa, 0, p) = \frac{(p\kappa)^2}{c_2(\kappa)} \frac{1}{\pi} \int_0^\pi \frac{(1 - \cos 2\theta) \exp(4\kappa \cos \theta)}{p \exp(2\kappa \cos \theta) + (1-p)c_2(\kappa)} d\theta, \quad (8.25)$$

$$w_3(\kappa, 0, p) = \frac{(p\kappa)^2 \exp(2\kappa)}{c_3(\kappa)} \frac{1}{\pi} \int_0^\pi \frac{(1 - \cos 2\theta)(1 - \cos \theta) \exp(4\kappa \cos \theta)}{p \exp(\kappa(1 + 2 \cos \theta)) + (1-p)c_3(\kappa)} d\theta.$$

These integrals may be evaluated numerically.

8.5 The Cramér-Rao bounds

We now apply the CRB's developed in Chapter 7 to the synchronization problem, using the FIM derived in the previous section. We distinguish between the anchored and the anchor-free cases.

We should bear in mind that these intrinsic CRB's are fundamentally asymptotic bounds for large SNR. At low SNR, the bounds may fail to capture features of the estimation problem that become dominant for large errors. In particular, since the parameter spaces \mathcal{P}_A and \mathcal{P}_\emptyset are compact, there is an upper bound on how badly one can estimate the true rotations. Because of their local nature (intrinsic CRB's result from a small-error analysis), the bounds we establish here are unable to capture this essential feature.

In the sequel, the proviso *at large SNR* thus designates noise levels such that efficient estimators commit errors small enough that the intrinsic CRB analysis holds. For reasons that will become clear in this section, for anchor-free synchronization, we define a notion of SNR as the quantity

$$\text{SNR}_\emptyset = \frac{(N-1)\mathbb{E}\{\text{dist}^2(Z_{\text{uni}}, I_n)\}}{d^2 \text{trace}(\mathcal{L}^\dagger)},$$

where the expectation is taken w.r.t. Z_{uni} , uniformly distributed over $\text{SO}(n)$. The numerator is a baseline which corresponds to the variance of a random estimator—see Section 8.7.1. The denominator has units of variance as well and is small when the measurement graph is well connected by good measurements. An SNR can be considered large if $\text{SNR}_\emptyset \gg 1$. For anchored synchronization, a similar definition holds with \mathcal{L} replaced by the masked Laplacian \mathcal{L}_A (8.27) and $N-1$ replaced by $N-|A|$.

8.5.1 Anchored synchronization

When anchors are provided, the rotation matrices R_i for $i \in A$, $A \neq \emptyset$ are known. The parameter space then becomes \mathcal{P}_A (5.7), which is a Rie-

mannian submanifold of \mathcal{P} . The synchronization problem is well-posed on \mathcal{P}_A , provided there is at least one anchor in each connected component of the measurement graph. Let us define the covariance matrix of an estimator for anchored synchronization, in agreement with Definition 6.6 and equation (7.2).

Definition 8.1 (anchored covariance). *The covariance matrix of an estimator $\hat{\mathbf{R}}$ mapping each possible set of measurements H_{ij} to a point in \mathcal{P}_A , expressed w.r.t. the orthonormal basis (8.16) of $\mathrm{T}_{\mathbf{R}}\mathcal{P}$, is given by:*

$$(C_A)_{ij,k\ell} = \mathbb{E} \left\{ \langle \mathrm{Log}_{\mathbf{R}}(\hat{\mathbf{R}}), \xi_{ik} \rangle \cdot \langle \mathrm{Log}_{\mathbf{R}}(\hat{\mathbf{R}}), \xi_{j\ell} \rangle \right\}, \quad (8.26)$$

where the indexing convention is the same as for the FIM. Of course, all $d \times d$ blocks (i, j) such that either i or j or both are in A (anchored) are zero by construction. In particular, the variance of $\hat{\mathbf{R}}$ is the trace of C_A :

$$\mathrm{trace} C_A = \mathbb{E} \left\{ \|\mathrm{Log}_{\mathbf{R}}(\hat{\mathbf{R}})\|^2 \right\} = \mathbb{E} \left\{ \mathrm{dist}^2(\mathbf{R}, \hat{\mathbf{R}}) \right\},$$

where dist is the geodesic distance on \mathcal{P}_A (5.13).

CRB's link this covariance matrix to the FIM derived in the previous section through Theorem 7.4.

Theorem 8.6 (anchored CRB). *Given any unbiased estimator $\hat{\mathbf{R}}$ for synchronization on \mathcal{P}_A , at large SNR, the covariance matrix C_A (8.26) and the FIM F (8.24) obey the matrix inequality (assuming at least one anchor in each connected component):*

$$C_A \succeq F_A^\dagger - \frac{1}{3} \left(R_m(F_A^\dagger) F_A^\dagger + F_A^\dagger R_m(F_A^\dagger) \right),$$

where $F_A = P_A F P_A$ and P_A is the orthogonal projector from $\mathrm{T}_{\mathbf{R}}\mathcal{P}$ to $\mathrm{T}_{\mathbf{R}}\mathcal{P}_A$, expressed w.r.t. the orthonormal basis (8.16). The operator $R_m: \mathbb{R}^{dN \times dN} \rightarrow \mathbb{R}^{dN \times dN}$ involves the Riemannian curvature tensor of \mathcal{P}_A and is detailed in Section 8.6.

The effect of P_A is to set all rows and columns corresponding to anchored rotations to zero. Thus, we introduce the masked Laplacian \mathcal{L}_A :

$$(\mathcal{L}_A)_{ij} = \begin{cases} \mathcal{L}_{ij} & \text{if } i, j \notin A, \\ 0 & \text{otherwise.} \end{cases} \quad (8.27)$$

Then, the projected FIM is simply:

$$F_A = \frac{1}{d} (\mathcal{L}_A \otimes I_d). \quad (8.28)$$

The pseudoinverse of F_A is given by $F_A^\dagger = d(\mathcal{L}_A^\dagger \otimes I_d)$, since for arbitrary matrices A and B , it holds that $(A \otimes B)^\dagger = A^\dagger \otimes B^\dagger$ (Bernstein, 2009, Fact 7.4.32). Notice that the rows and columns of \mathcal{L}_A^\dagger corresponding to anchors are also zero. Theorem 8.6 then yields the sought CRB:

$$C_A \succeq d(\mathcal{L}_A^\dagger \otimes I_d) + \text{curvature terms}.$$

In particular, for $n = 2$, the manifold \mathcal{P}_A is flat and $d = 1$. Hence, the curvature terms vanish exactly ($R_m \equiv 0$) and the CRB reads:

$$C_A \succeq \mathcal{L}_A^\dagger.$$

For $n = 3$, including the curvature terms as detailed in Section 8.6 yields this CRB:

$$C_A \succeq 3 \left(\mathcal{L}_A^\dagger - \frac{1}{4} \left(\text{ddiag}(\mathcal{L}_A^\dagger) \mathcal{L}_A^\dagger + \mathcal{L}_A^\dagger \text{ddiag}(\mathcal{L}_A^\dagger) \right) \right) \otimes I_3, \quad (8.29)$$

where ddiag sets all off-diagonal entries of a matrix to zero. At large SNR, that is, for small values of $\text{trace}(\mathcal{L}_A^\dagger)$, the curvature terms hence effectively become negligible compared to the leading term. For general n , neglecting curvature if $n \geq 3$, the variance is lower-bounded as follows:

$$\mathbb{E} \left\{ \text{dist}^2(\mathbf{R}, \hat{\mathbf{R}}) \right\} \geq d^2 \text{trace} \mathcal{L}_A^\dagger,$$

where dist is as defined by (5.13). It also holds for each node i that

$$\mathbb{E} \left\{ \text{dist}^2(R_i, \hat{R}_i) \right\} \geq d^2 (\mathcal{L}_A^\dagger)_{ii}. \quad (8.30)$$

This leads to a useful interpretation of the CRB in terms of a resistance distance on the measurement graph, as depicted in Figure 8.1. Indeed, for a general setting with one or more anchors, it can be checked that (Bouldin, 1973)

$$\mathcal{L}_A^\dagger = (J_A(\mathcal{D} - \mathcal{A})J_A)^\dagger = (J_A(I_N - \mathcal{D}^{-1}\mathcal{A})J_A)^\dagger \mathcal{D}^{-1},$$

where J_A is a diagonal matrix such that $(J_A)_{ii} = 1$ if $i \in A$ and $(J_A)_{ii} = 0$ otherwise. It is well-known, e.g. from (Doyle & Snell, 2000, § 1.2.6), that in the first factor of the right-hand side, $((J_A(I_N - \mathcal{D}^{-1}\mathcal{A})J_A)^\dagger)_{ii}$ is the average number of times a random walker starting at node i on the graph with transition probabilities $\mathcal{D}^{-1}\mathcal{A}$ will be at node i before hitting an anchor. This number is small if node i is strongly connected to anchors. In the CRB (8.30) on node i , $(\mathcal{L}_A^\dagger)_{ii}$ is thus the ratio between this anchor-connectivity measure and the overall amount of information available about node i directly, namely $\mathcal{D}_{ii} = \sum_{j \in V_i} w_{ij}$.

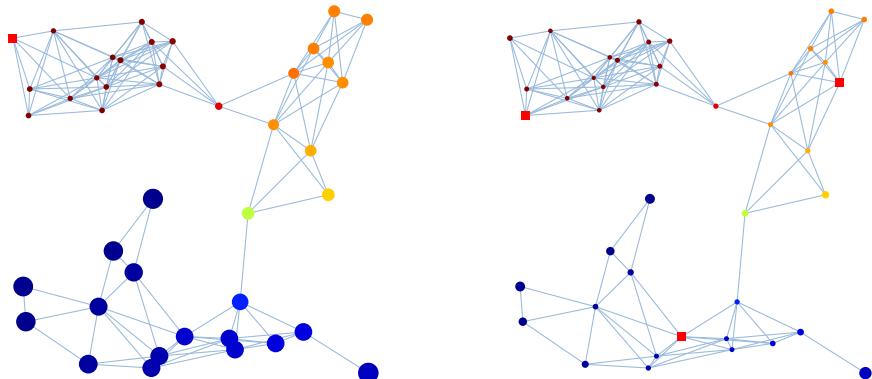


Figure 8.1: The Cramér-Rao bound for *anchored* synchronization (8.30) limits how well each individual rotation can be estimated. The two identical synchronization graphs above illustrate the effect of anchors. All edges have the same weight (i.i.d. noise). Anchors are red squares. Unknown rotations are round nodes colored according to the second eigenvector of \mathcal{L} to bring out the clusters. The area of node i is proportional to the lower bound on the average error for this node $\mathbb{E}\{\text{dist}^2(R_i, \hat{R}_i)\}$. On the left, there is only one anchor in the upper-left cluster. Hence, nodes in the lower-left cluster, which are separated from the anchor by two bottlenecks, will be harder to estimate accurately than in the situation on the right, where there is one anchor for each cluster. Node positions in the picture are irrelevant.

8.5.2 Anchor-free synchronization

When no anchors are provided, the global rotation ambiguity leads to the equivalence relation (8.4) on \mathcal{P} , which in turn leads to work on the Riemannian quotient parameter space \mathcal{P}_\emptyset (8.5). The synchronization problem is well-posed on \mathcal{P}_\emptyset as long as the measurement graph is connected, which we always assume in this work. Let us define the covariance matrix of an estimator for anchor-free synchronization, in agreement with Definition 6.6 and equation (7.6).

Definition 8.2 (anchor-free covariance). *The covariance matrix of an estimator $[\hat{\mathbf{R}}]$ mapping each possible set of measurements H_{ij} to a point in \mathcal{P}_\emptyset (that is, to an equivalence class in \mathcal{P}), expressed w.r.t. the orthonormal basis (8.16) of $\text{Tr}_{\mathbf{R}}\mathcal{P}$, is given by:*

$$(C_\emptyset)_{ij,kl} = \mathbb{E} \{ \langle \boldsymbol{\xi}, \boldsymbol{\xi}_{ik} \rangle \cdot \langle \boldsymbol{\xi}, \boldsymbol{\xi}_{je} \rangle \}, \text{ with} \\ \boldsymbol{\xi} = (\text{D}\pi(\mathbf{R})|_{H_{\mathbf{R}}})^{-1} [\text{Log}_{[\mathbf{R}]}([\hat{\mathbf{R}}])]. \quad (8.31)$$

That is, ξ (the random error vector) is the shortest horizontal vector such that $\text{Exp}_{\mathbf{R}}(\xi) \in [\hat{\mathbf{R}}]$ (8.7). We used the same indexing convention as for the FIM. In particular, the variance of $[\hat{\mathbf{R}}]$ is the trace of C_\emptyset :

$$\text{trace } C_\emptyset = \mathbb{E} \left\{ \| \text{Log}_{[\mathbf{R}]}([\hat{\mathbf{R}}]) \|^2 \right\} = \mathbb{E} \left\{ \text{dist}^2([\mathbf{R}], [\hat{\mathbf{R}}]) \right\},$$

where dist is the geodesic distance on \mathcal{P}_\emptyset (8.8).

CRB's link this covariance matrix to the FIM derived in the previous section through Theorem 7.5.

Theorem 8.7 (anchor-free CRB). *Given any unbiased estimator $[\hat{\mathbf{R}}]$ for synchronization on \mathcal{P}_\emptyset , at large SNR, the covariance matrix C_\emptyset (8.31) and the FIM F (8.24) obey the matrix inequality (assuming the measurement graph is connected):*

$$C_\emptyset \succeq F^\dagger - \frac{1}{3} (R_m(F^\dagger)F^\dagger + F^\dagger R_m(F^\dagger)),$$

where $R_m: \mathbb{R}^{dN \times dN} \rightarrow \mathbb{R}^{dN \times dN}$ involves the Riemannian curvature tensor of \mathcal{P}_\emptyset and is detailed in Section 8.6.

Theorem 8.7 then yields the sought CRB:

$$C_\emptyset \succeq d(\mathcal{L}^\dagger \otimes I_d) + \text{curvature terms}. \quad (8.32)$$

We compute the curvature terms explicitly in Section 8.6. and show they can be neglected for large SNR. In particular, for $n = 2$, the manifold \mathcal{P}_\emptyset is flat and $d = 1$. Hence:

$$C_\emptyset \succeq \mathcal{L}^\dagger.$$

For $n = 3$, the curvature terms are the same as those for the anchored case, with an additional term that decreases as $1/N$. For (not so) large N then, the bound (8.29) is a good bound for $n = 3$, anchor-free synchronization. For general n , neglecting curvature for $n \geq 3$, the variance is lower-bounded as follows:

$$\mathbb{E} \left\{ \text{dist}^2([\mathbf{R}], [\hat{\mathbf{R}}]) \right\} \geq d^2 \text{trace } \mathcal{L}^\dagger, \quad (8.33)$$

where dist is as defined by (8.8).

For the remainder of this section, we work out an interpretation of (8.32). This matrix inequality entails that, for all $x \in \mathbb{R}^{dN}$ (neglecting curvature terms if $n \geq 3$):

$$x^\top C_\emptyset x \geq d x^\top (\mathcal{L}^\dagger \otimes I_d) x. \quad (8.34)$$

As both the covariance and the FIM correspond to positive semidefinite operators on the horizontal space $H_{\mathbf{R}}$, this is really only meaningful when x is the vector of coordinates of a horizontal vector $\boldsymbol{\eta} = (\eta_1, \dots, \eta_N) \in H_{\mathbf{R}}$. We emphasize that this restriction implies that the anchor-free CRB, as it should, only conveys information about *relative* rotations. It does not say anything about singled-out rotations in particular. Let e_i, e_j denote the i^{th} and j^{th} columns of the identity matrix I_N and let e_k denote the k^{th} column of I_d . We consider $x = (e_i - e_j) \otimes e_k$, which corresponds to the zero horizontal vector $\boldsymbol{\eta}$ except for $\eta_i = R_i E_k$ and $\eta_j = -R_j E_k$, with $E_k \in \mathfrak{so}(n)$ the k^{th} element of the orthonormal basis of $\mathfrak{so}(n)$ picked as in (8.15). By definition of C_\emptyset and of the error vector $\boldsymbol{\xi} = (R_1 \Omega_1, \dots, R_N \Omega_N) \in H_{\mathbf{R}}$ (8.31),

$$x^\top C_\emptyset x = \mathbb{E} \left\{ \langle \boldsymbol{\xi}, \boldsymbol{\eta} \rangle^2 \right\} = \mathbb{E} \left\{ \langle \Omega_i - \Omega_j, E_k \rangle^2 \right\}.$$

On the other hand, we have

$$d x^\top (\mathcal{L}^\dagger \otimes I_d) x = d (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j).$$

These two last quantities are related by inequality (8.34). Summing for $k = 1 \dots d$ on both sides of this inequality, we find:

$$\mathbb{E} \left\{ \|\Omega_i - \Omega_j\|_{\text{F}}^2 \right\} \geq d^2 (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j).$$

Now remember that the error vector $\boldsymbol{\xi}$ (8.31) is the shortest horizontal vector such that $\text{Exp}_{\mathbf{R}}(\boldsymbol{\xi}) \in [\hat{\mathbf{R}}]$. Without loss of generality, assume $\hat{\mathbf{R}}$ is aligned such that $\text{Exp}_{\mathbf{R}}(\boldsymbol{\xi}) = \hat{\mathbf{R}}$. Then, $\hat{R}_i = R_i \exp(\Omega_i)$ for all i . It follows that

$$\begin{aligned} \hat{R}_i \hat{R}_j^\top &= R_i \exp(\Omega_i) \exp(-\Omega_j) R_j^\top, \quad \text{hence} \\ \text{dist}^2(R_i R_j^\top, \hat{R}_i \hat{R}_j^\top) &= \|\log(\exp(\Omega_i) \exp(-\Omega_j))\|_{\text{F}}^2. \end{aligned}$$

For commuting Ω_i and Ω_j —which is always the case for $n = 2$ —we have

$$\log(\exp(\Omega_i) \exp(-\Omega_j)) = \Omega_i - \Omega_j.$$

For $n \geq 3$, this still approximately holds in small error regimes (that is, for small enough Ω_i, Ω_j), by the Baker-Campbell-Hausdorff formula. Hence,

$$\begin{aligned} \mathbb{E} \left\{ \text{dist}^2(R_i R_j^\top, \hat{R}_i \hat{R}_j^\top) \right\} &\approx \mathbb{E} \left\{ \|\Omega_i - \Omega_j\|_{\text{F}}^2 \right\} \\ &\geq d^2 (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j). \end{aligned} \tag{8.35}$$

The quantity $\text{trace}(\mathcal{D}) \cdot (e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j)$ is sometimes called the squared Euclidean commute time distance (ECTD) (Saerens *et al.*, 2004) between

nodes i and j . It is also known as the electrical resistance distance. For a random walker on the graph with transition probabilities $\mathcal{D}^{-1}\mathcal{A}$, this quantity is the average commute time distance, that is, the number of steps it takes on average for a random walker starting at node i to hit node j then node i again. The right-hand side of (8.35) is thus inversely proportional to the quantity and quality of information linking these two nodes. It decreases whenever the number of paths between them increases or whenever an existing path is made more informative, i.e., weights on that path are increased.

Still in (Saerens *et al.*, 2004), it is shown in Section 5 how principal component analysis (PCA) on \mathcal{L}^\dagger can be used to embed the nodes in a low dimensional subspace such that the Euclidean distance separating two nodes is similar to the ECTD separating them in the graph. For synchronization, such an embedding naturally groups together nodes whose relative rotations can be accurately estimated, as depicted in Figure 8.2.

8.6 Curvature terms

We compute the curvature terms from theorems 8.6 and 8.7 for $n = 2$ and $n = 3$ explicitly. (See Section 2.8 for a brief introduction to curvature.) We first treat \mathcal{P}_A (5.7), then \mathcal{P}_\emptyset (8.5). We show that for rotations in the plane ($n = 2$), the parameter spaces are flat, so that curvature terms vanish exactly. For rotations in space ($n = 3$), we compute the curvature terms explicitly and show that they are on the order of $\mathcal{O}(\text{SNR}^{-2})$, whereas dominant terms in the CRB are on the order of $\mathcal{O}(\text{SNR}^{-1})$, for the notion of SNR proposed in Section 8.5. It is expected that curvature terms are negligible for $n \geq 4$ too for the same reasons, but we do not conduct the calculations.

8.6.1 Curvature terms for \mathcal{P}_A

The manifold \mathcal{P}_A (5.7) is a (product) Lie group. Hence, the Riemannian curvature tensor \mathcal{R} of \mathcal{P}_A on the tangent space $T_{\mathbf{R}}\mathcal{P}_A$ is given by a simple formula (O'Neill, 1983, Corollary 11.10, p. 305):

$$\langle \mathcal{R}(\mathbf{X}, \mathbf{Y})\mathbf{Y}, \mathbf{X} \rangle = \frac{1}{4} \|[\mathbf{X}, \mathbf{Y}]\|^2,$$

where $[\mathbf{X}, \mathbf{Y}]$ is the Lie bracket of $\mathbf{X} = (X_1, \dots, X_N)$ and $\mathbf{Y} = (Y_1, \dots, Y_N)$, two vectors (not necessarily orthonormal) in the tangent space $T_{\mathbf{R}}\mathcal{P}_A$. Following Theorem 7.4, in order to compute the curvature terms for the CRB of synchronization on \mathcal{P}_A , we first need to compute

$$\mathbf{R}_m[\mathbf{Y}', \mathbf{Y}'] \triangleq \mathbb{E} \{ \langle \mathcal{R}(\mathbf{X}, P_A \mathbf{Y}') P_A \mathbf{Y}', \mathbf{X} \rangle \}, \quad (8.36)$$

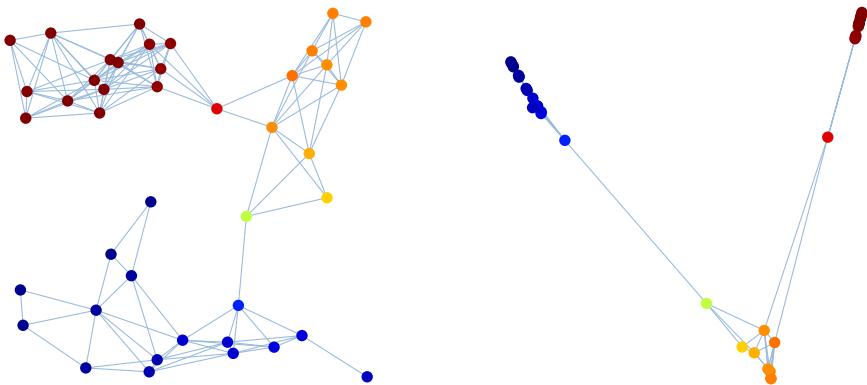


Figure 8.2: The Cramér-Rao bound for *anchor-free* synchronization (8.35) limits how well the relative rotation between two nodes can be estimated, in proportion to the Euclidean commute time distance (ECTD) separating them in the graph. Left: each node in the synchronization graph corresponds to a rotation to estimate and each edge corresponds to a measurement of relative rotation. Noise affecting the measurements is i.i.d., hence all edges have the same weight. Nodes are colored according to the second eigenvector of \mathcal{L} (the Fiedler vector). Node positions are irrelevant. Right: ECTD-embedding of the same graph in the plane, such that the distance between two nodes i and j in the picture is mostly proportional to the ECTD separating them, which is essentially a lower bound on $\mathbb{E}\{\text{dist}^2(R_i R_j^\top, \hat{R}_i \hat{R}_j^\top)\}^{1/2}$. In other words: the closer two nodes are, the better their relative rotation can be estimated. Notice that the node colors correspond to the horizontal coordinate in the right picture. See also Section 8.7.2.

where \mathbf{Y}' is any tangent vector in $T_{\mathbf{R}}\mathcal{P}$ and $P_A\mathbf{Y}'$ is its orthogonal projection on $T_{\mathbf{R}}\mathcal{P}_A$. We expand \mathbf{X} and $\mathbf{Y} = P_A\mathbf{Y}'$ using the orthonormal basis $(\xi_{k\ell})_{k=1\dots N, \ell=1\dots d}$ (8.16) of $T_{\mathbf{R}}\mathcal{P} \supset T_{\mathbf{R}}\mathcal{P}_A$:

$$\mathbf{X} = \sum_{k,\ell} \beta_{k\ell} \xi_{k\ell} \quad \text{and} \quad \mathbf{Y} = \sum_{k,\ell} \alpha_{k\ell} \xi_{k\ell},$$

such that $X_k = R_k \sum_\ell \beta_{k\ell} E_\ell$ and $Y_k = R_k \sum_\ell \alpha_{k\ell} E_\ell$. Of course, $\alpha_{k\ell} = \beta_{k\ell} = 0 \forall k \in A$. Then, since

$$[\mathbf{X}, \mathbf{Y}] = ([X_1, Y_1], \dots, [X_N, Y_N]),$$

it follows that:

$$\begin{aligned}\mathbf{R}_m[\mathbf{Y}, \mathbf{Y}] &= \mathbb{E} \left\{ \frac{1}{4} \|[\mathbf{X}, \mathbf{Y}]\|^2 \right\} = \mathbb{E} \left\{ \frac{1}{4} \sum_k \| [X_k, Y_k] \|^2 \right\} \\ &= \frac{1}{4} \sum_k \mathbb{E} \left\{ \left\| \sum_{\ell, s} \alpha_{k\ell} \beta_{ks} [E_\ell, E_s] \right\|^2 \right\}. \end{aligned} \quad (8.37)$$

For \mathbf{X} the tangent vector in $T_{\mathbf{R}}\mathcal{P}_A$ corresponding to the (random) estimation error $\text{Log}_{\mathbf{R}}(\hat{\mathbf{R}})$, the coefficients $\beta_{k\ell}$ are random variables. The covariance matrix C_A (8.26) is given in terms of these coefficients by:

$$(C_A)_{kk', \ell\ell'} = \mathbb{E} \{ \langle \mathbf{X}, \xi_{k\ell} \rangle \langle \mathbf{X}, \xi_{k'\ell'} \rangle \} = \mathbb{E} \{ \beta_{k\ell} \beta_{k'\ell'} \}.$$

The goal now is to express the entries of the matrix associated to \mathbf{R}_m as linear combinations of the entries of C_A .

For $n = 2$, of course, $\mathbf{R}_m \equiv 0$ since Lie brackets vanish owing to the commutativity of rotations in the plane.

For $n = 3$, the constant curvature of $\text{SO}(3)$ leads to nice expressions, which we obtain now. Let us consider the orthonormal basis (E_1, E_2, E_3) of $\mathfrak{so}(3)$ (8.15). Observe that it obeys

$$[E_1, E_2] = E_3/\sqrt{2}, \quad [E_2, E_3] = E_1/\sqrt{2}, \quad [E_3, E_1] = E_2/\sqrt{2}.$$

As a result, equation (8.37) simplifies and becomes:

$$\begin{aligned}\mathbf{R}_m[\mathbf{Y}, \mathbf{Y}] &= \\ \frac{1}{8} \sum_k \mathbb{E} \{ &(\alpha_{k2} \beta_{k3} - \alpha_{k3} \beta_{k2})^2 + (\alpha_{k3} \beta_{k1} - \alpha_{k1} \beta_{k3})^2 + (\alpha_{k1} \beta_{k2} - \alpha_{k2} \beta_{k1})^2 \}. \end{aligned} \quad (8.38)$$

We set out to compute the $dN \times dN$ matrix $R_m = R_m(C_A)$ (7.10) associated to the bi-linear operator \mathbf{R}_m w.r.t. the basis (8.16). By definition, $(R_m)_{kk', \ell\ell'} = \mathbf{R}_m[\xi_{k\ell}, \xi_{k'\ell'}]$. Equation (8.38) readily yields the diagonal entries ($k = k', \ell = \ell'$). Using the polarization identity to determine off-diagonal entries,

$$(R_m)_{kk', \ell\ell'} = \frac{1}{4} (\mathbf{R}_m[\xi_{k\ell} + \xi_{k'\ell'}, \xi_{k\ell} + \xi_{k'\ell'}] - \mathbf{R}_m[\xi_{k\ell} - \xi_{k'\ell'}, \xi_{k\ell} - \xi_{k'\ell'}]),$$

it follows through simple calculations (taking into account the orthogonal projection onto $T_{\mathbf{R}}\mathcal{P}_A$ that appears in (8.36)) that:

$$(R_m)_{kk', \ell\ell'} = \begin{cases} \frac{1}{8} \sum_{s \neq \ell} (C_A)_{kk, ss} & \text{if } k = k' \notin A, \ell = \ell', \\ -\frac{1}{8} (C_A)_{kk, \ell\ell'} & \text{if } k = k' \notin A, \ell \neq \ell', \\ 0 & \text{otherwise.} \end{cases} \quad (8.39)$$

Hence, $R_m(C_A)$ is a block-diagonal matrix whose nonzero entries are linear functions of the entries of C_A . Theorem 8.6 requires (8.39) to compute the matrix $R_m(F_A^\dagger)$. Considering the special structure of the diagonal blocks of F_A^\dagger (8.28) (they are proportional to I_3), we find that

$$R_m(F_A^\dagger) = \frac{1}{4} \text{ddiag}(F_A^\dagger) = \frac{3}{4} \text{ddiag}(\mathcal{L}_A^\dagger) \otimes I_3,$$

where ddiag puts all off-diagonal entries of a matrix to zero. Thus, as the SNR goes up and hence as \mathcal{L}_A^\dagger goes down, the curvature term $R_m(F_A^\dagger)F_A^\dagger + F_A^\dagger R_m(F_A^\dagger)$ in Theorem 8.6 will become negligible compared to the main term in the CRB, F_A^\dagger .

8.6.2 Curvature terms for \mathcal{P}_\emptyset

The manifold \mathcal{P}_\emptyset (8.5) is a quotient manifold of \mathcal{P} . Hence, the Riemannian curvature tensor \mathcal{R} of \mathcal{P}_\emptyset is given by O'Neill's formula (O'Neill, 1983, Thm 7.47, p. 213 and Lemma 3.39, p. 77), showing that the quotient operation can only increase the curvature of the parameter space:

$$\langle \mathcal{R}(D\pi \mathbf{X}, D\pi \mathbf{Y}) D\pi \mathbf{Y}, D\pi \mathbf{X} \rangle = \frac{1}{4} \|[\mathbf{X}, \mathbf{Y}]\|^2 + \frac{3}{4} \|[\mathbf{X}, \mathbf{Y}]^\mathcal{V}\|^2, \quad (8.40)$$

where \mathbf{X}, \mathbf{Y} are horizontal vectors in $H_{\mathbf{R}} \subset T_{\mathbf{R}}\mathcal{P}$ identified with tangent vectors to \mathcal{P}_\emptyset via the differential of the Riemannian submersion $D\pi(\mathbf{R})$ (8.6), denoted simply as $D\pi$ for convenience. The vector $[\mathbf{X}, \mathbf{Y}]^\mathcal{V} \in V_{\mathbf{R}} \subset T_{\mathbf{R}}\mathcal{P}$ is the vertical part of $[\mathbf{X}, \mathbf{Y}]$, i.e., the component that is parallel to the fibers. Since in our case, moving along a fiber consists in changing all rotations along the same direction, $[\mathbf{X}, \mathbf{Y}]^\mathcal{V}$ corresponds to the mean component of $[\mathbf{X}, \mathbf{Y}]$:

$$[\mathbf{X}, \mathbf{Y}]^\mathcal{V} = (R_1\omega, \dots, R_N\omega), \text{ with } \omega = \frac{1}{N} \sum_{k=1}^N [R_k^\top X_k, R_k^\top Y_k].$$

For $n = 2$, since $[\mathbf{X}, \mathbf{Y}] = 0$, $[\mathbf{X}, \mathbf{Y}]^\mathcal{V} = 0$ also, hence \mathcal{P}_\emptyset is still a flat manifold, despite the quotient operation. We now show that for $n = 3$ the curvature terms in Theorem 8.7 are equivalent to the curvature terms for \mathcal{P}_A with $A := \emptyset$ plus extra terms that decay as $1/N$ and can thus be neglected.

The curvature operator \mathbf{R}_m (Theorem 7.5) is given by:

$$\begin{aligned} \mathbf{R}_m[\xi_{k\ell}, \xi_{k\ell}] &\triangleq \mathbb{E} \{ \langle \mathcal{R}(D\pi \mathbf{X}, D\pi \xi_{k\ell}) D\pi \xi_{k\ell}, D\pi \mathbf{X} \rangle \} \\ &= \mathbb{E} \left\{ \frac{1}{4} \|[\mathbf{X}, \xi_{k\ell} - \xi_{k\ell}^\mathcal{V}]\|^2 + \frac{3}{4} \|[\mathbf{X}, \xi_{k\ell} - \xi_{k\ell}^\mathcal{V}]^\mathcal{V}\|^2 \right\}. \end{aligned}$$

The tangent vector $\xi_{k\ell} - \xi_{k\ell}^{\mathcal{V}}$ is, by construction, the horizontal part of $\xi_{k\ell}$. The vertical part decreases in size as N grows: $\xi_{k\ell}^{\mathcal{V}} = \frac{1}{N}(R_1 E_\ell, \dots, R_N E_\ell)$. It follows that:

$$\mathbb{E}\{\|[\mathbf{X}, \xi_{k\ell}]^{\mathcal{V}}\|^2\} = \mathbb{E}\{\|[\mathbf{X}, \xi_{k\ell}]\|^2\} (1 + \mathcal{O}(1/N)).$$

Hence, up to a factor that decays as $1/N$, the first term in the curvature operator \mathbf{R}_m is the same as that of the previous section for \mathcal{P}_A , with $A := \emptyset$. We now deal with the second term defining \mathbf{R}_m :

$$\begin{aligned} [\mathbf{X}, \xi_{k\ell}]^{\mathcal{V}} &= (R_1 \omega, \dots, R_N \omega), \text{ with} \\ \omega &= \frac{1}{N}[R_k^\top X_k, E_\ell] = \frac{1}{N} \sum_s \beta_{ks} [E_s, E_\ell]. \end{aligned}$$

It is now clear that for large N this second term is negligible compared to $\mathbb{E}\{\|[\mathbf{X}, \xi_{k\ell}]\|^2\}$:

$$\|[\mathbf{X}, \xi_{k\ell}]^{\mathcal{V}}\|^2 = N\|\omega\|^2 = \mathcal{O}(1/N).$$

Applying polarization to \mathbf{R}_m to compute off-diagonal terms then concludes the argument showing that the curvature terms in the CRB for synchronization of rotations on \mathcal{P}_\emptyset , despite an increased curvature owing to the quotient operation (8.40), are very close (within a $\mathcal{O}(1/N)$ term) to the curvature terms established earlier for synchronization on \mathcal{P}_A , with $A := 0$. We omit an exact derivation of these terms as it is quite lengthy and does not bring much insight to the problem.

8.7 Comments on, and consequences of the CRB

So far, we derived the CRB's for synchronization in both anchored and anchor-free settings. These bounds enjoy a rich structure and lend themselves to useful interpretations, as for the random walk perspective for example. The insight we gather about the synchronization problem by exploring the CRB's is validated by Chapter 5, where it is shown (numerically) that the CRB's seem to be achievable, thus making them relevant. In this section, we start by pointing out validity limits of the CRB's, namely the large SNR proviso. Visualization tools are then proposed to assist in graph analysis. Finally, we focus on anchor-free synchronization and comment upon the role of the Fiedler value of a measurement graph, the synchronizability of Erdős-Rényi random graphs and the remarkable resilience to outliers of synchronization.

8.7.1 The CRB is an asymptotic bound

As stressed in the introduction of Chapter 6, intrinsic CRB's are asymptotic bounds, that is, they are meaningful for small errors. This is in part due to the curvature terms which are only approximated by a truncated Taylor series, and in part due to the fact that the parameter spaces \mathcal{P}_A and \mathcal{P}_\emptyset are compact. Because of that, there is an upper-bound on the variance of any estimator. The CRB is unable to capture such a global feature because it is derived under the assumption that the logarithmic map Log is globally invertible, which compactness prevents. Hence, for arbitrarily low SNR, the CRB without curvature terms will predict an arbitrarily large variance and will be violated (this does not show on figures from Section 5.5 since the CRB's depicted include curvature terms, which in this case make them go to zero at low SNR). As a means to locate the point at which the CRB certainly stops making sense, consider the problem of estimating a rotation matrix $R \in \text{SO}(n)$ based on a measurement $Z \in \text{SO}(n)$ of R , and compute the variance of the (unbiased) estimator $\hat{R}(Z) := Z$ when Z is uniformly random, i.e., when no information is available.

Define $V_n = \mathbb{E} \{ \text{dist}^2(Z, R) \}$ for Z uniformly distributed over $\text{SO}(n)$. A computation using Weyl's formula yields:

$$V_2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} \|\log(Z^\top R)\|_{\text{F}}^2 d\theta = \frac{1}{2\pi} \int_{-\pi}^{\pi} 2\theta^2 d\theta = \frac{2\pi^2}{3}, \quad V_3 = \frac{2\pi^2}{3} + 4.$$

A reasonable upper-bound on the variance of an estimator should thus be $N'V_n$, where N' is the number of independent rotations to estimate ($N - 1$ for anchor-free synchronization, $N - |A|$ for anchored synchronization). A CRB larger than this should be disregarded.

8.7.2 Visualization tools

In deriving the anchor-free bounds for synchronization, we established that a lower bound on

$$\mathbb{E} \left\{ \text{dist}^2(R_i R_j^\top, \hat{R}_i \hat{R}_j^\top) \right\}$$

is proportional to the quantity $(e_i - e_j)^\top \mathcal{L}^\dagger (e_i - e_j)$. Of course, this analysis also holds for anchored graphs. Here, we detail how Figure 8.2 was produced following a PCA procedure (Saerens *et al.*, 2004) and show how this translates for anchored graphs, as depicted in Figure 8.3.

We treat both anchored and anchor-free scenarios, thus allowing A to be empty in this paragraph. Let $\mathcal{L}_A = V D V^\top$ be an eigendecomposition of \mathcal{L}_A , such that V is orthogonal and $D = \text{diag}(\lambda_1, \dots, \lambda_N)$. Let $X = (D^\dagger)^{1/2} V^\top$, an $N \times N$ matrix with columns x_1, \dots, x_N . Assume without

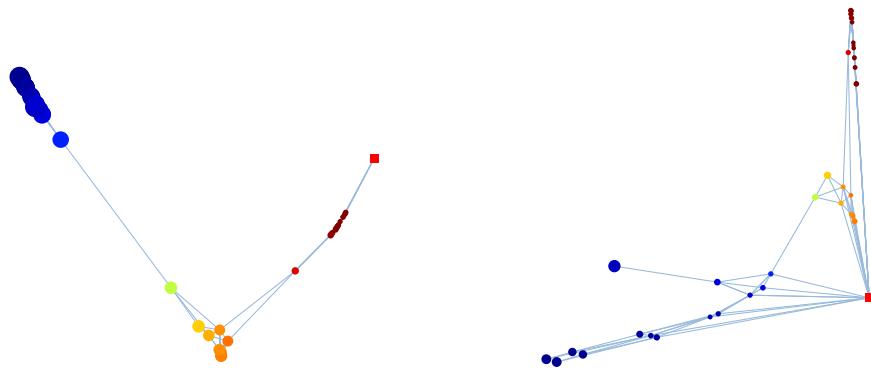


Figure 8.3: The visualization tool described in Section 8.7.2, applied here to the anchored synchronization tasks from Figure 8.1 (left with one anchor, right with three anchors), produces low-dimensional embeddings of synchronization graphs such that the distance between two nodes is large if their relative rotation is hard to estimate, and their distance to the origin (the anchors: red squares) is large if their individual rotation is hard to estimate.

loss of generality that the eigenvalues and eigenvectors are ordered such that the diagonal entries of D^\dagger are decreasing. Then,

$$(e_i - e_j)^\top \mathcal{L}_A^\dagger (e_i - e_j) = (e_i - e_j)^\top V D^\dagger V^\top (e_i - e_j) = \|x_i - x_j\|^2.$$

Thus, embedding the nodes at positions x_i realizes the ECTD in \mathbb{R}^N . Anchors, if any, are placed at the origin. An optimal embedding, in the sense of preserving the ECTD as well as possible, in a subspace of dimension $k < N$ is obtained by considering \tilde{X} : the first k rows of X . The larger the ratio $\sum_{\ell=1}^k \lambda_\ell^\dagger / \text{trace}(D^\dagger)$, the better the low-dimensional embedding captures the ECTD.

In the presence of anchors, if $j \in A$, then $\mathcal{L}_A^\dagger e_j = 0$ and $(e_i - e_j)^\top \mathcal{L}_A^\dagger (e_i - e_j) = (\mathcal{L}_A^\dagger)_{ii} = \|x_i\|^2 \approx \|\tilde{x}_i\|^2$. Hence, the embedded distance to the origin indicates how well a specific node can be estimated.

In practice, this embedding can be produced by computing the $m + k$ eigenvectors of \mathcal{L}_A with smallest eigenvalue, where $m = \max(1, |A|)$ is the number of zero eigenvalues to be discarded (assuming a connected graph). This computation can be conducted efficiently if the graph is structured, e.g., sparse.

8.7.3 A larger Fiedler value is better

We now focus on anchor-free synchronization. At large SNR, the anchor-free CRB (8.33) normalized by the number of independent rotations $N - 1$ reads:

$$\mathbb{E}\{\text{MSE}\} \triangleq \mathbb{E}\left\{\frac{1}{N-1}\text{dist}^2([\mathbf{R}], [\hat{\mathbf{R}}])\right\} \geq \frac{d^2}{N-1} \text{trace}(\mathcal{L}^\dagger), \quad (8.41)$$

where $\mathbb{E}\{\text{MSE}\}$ as defined is the *expected mean squared error* of an unbiased estimator $[\hat{\mathbf{R}}]$. This expression shows the limiting role of the trace of the pseudoinverse of the information-weighted Laplacian \mathcal{L} (8.23) of the measurement graph. This role has been established before for other synchronization problems for simpler groups and simpler noise models (Howard *et al.*, 2010). We now shed some light on this result by stating a few elementary consequences of it. Let

$$0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_N$$

denote the eigenvalues of \mathcal{L} , where $\lambda_2 > 0$ means the measurement graph is assumed connected.

The right-hand side of (8.41) in terms of the λ_i 's is given by:

$$\frac{d^2}{N-1} \text{trace}(\mathcal{L}^\dagger) = \frac{d^2}{N-1} \sum_{i=2}^N \frac{1}{\lambda_i} \leq \frac{d^2}{\lambda_2}.$$

The second eigenvalue λ_2 is known as the *Fiedler value* (or *algebraic connectivity*) of the information-weighted measurement graph. It is well known that the Fiedler value is low in the presence of bottlenecks in the graph and high in the presence of many, heavy spanning trees. The latter equation translates in the following intuitive statement: by increasing the Fiedler value of the measurement graph, one can force a lower CRB. Not surprisingly then, expander graphs are ideal for synchronization, since, by design, their Fiedler value λ_2 is bounded away from zero while simultaneously being sparse (Hoory *et al.*, 2006).

Notice that the Fiedler vector has zero mean (it is orthogonal to $\mathbf{1}_N$) and hence describes the horizontal vectors of maximum variance. It is thus also the first axis of the right plot in Figure 8.2.

8.7.4 $\text{trace}(\mathcal{L}^\dagger)$ plays a limiting role in synchronization

We continue to focus on anchor-free synchronization. The quantity $\text{trace}(\mathcal{L}^\dagger)$ appears naturally in CRB's for synchronization problems on groups. For

complete graphs and constant weight w , $\text{trace}(\mathcal{L}^\dagger) = \frac{N-1}{wN}$. Then, by (8.41),

$$\mathbb{E}\{\text{MSE}\} \geq \frac{d^2}{wN}. \quad (8.42)$$

If the measurement graph is sampled from a distribution of random graphs, $\text{trace}(\mathcal{L}^\dagger)$ becomes a random variable. We feel that the study of this random variable for various families of random graph models, such as Erdős-Rényi, small-world or scale-free graphs (Jamakovic & Uhlig, 2007) is a question of interest, probably best addressed using the language of random matrix theory.

Let us consider Erdős-Rényi graphs $G_{N,q}$ with N nodes and edge density $q \in (0, 1)$, that is, graphs such that any edge is present with probability q , independently from the other edges. Let all the edges have equal weight w . Let $\mathcal{L}_{N,q}$ be the Laplacian of a $G_{N,q}$ graph. The expected Laplacian is $\mathbb{E}\{\mathcal{L}_{N,q}\} = wq(NI_N - \mathbf{1}_{N \times N})$, which has eigenvalues $\lambda_1 = 0, \lambda_2 = \dots = \lambda_N = Nwq$. Hence, $\text{trace}(\mathbb{E}\{\mathcal{L}_{N,q}\}^\dagger) = \frac{N-1}{N} \frac{1}{wq}$. A more useful statement can be made using (Bryc *et al.*, 2006, Thm. 1.4) and (Ding & Jiang, 2010, Thm. 2). These theorems state that, asymptotically as N grows to infinity, all eigenvalues of $\mathcal{L}_{N,q}/N$ converge to wq (except of course for one zero eigenvalue). Consequently,

$$\lim_{N \rightarrow \infty} \mathbb{E}\left\{\text{trace}(\mathcal{L}_{N,q}^\dagger)\right\} = \frac{1}{wq} \text{ (in probability)}. \quad (8.43)$$

The expectation and concentration of the random variable $\text{trace}(\mathcal{L}_{N,q}^\dagger)$ is further investigated by Boumal & Cheng (2013). For large N , we use the approximation $\text{trace}(\mathcal{L}_{N,q}^\dagger) \approx 1/wq$. Then, by (8.41), for large N we have:

$$\mathbb{E}\{\text{MSE}\} \gtrsim \frac{d^2}{wqN}.$$

Notice how for fixed measurement quality w and density q , the lower bound on the expected MSE decreases with the number N of rotations to estimate.

8.7.5 Synchronization can withstand many outliers

Consider the mixture of Langevin distribution from Example 8.5 in the particular case $\kappa' = 0$, that is, an average fraction $1-p$ of measurements are sampled uniformly at random. The information weight $w(p) = w_n(\kappa, 0, p)$ for some fixed concentration $\kappa > 0$ is given by equations (8.25) for $n = 2$ and 3 respectively. A Taylor expansion around $p = 0$ shows that, when most measurements are outliers,

$$w(p) = a_{n,\kappa}p^2 + \mathcal{O}(p^3)$$

for some positive constant $a_{n,\kappa}$. Then, for $p \ll 1$, building upon (8.42) for complete graphs with i.i.d. measurements we get:

$$\mathbb{E}\{\text{MSE}\} \gtrsim \frac{d^2}{a_{n,\kappa} p^2 N}.$$

If one needs to get the right-hand side of this inequality down to a tolerance ε^2 , the probability p of a measurement not being an outlier needs to be at least as large as:

$$p_\varepsilon \triangleq \frac{d}{\sqrt{a_{n,\kappa}} \varepsilon} \frac{1}{\sqrt{N}}.$$

The $1/\sqrt{N}$ factor is the most interesting: it establishes that as the number of nodes increases, synchronization can withstand a larger fraction of outliers.

This result is to be put in perspective with the bound in (Singer, 2011, eq. (37)) for $n = 2$, $\kappa = \infty$, where it is shown that as soon as $p > 1/\sqrt{N}$, there is enough information in the measurements (on average) for the eigenvector method to do better than random synchronization (that analysis is also laid out in Section 5.3). It is also shown in the latter paper that, as $p^2 N$ goes to infinity, the correlation between the eigenvector estimator and the true rotations goes to 1. Similarly, we see here that as $p^2 N$ increases to infinity, the right-hand side of the CRB goes to zero. Our analysis further shows that the role of $p^2 N$ is tied to the problem itself (not to a specific estimation algorithm), and remains the same for $n > 2$ and in the presence of Langevin noise on the good measurements.

Building upon (8.43) for Erdős-Rényi graphs with N nodes and M edges, we define p_ε as:

$$p_\varepsilon \triangleq \frac{d}{\sqrt{a_{n,\kappa}} \varepsilon} \sqrt{\frac{N}{2M}}. \quad (8.44)$$

To conclude this remark, we provide numerically computable expressions for $a_{n,\kappa}$, $n = 2$ and 3 and give an example:

$$a_{2,\kappa} = \frac{\kappa^2}{\pi c_2^2(\kappa)} \int_0^\pi (1 - \cos 2\theta) \exp(4\kappa \cos \theta) d\theta,$$

$$a_{3,\kappa} = \frac{\kappa^2 e^{2\kappa}}{\pi c_3^2(\kappa)} \int_0^\pi (1 - \cos 2\theta)(1 - \cos \theta) \exp(4\kappa \cos \theta) d\theta.$$

As an example, we generate an Erdős-Rényi graph with $N = 2500$ nodes and edge density of 60% for synchronization of rotations in $\text{SO}(3)$ with i.i.d. noise following a mixture of Langevin with $\kappa = 7$ and $\kappa' = 0$. The CRB (8.41),

which requires complete knowledge of the graph to compute $\text{trace}(\mathcal{L}^\dagger)$, tells us that we need $p \geq 2.1\%$ to reach an accuracy level of $\varepsilon = 10^{-1}$ (for comparison, ε^2 is roughly 1000 times smaller than V_3 (Section 8.7.1). The simple formula (8.44), which can be computed quickly solely based on the graph statistics N and M , yields $p_\varepsilon = 2.2\%$.

8.8 Conclusions

In this chapter, we considered synchronization of rotations in \mathbb{R}^n as discussed in Chapter 5. We framed it as a Riemannian estimation problem for arbitrary n under a large family of noise models. We established formulas for the FIM and associated CRB's of synchronization together with interpretation and visualization tools for them in both the anchored and anchor-free scenarios. In the analysis of these bounds, we notably pointed out the high robustness of synchronization against random outliers and their random walk interpretation.

The Laplacian of the measurement graph plays the same role in bounds for synchronization of rotations as for synchronization of translations (see Section 7.4). Carefully checking the proof given in the present work, it is reasonable to speculate that the Laplacian would appear similarly in CRB's for synchronization on any Lie group, as long as we assume independence of noise affecting different measurements and some symmetry in the noise distribution. Such a generalization would in particular yield CRB's for synchronization on the special Euclidean group of rigid body motions, $\mathbb{R}^3 \rtimes \text{SO}(3)$. This group appears in the global registration problem addressed in (Chaudhury *et al.*, 2013) for example, as well as in the study of 3D scan registration in Section 5.6.

Because of the crucial role of the pseudoinverse of the Laplacian \mathcal{L}^\dagger of weighted graphs (and their traces) in the CRB's we established, it would be interesting to study efficient methods to compute such objects, see e.g. (Ho & Van Dooren, 2005; Lin *et al.*, 2009). Likewise, exploring the distribution of $\text{trace}(\mathcal{L}^\dagger)$ seen as a random variable for various models of random graphs should bring some insight as to which networks are naturally easy to synchronize. We study the case of Erdős-Rényi graphs in (Boumal & Cheng, 2013).

Chapter 9

Conclusions

This chapter concludes, for now, my investigation of optimization and estimation on manifolds. We first look back and summarize our achievements so far. Later in this chapter, we anticipate a few possible developments that might originate from or echo to the present work.

At the onset of this thesis work in late 2010, it was already clear that Riemannian optimization could have an important role to play in various areas of applied mathematics, as evidenced for example by the long applications list given in Section 3.3. By then, researchers in the field had already reached a stable understanding of the concepts required to deal with optimization on manifolds and of the main general-purpose algorithms, complete with analysis. However, we found that there was still a significant entrance barrier precluding more applied researchers from leveraging these tools, in good part because of the differential geometry prerequisites.

With the Manopt toolbox developed and publicized during this thesis, we contribute to lowering this barrier. In its present form, the toolbox makes it possible to rapidly assess the usefulness of Riemannian optimization for a given problem, with minimal knowledge of unconstrained nonlinear optimization, and little to no knowledge of differential geometry. The hope is that positive outcomes will encourage practitioners to learn about the underlying algorithms. Being a practical tool of general purpose, we believe Manopt has the potential to make an impact on a short-term horizon.

We crystallized our investigations around two applications and now briefly discuss the solutions we proposed for them.

For low-rank matrix completion, we found that Riemannian optimization offers a scalable algorithmic framework to attain accurate solutions in various controlled (synthetic) experiments and decent solutions on the Netflix dataset for recommendation systems. In the controlled experiments, we observed that our methods are competitive with, or even widely outperform,

the state of the art in the face of challenges such as bad conditioning or non uniform sampling. On the Netflix dataset though, one may argue that the quality of the obtained recommendations does not warrant the complexity of the method, in terms of code development and maintenance. However, the Netflix competition has taught us that the best (RMSE) results are obtained using a blend of many different predictors, so that any new method sufficiently fast and different from the other ones in use has the potential to contribute valuably to a blend. In this respect, it is interesting to note that the proposed preconditioner for our method—which is designed based on a drastically idealized matrix completion task—reduces the computation time it requires to make the predictions on the Netflix dataset by a factor of two. A usual suspect for the limited quality of the Netflix solution provided by our algorithms is their least-squares nature. This indeed typically leads to poor outlier rejection, which is often of prime importance on real data. Our method design, based on optimization over a single Grassmannian, heavily relies on the least-squares cost. As such, it does not lend itself to an easy adaptation for alternate costs. A possible extension would be to use any of our algorithms as a building block in an iteratively reweighted least-squares scheme, in an attempt to minimize a sum of errors rather than a sum of squared errors.

For synchronization of rotations, we found that Riemannian optimization offers a flexible option to incorporate a noise model in the estimation, which we did to capture the presence of outliers in the data. The resulting optimization task presents poor quality local optimizers, as demonstrated by Figure 5.2 where using a random initial guess leads to catastrophic estimation errors. On the latter figure, it similarly appears that the eigenvector method constitutes an ideal initial guess, as it is both simple and fast to compute and it enables our Riemannian MLE procedure to achieve excellent accuracy. We view this observation as a major incentive to pursue the study of combinations of tractable relaxations with Riemannian refinement algorithms. On real data such as the Lucy dataset, we found that alternatively estimating the rotations and the noise distribution provides a fast and accurate overall method (dubbed MLE+) which does not require exact knowledge of the noise parameters. Furthermore, the accuracy of the solutions found on the Lucy dataset validate (to some extent) the usefulness of the mixture of Langevin noise model we assumed a priori.

In the second part of the manuscript, we focused on fundamental bounds on the accuracy one can hope for in solving an estimation problem. In particular, we focused on Cramér-Rao bounds. Such bounds were already derived for the low-rank matrix completion problem by Tang & Nehorai (2011a,b) using standard tools, so that we directed our attention to synchronization of rotations. In so doing, we found that existing work by Smith

(2005) regarding CRB's on manifolds constituted a firm reference to anchor our exploration. We first specialized these bounds to the case of Riemannian submanifolds and Riemannian quotient manifolds, purposefully simplifying their application to, respectively, the anchored and anchor-free versions of synchronization.

The main finding is that, under some assumptions, the CRB's for synchronization of rotations are dictated by the Laplacian of the measurement graph. The role of the graph Laplacian in the bounds and their ensuing interpretation in terms of random walks brings appreciable insight to the synchronization problem. This insight is furthermore validated by the empirical observation in Chapter 5 that the CRB's seem to be attainable in non trivial scenarios. One example of a lesson taken from the CRB's under the mixture of Langevin noise model is that synchronization is intrinsically resilient to outliers.

The remarkable structure of the CRB's for synchronization originates in three key properties of the problem at hand. First, our assumption that noise on different measurements is *independent* induces a block sparsity structure for the Fisher information matrix compatible with that of the Laplacian. Second, the assumption that noise is distributed *isotropically* leads to each of these blocks being proportional to the identity matrix. Together with the strong *symmetry* of the space of rotations, these properties lead to the FIM being independent of the rotations to estimate. This latter point crucially simplifies the interpretation of the CRB's. Some of these properties remain valid on broader classes of synchronization problems, which we see as an incentive to generalize the established results.

Perspectives

Optimization and estimation on manifolds are blossoming fields. As we argued in this thesis, tools are readily available to solve and analyze data processing problems on manifolds, and we contributed to some of them. But as is customary with such research endeavors, more questions are left unanswered at the end of the journey than at its onset.

To solve nonconvex optimization problems whose search spaces admit a Riemannian structure, we advocated combining tractable relaxations (when available) with a Riemannian optimization refinement procedure. The burning question is whether it is possible to prove that the refinement procedure indeed improves the solution, not only with respect to the cost function (which should be the case) but more importantly with respect to the quality of the solution, for which the cost function may only be an imperfect proxy. The added knowledge that the initial guess does not exceed certain error bounds might be a decisive piece of information to conduct the

proofs. More importantly, can we put together proof techniques to that effect, which could become useful in more than one context? The OptSpace algorithm (Keshavan *et al.*, 2010) is one example where such analysis is successfully derived and could constitute an entry point to this enticing research question.

When Riemannian optimization problems admit effective semidefinite relaxations (SDR's), one may wonder how the strong geometry of the original problem affects (or restricts) the structure of these SDR's. For example, in a paper by Journée *et al.* (2010b), it is shown how certain such convex programs can be solved efficiently, precisely using optimization on manifolds as the central tool (see also Section 3.3.2). This raises the question of whether all-manifold solutions can be proposed more generally, to both solve the SDR and execute the refinement. In a more recent paper, Bandeira *et al.* (2013a) show how a large class of problems (which includes synchronization of rotations and max-cut as particular cases) admits a polynomial-time approximation algorithm they call orthogonal-cut. The latter achieves a guaranteed approximation ratio following the resolution of an SDR. Because this SDR descends from a Riemannian optimization problem, we suspect it is amenable to analysis similar to that of Journée *et al.* (2010b) and we plan to investigate this lead in future work.

An observation more directly focused on Riemannian optimization in its own right is that there is currently a relative absence of practical algorithms to address nonsmooth optimization on manifolds. Nonsmooth cost functions, specifically piecewise smooth cost functions, occur naturally in a number of applications. An example was given in Section 3.3.3 about sphere packing on the sphere, where the proposed solution entails a smoothing of the cost. Another example is the Weiszfeld algorithm for synchronization (Hartley *et al.*, 2011), where it is not the sum of squared errors which is minimized but the sum of unsquared errors, akin to the LUD approach (see Section 5.5.2). Such cost functions have been observed time and again to handle outliers in data far better than squared losses. While the nonsmoothness may not be critical for outlier rejection, the theory of compressed sensing insists it is instrumental when sparse solutions are targeted, hence ruling out smoothing-based methods. Dirr *et al.* (2007) have proposed a subgradient approach to nonsmooth Riemannian optimization problems, with some success. Nevertheless, the practical implementation of subgradient techniques remains tedious and we look forward to the development of more practical algorithms on that front, hopefully very soon.

Riemannian optimization, as presented in Chapter 3, applies to smooth optimization problems defined over any finite-dimensional Riemannian manifold. In practice though, these tools are only applied on very special manifolds, with strong symmetries. A more far-reaching question then would be

to assess how much of the success of Riemannian optimization lives and dies with these rich structures we are granted in our numerical investigations, yet do not acknowledge in the general theory.

Pertaining to the estimation bounds established for synchronization of rotations, it is natural to conjecture that the role of the pseudoinverse of the Laplacian of the measurement graph is not tied to the specific case of rotations, but is more fundamentally tied to the structure of synchronization in general. As we discussed, synchronization can be thought of as the generic task of estimating elements g_1, \dots, g_N belonging to a group G , based on measurements of relative elements $g_i g_j^{-1}$. Regardless of the group, the topology of the graph built from these data—with N nodes and an edge between two nodes if a relative measurement about them is available—is expected to play a central role. For continuous (Lie) groups G , we expect the analysis via Cramér-Rao bounds (CRB's) to carry over in a generalized version of the statements in this thesis. For discrete groups (e.g., $\mathbb{Z}_2 = \{\pm 1\}$ (Cucuringu, 2013) or the group of permutations (Huang & Guibas, 2013)), the CRB analysis is no longer appropriate, but it seems plausible that other types of bounds (such as minimax bounds for example) would exhibit a similar structure.

The unresolved questions of today are the opportunities of tomorrow, and I look forward to the answers to come.

Appendix A

Integration over $\mathrm{SO}(n)$

This appendix details how to execute the integrals over the group of rotations $\mathrm{SO}(n)$ (5.1) which appear in chapters 5 and 8. Let μ denote the Haar measure over $\mathrm{SO}(n)$ (Section 8.3). We are interested in integrating $g: \mathrm{SO}(n) \rightarrow \mathbb{R}$ over its domain. For a general integrand g , computing this integral may require parameterizing $\mathrm{SO}(n)$ in order to reduce it to a classical integral over a domain of \mathbb{R}^d , with $d = \dim \mathrm{SO}(n)$. In general, this is not convenient. Fortunately, when the integrand is a class function we are in a position to use the Weyl integration formula (Bump, 2004, Exercise 18.1–2).

Definition A.1 (class function). *A function $g: \mathrm{SO}(n) \rightarrow \mathbb{R}$ is a class function if for all $Z, Q \in \mathrm{SO}(n)$, it holds that $g(Z) = g(QZQ^\top)$, that is, g is invariant under conjugation.*

Weyl's formula reduces integrals on $\mathrm{SO}(n)$ to classical integrals over tori of dimension $\lfloor n/2 \rfloor$, typically more amenable to analytical or numerical evaluation. For $n = 2, 3$, that is a classical integral on the interval $[-\pi, \pi]$:

$$\begin{aligned} \int_{\mathrm{SO}(2)} g(Z) d\mu(Z) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} g \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} d\theta, \\ \int_{\mathrm{SO}(3)} g(Z) d\mu(Z) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} g \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} (1 - \cos \theta) d\theta. \end{aligned} \quad (\text{A.1})$$

For $n = 4$, Weyl's formula is a double integral:

$$\begin{aligned} \int_{\mathrm{SO}(4)} g(Z) d\mu(Z) &= \\ \frac{1}{4(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \tilde{g}(\theta_1, \theta_2) \cdot |e^{i\theta_1} - e^{i\theta_2}|^2 \cdot |e^{i\theta_1} - e^{-i\theta_2}|^2 d\theta_1 d\theta_2, \end{aligned} \quad (\text{A.2})$$

with

$$\tilde{g}(\theta_1, \theta_2) \triangleq g \left(\text{diag} \left(\begin{pmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{pmatrix}, \begin{pmatrix} \cos \theta_2 & -\sin \theta_2 \\ \sin \theta_2 & \cos \theta_2 \end{pmatrix} \right) \right). \quad (\text{A.3})$$

In the sections of this appendix, these formulas are leveraged to obtain computable expressions for some of the coefficients that appear in analyzing the synchronization of rotations problem.

Since we assume all probability distribution functions in this work are spectral functions (Assumption 8.3) and since all spectral functions are, a fortiori, class functions, these tools apply often in this work's setting. The converse is also true for $\text{SO}(2k+1)$ but not for $\text{SO}(2k)$. Indeed, assume n is odd and let $g: \text{SO}(n) \rightarrow \mathbb{R}$ be a class function. Let $Z \in \text{SO}(n)$ and $Q \in \text{O}(n)$. Certainly, if $\det(Q) = 1$, then $g(QZQ^\top) = g(Z)$ since g is a class function. On the other hand, if $\det(Q) = -1$, then, because n is odd, $\det(-Q) = 1$ and $g(Z) = g((-Q)Z(-Q)^\top) = g(QZQ^\top)$. Thus, g is a spectral function. For n even, this is not true in general. Consider $n = 2$ and let $g([\cos \theta, -\sin \theta; \sin \theta, \cos \theta]) := \sin \theta$. Certainly, g is a class function since all functions on $\text{SO}(2)$ are class functions owing to the commutativity of in-plane rotations. But g is not a spectral function since $g(Z) = -g(Z^\top)$ even though Z and Z^\top share the same eigenvalues.

The modified Bessel functions of the first kind (Wolfram, 2001), defined by the identity

$$I_\nu(x) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{x \cos \theta} \cos(\nu \theta) d\theta, \quad (\text{A.4})$$

will come in handy. Beware that these functions scale exponentially with their input x . It is often numerically sound to compute $e^{-x} I_\nu(x)$ instead, which is possible with many numerical packages. For example, in Matlab, use `besseli(ν , x , 1)`.

A.1 Langevin density normalization

We now compute the normalization coefficient $c_n(\kappa)$ for $n = 4$ (8.13) that appears in the Langevin probability density function (5.4). For generic n , the necessary manipulations are very similar to the developments in this section and formulas for $n = 2, 3$ are provided (8.11)(8.12). The coefficient $c_n(\kappa)$ is defined by (8.10):

$$c_n(\kappa) = \int_{\text{SO}(n)} \exp(\kappa \text{trace}(Z)) d\mu(Z),$$

In particular, $c_n(0) = 1$. The integrand, $g(Z) = \exp(\kappa \operatorname{trace}(Z))$, is a class function. Thus, by formula (A.2),

$$c_4(\kappa) = \frac{1}{4(2\pi)^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \tilde{g}(\theta_1, \theta_2) \cdot |e^{i\theta_1} - e^{i\theta_2}|^2 \\ \cdot |e^{i\theta_1} - e^{-i\theta_2}|^2 d\theta_1 d\theta_2, \quad (\text{A.5})$$

with \tilde{g} as in (A.3). This reduces the problem to a classical integral over the square—or really the torus— $[-\pi, \pi] \times [-\pi, \pi]$. Evaluating \tilde{g} is straightforward:

$$\tilde{g}(\theta_1, \theta_2) = \exp(2\kappa \cdot [\cos \theta_1 + \cos \theta_2]). \quad (\text{A.6})$$

Using trigonometric identities, we also get:

$$|e^{i\theta_1} - e^{i\theta_2}|^2 \cdot |e^{i\theta_1} - e^{-i\theta_2}|^2 \\ = 4(1 - \cos(\theta_1 - \theta_2))(1 - \cos(\theta_1 + \theta_2)) \\ = 4(1 - \cos(\theta_1 - \theta_2) - \cos(\theta_1 + \theta_2) + \cos(\theta_1 - \theta_2)\cos(\theta_1 + \theta_2)) \\ = 4 \left(1 - 2\cos \theta_1 \cos \theta_2 + \frac{1}{2}(\cos 2\theta_1 + \cos 2\theta_2) \right). \quad (\text{A.7})$$

Each cosine factor now only depends on one of the angles. Plugging (A.6) and (A.7) in (A.5) and using Fubini's theorem, we get:

$$c_4(\kappa) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{2\kappa \cos \theta_1} \cdot h(\theta_1) d\theta_1, \quad (\text{A.8})$$

with:

$$h(\theta_1) = \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{2\kappa \cos \theta_2} \left(1 + \frac{1}{2} \cos 2\theta_1 - 2\cos \theta_1 \cos \theta_2 + \frac{1}{2} \cos 2\theta_2 \right) d\theta_2.$$

Express h in terms of Bessel functions (A.4):

$$h(\theta_1) = \left(1 + \frac{1}{2} \cos 2\theta_1 \right) \cdot I_0(2\kappa) - 2\cos \theta_1 \cdot I_1(2\kappa) + \frac{1}{2} \cdot I_2(2\kappa). \quad (\text{A.9})$$

Plugging (A.9) in (A.8) and resorting to Bessel functions again, we finally obtain the practical formula (8.13) for $c_4(\kappa)$:

$$c_4(\kappa) = \left[I_0(2\kappa) + \frac{1}{2} I_2(2\kappa) \right] \cdot I_0(2\kappa) - 2I_1(2\kappa) \cdot I_1(2\kappa) + \frac{1}{2} I_0(2\kappa) \cdot I_2(2\kappa) \\ = I_0(2\kappa)^2 - 2I_1(2\kappa)^2 + I_0(2\kappa)I_2(2\kappa).$$

In (Chikuse, 2003, Appendix A.6), Chikuse describes how the normalization coefficients for Langevin distributions on $\text{O}(n)$ can be expressed in terms of hypergeometric functions with matrix arguments. One advantage of this method is that it generalizes to non-isotropic Langevin's. The method we demonstrated here, in comparison, is tailored for our need (isotropic Langevin's on $\text{SO}(n)$) and yields simple expressions in terms of Bessel functions—which are readily available in Matlab for example.

A.2 Mixture of Langevin information weight

In deriving the Fisher information matrix for synchronization of rotations (Theorem 8.5), the information weight w (8.22) appears and needs to be computed:

$$w = \mathbb{E} \left\{ \|\text{grad log } f(Z)\|^2 \right\} = \int_{\text{SO}(n)} \|\text{grad log } f(Z)\|^2 f(Z) d\mu(Z).$$

Under the mixture of Langevin noise model from Chapter 5, the pdf f (4.18) is defined by

$$\begin{aligned} f: \text{SO}(n) &\rightarrow \mathbb{R}^+, & f(Z) &= p \ell_\kappa(Z) + (1-p) \ell_{\kappa'}(Z), \\ \ell_\kappa: \text{SO}(n) &\rightarrow \mathbb{R}^+, & \ell_\kappa(Z) &= \frac{1}{c_n(\kappa)} \exp(\kappa \text{trace}(Z)), \end{aligned}$$

where $\kappa, \kappa' \geq 0$ and $p \in [0, 1]$ are some fixed parameters and $c_n(\kappa)$ (8.10) is the normalization constant discussed in the previous section. This is the model addressed in Example 8.5.

From Section 5.4.2, it follows easily that

$$\|\text{grad log } f(Z)\|^2 = g^2(Z) \left\| \frac{Z - Z^\top}{2} \right\|_{\text{F}}^2,$$

with g as defined by equation (5.31):

$$g(Z) = \frac{p\kappa \ell_\kappa(Z) + (1-p)\kappa' \ell_{\kappa'}(Z)}{f(Z)}.$$

Thus, computing w reduces to evaluating this integral:

$$w = w_n(\kappa, \kappa', p) = \int_{\text{SO}(n)} \frac{(p\kappa \ell_\kappa(Z) + (1-p)\kappa' \ell_{\kappa'}(Z))^2}{p \ell_\kappa(Z) + (1-p) \ell_{\kappa'}(Z)} \left\| \frac{Z - Z^\top}{2} \right\|_{\text{F}}^2 d\mu(Z).$$

Let $h(Z)$ denote the integrand, i.e., $w = \int_{\text{SO}(n)} h(Z) d\mu(Z)$. Notice that h is a class function (Definition A.1). Then, applying Weyl's formula for

$n = 2$ (A.1):

$$w_2(\kappa, \kappa', p) = \int_{\text{SO}(2)} h(Z) d\mu(Z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} h(Z_\theta) d\theta,$$

$$Z_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}.$$

Observing that $\ell_\kappa(Z_\theta) = \exp(2\kappa \cos \theta)/c_2(\kappa)$ and $\|(Z_\theta - Z_\theta^\top)/2\|_{\text{F}}^2 = 2 \sin^2 \theta$ makes it possible to evaluate this integral numerically. More interestingly, for $n = 3$, it holds that

$$w_3(\kappa, \kappa', p) = \int_{\text{SO}(3)} h(Z) d\mu(Z) = \frac{1}{2\pi} \int_{-\pi}^{\pi} h(Z_\theta) (1 - \cos \theta) d\theta,$$

$$Z_\theta = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Again, $\ell_\kappa(Z_\theta) = \exp(\kappa[1 + 2 \cos \theta])/c_3(\kappa)$ and $\|(Z_\theta - Z_\theta^\top)/2\|_{\text{F}}^2 = 2 \sin^2 \theta$ make it possible to evaluate this integral numerically.

Explicit formulas in terms of Bessel functions for $p = 1$ appear in Example 8.4. Numerically integrable formulas for $\kappa' = 0$ appear in Example 8.5.

Appendix B

CRB's for synchronization of rotations: proof details

This appendix hosts technical details of the proof for the Cramér-Rao bounds of synchronization of rotations, in Chapter 8.

B.1 Proof of two properties of G_{ij}

Recall the definition of $G_{ij} : \text{SO}(n) \rightarrow \mathfrak{so}(n)$ (8.14) introduced in Section 8.4:

$$G_{ij}(Z) = [\text{grad } \log f_{ij}(Z)]^\top Z.$$

We now establish two properties of this mapping, namely that $G_{ij}(QZQ^\top) = QG_{ij}(Z)Q^\top$ and that $G_{ij}(Z^\top) = -G_{ij}(Z)$. Let us introduce a few functions:

$$\begin{aligned} g : \text{SO}(n) &\rightarrow \mathbb{R} : Z \mapsto g(Z) = \log f_{ij}(Z), \\ h_1 : \text{SO}(n) &\rightarrow \text{SO}(n) : Z \mapsto h_1(Z) = QZQ^\top, \\ h_2 : \text{SO}(n) &\rightarrow \text{SO}(n) : Z \mapsto h_2(Z) = Z^\top. \end{aligned}$$

Notice that because of Assumption 8.3 (f_{ij} is only a function of the eigenvalues of its argument), we have $g \circ h_i \equiv g$ for $i = 1, 2$. Hence,

$$\text{grad } g(Z) = \text{grad}(g \circ h_i)(Z) = (\text{D}h_i(Z))^* [\text{grad } g(h_i(Z))], \quad (\text{B.1})$$

where $(\text{D}h_i(Z))^*$ denotes the adjoint of the differential $\text{D}h_i(Z)$, defined by

$$\forall H_1, H_2 \in T_Z \text{SO}(n), \quad \langle \text{D}h_i(Z)[H_1], H_2 \rangle = \langle H_1, (\text{D}h_i(Z))^*[H_2] \rangle.$$

The rightmost equality of (B.1) follows from the chain rule. Indeed, starting with the definition of gradient, we have, for all $H \in T_Z \text{SO}(n)$,

$$\begin{aligned}\langle \text{grad}(g \circ h_i)(Z), H \rangle &= D(g \circ h_i)(Z)[H] \\ &= Dg(h_i(Z))[Dh_i(Z)[H]] \\ &= \langle \text{grad } g(h_i(Z)), Dh_i(Z)[H] \rangle \\ &= \langle (Dh_i(Z))^* [\text{grad } g(h_i(Z))], H \rangle.\end{aligned}$$

Let us compute the differentials of the h_i 's and their adjoints:

$$\begin{aligned}Dh_1(Z)[H] &= QHQ^\top, & (Dh_1(Z))^*[H] &= Q^\top HQ, \\ Dh_2(Z)[H] &= H^\top, & (Dh_2(Z))^*[H] &= H^\top.\end{aligned}$$

Plugging this in (B.1), we find two identities (one for h_1 and one for h_2):

$$\begin{aligned}\text{grad } \log f_{ij}(Z) &= Q^\top [\text{grad } \log f_{ij}(QZQ^\top)]Q, \\ \text{grad } \log f_{ij}(Z) &= [\text{grad } \log f_{ij}(Z^\top)]^\top.\end{aligned}$$

The desired result about the G_{ij} 's now follows easily. For any $Q \in O(n)$,

$$\begin{aligned}G_{ij}(QZQ^\top) &= [\text{grad } \log f_{ij}(QZQ^\top)]^\top QZQ^\top \\ &= [Q\text{grad } \log f_{ij}(Z)Q^\top]^\top QZQ^\top \\ &= QG_{ij}(Z)Q^\top;\end{aligned}\tag{B.2}$$

and similarly:

$$\begin{aligned}G_{ij}(Z^\top) &= [\text{grad } \log f_{ij}(Z^\top)]^\top Z^\top \\ &= \text{grad } \log f_{ij}(Z)Z^\top \\ &= ZG_{ij}^\top(Z)Z^\top \\ &= -ZG_{ij}(Z)Z^\top \\ &= -G_{ij}(Z),\end{aligned}$$

where we used that $G_{ij}(Z)$ is skew-symmetric and we used (B.2) for the last equality.

B.2 Proof of Lemma 8.3

Lemma 8.3 essentially states that, given two orthogonal, same-norm vectors E and E' in $\mathfrak{so}(n)$, there exists a rotation which maps E to E' . Applying that same rotation to E' (loosely, rotating by an additional 90°) recovers $-E$. This fact is obvious if we may use any rotation on the subspace $\mathfrak{so}(n)$. The set of rotations on $\mathfrak{so}(n)$ has dimension $d(d-1)/2$, with $d = \dim \mathfrak{so}(n) = n(n-1)/2$. In contrast, for the proof of Lemma 8.4 to go through, we need to restrict ourselves to rotations of $\mathfrak{so}(n)$ which can be written as $\Omega \mapsto P^\top \Omega P$, with $P \in O(n)$ orthogonal. We thus have only d degrees of freedom. The purpose of the present lemma is to show that this can still be done if we further restrict the vectors E and E' as prescribed in Lemma 8.3.

Proof. We give a constructive proof, distinguishing among different cases.

1. $\{i, j\} \cap \{k, \ell\} = \emptyset$. Construct T as the identity I_n with columns i and k swapped, as well as columns j and ℓ . Construct S as I_n with $S_{ii} := -1$. By construction, it holds that $T^\top ET = E'$, $T^\top E'T = E$, $SES = -E$ and $SE'S = E'$. Set $P = TS$ to conclude: $P^\top EP = ST^\top ETS = SE'S = E'$, $P^\top E'P = ST^\top E'TS = SES = -E$.
2. $i = k, j \neq \ell$. Construct T as the identity I_n with columns j and ℓ swapped. Construct S as I_n with $S_{jj} := -1$. The same properties will hold. Set $P = TS$ to conclude.
3. $i = \ell, j \neq k$. Construct T as the identity I_n with columns j and k swapped and with $T_{ii} := -1$. Construct S as I_n with $S_{jj} := -1$. Set $P = TS$ to conclude.
4. $j = k$ or $j = \ell$. The same construction goes through. \square

Bibliography

- Absil, P.-A., & Gallivan, K.A. 2006. Joint Diagonalization on the Oblique Manifold for Independent Component Analysis. *In: Acoustics, Speech and Signal Processing, ICASSP 2006. IEEE International Conference on*, vol. 5.
- Absil, P.-A., Baker, C. G., & Gallivan, K. A. 2007. Trust-region methods on Riemannian manifolds. *Found. Comput. Math.*, **7**(3), 303–330.
- Absil, P.-A., Mahony, R., & Sepulchre, R. 2008. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press.
- Absil, P.-A., Amodei, L., & Meyer, G. 2013. Two Newton methods on the manifold of fixed-rank matrices endowed with Riemannian quotient geometries. *Computational Statistics*, 1–22.
- Afshari, H., Jacques, L., Bagnato, L., Schmid, A., Vandergheynst, P., & Leblebici, Y. 2013. The PANOPTIC camera: a plenoptic sensor with real-time omnidirectional capability. *Journal of Signal Processing Systems*, **70**(3), 305–328.
- Amari, S. 1999. Natural gradient learning for over- and under-complete bases in ICA. *Neural Computation*, **11**(8), 1875–1883.
- Amari, S., & Nagaoka, H. 2007. *Methods of information geometry*. Vol. 191. American Mathematical Society.
- Arie-Nachimson, M., Kovalsky, S.Z., Kemelmacher-Shlizerman, I., Singer, A., & Basri, R. 2012. Global motion estimation from point matches. *Pages 81–88 of: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE.
- Ash, J.N., & Moses, R.L. 2007. Relative and absolute errors in sensor network localization. *Pages 1033–1036 of: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007.*, vol. 2. IEEE.

- Bai, Z. D., & Yin, Y. Q. 1988. Necessary and Sufficient Conditions for Almost Sure Convergence of the Largest Eigenvalue of a Wigner Matrix. *The Annals of Probability*, **16**(4), 1729–1741.
- Balzano, L., Nowak, R., & Recht, B. 2010. Online identification and tracking of subspaces from highly incomplete information. *Pages 704–711 of: Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*. IEEE.
- Bandeira, A.S., Kennedy, C., & Singer, A. 2013a. Approximating the little Grothendieck problem over the orthogonal group. *arXiv preprint arXiv:1308.5207*.
- Bandeira, A.S., Singer, A., & Spielman, D.A. 2013b. A Cheeger Inequality for the Graph Connection Laplacian. *to appear in SIAM Journal on Matrix Analysis and Applications (SIMAX)*.
- Barooah, P., & Hespanha, J.P. 2007. Estimation on graphs from relative measurements. *Control Systems Magazine, IEEE*, **27**(4), 57–74.
- Bell, R.M., Koren, Y., & Volinsky, C. 2008. The BellKor 2008 solution to the Netflix prize. *Statistics Research Department at AT&T Research*.
- Bellet, A., Habrard, A., & Sebban, M. 2013. A Survey on Metric Learning for Feature Vectors and Structured Data. *arXiv preprint arXiv:1306.6709*.
- Ben-Haim, Z., & Eldar, Y.C. 2009. On the Constrained Cramér-Rao Bound With a Singular Fisher Information Matrix. *Signal Processing Letters, IEEE*, **16**(6), 453–456.
- Bennett, J., & Lanning, S. 2007. The Netflix prize. *In: Proceedings of KDD Cup and Workshop*.
- Bernstein, D.S. 2009. *Matrix mathematics: theory, facts, and formulas*. Princeton University Press.
- Bonnabel, S. 2013. Stochastic Gradient Descent on Riemannian Manifolds. *Automatic Control, IEEE Transactions on*, **58**(9), 2217–2229.
- Boothby, W.M. 1986. *An introduction to differentiable manifolds and Riemannian geometry*. Pure and Applied Mathematics, vol. 120. Elsevier.
- Bouldin, R. 1973. The pseudo-inverse of a product. *SIAM Journal on Applied Mathematics*, **24**(4), 489–495.

- Boumal, N. 2013a. Interpolation and Regression of Rotation Matrices. *Pages 345–352 of:* Nielsen, F., & Barbaresco, F. (eds), *Geometric Science of Information.* Lecture Notes in Computer Science, vol. 8085. Springer Berlin Heidelberg.
- Boumal, N. 2013b. On Intrinsic Cramér-Rao Bounds for Riemannian Submanifolds and Quotient Manifolds. *Signal Processing, IEEE Transactions on*, **61**(7), 1809–1821.
- Boumal, N., & Absil, P.-A. 2011a. A discrete regression method on manifolds and its application to data on SO(n). *Pages 2284–2289 of:* *Proceedings of the 18th IFAC World Congress (Milan)*, vol. 18.
- Boumal, N., & Absil, P.-A. 2011b. Discrete regression methods on the cone of positive-definite matrices. *Pages 4232–4235 of:* *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE.
- Boumal, N., & Absil, P.-A. 2011c. RTRMC: A Riemannian trust-region method for low-rank matrix completion. *Pages 406–414 of:* Shawe-Taylor, J., Zemel, R.S., Bartlett, P., Pereira, F.C.N., & Weinberger, K.Q. (eds), *Advances in Neural Information Processing Systems 24 (NIPS)*.
- Boumal, N., & Absil, P.-A. 2012. *Low-rank matrix completion via trust-regions on the Grassmann manifold.* Available on Optimization Online.
- Boumal, N., & Cheng, X. 2013. Expected performance bounds for estimation on graphs from random relative measurements. *Arxiv preprint arXiv:1307.6398*.
- Boumal, N., Singer, A., Absil, P.-A., & Blondel, V.D. 2013a. *Cramér-Rao bounds for synchronization of rotations.* to appear in *Information and Inference: A Journal of the IMA*.
- Boumal, N., Singer, A., & Absil, P.-A. 2013b. Robust estimation of rotations from relative measurements by maximum likelihood. *Proceedings of the 52nd Conference on Decision and Control, CDC 2013*.
- Boumal, N., Mishra, B., Absil, P.-A., & Sepulchre, R. 2014. Manopt: a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research.* Accepted for publication.
- Brookes, M. 2005. The matrix reference manual. *Imperial College London*.
- Bryc, W., Dembo, A., & Jiang, T. 2006. Spectral measure of large random Hankel, Markov and Toeplitz matrices. *The Annals of Probability*, **34**(1), 1–38.

- Bump, D. 2004. *Lie groups.* Graduate Texts in Mathematics, vol. 225. Springer.
- Cai, J.F., Candès, E.J., & Shen, Z. 2010. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM Journal on Optimization*, **20**(4), 1956–1982.
- Candès, E.J., & Recht, B. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, **9**(6), 717–772.
- Candès, E.J., Strohmer, T., & Voroninski, V. 2012. PhaseLift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*.
- Capitaine, M., Donati-Martin, C., & Féral, D. 2009. The largest eigenvalues of finite rank deformation of large Wigner matrices: convergence and nonuniversality of the fluctuations. *The Annals of Probability*, **37**(1), 1–47.
- Carmona, M., Michel, O., Lacoume, J.-L., Sprynski, N., & Nicolas, B. 2011. Algorithm for Sensor Network Attitude Problem. *arXiv preprint arXiv:1104.1317*.
- Chang, C., & Sahai, A. 2006. Cramér-Rao-Type Bounds for Localization. *EURASIP Journal on Advances in Signal Processing*, **2006**(1), 1–13.
- Chatterjee, S. 2012. Matrix estimation by universal singular value thresholding. *Arxiv preprint arXiv:1212.1247*.
- Chaudhury, K.N., Khoo, Y., Singer, A., & Cowburn, D. 2013. Global registration of multiple point clouds using semidefinite programming. *Arxiv preprint arXiv:1306.5226*.
- Chavel, I. 1993. *Riemannian geometry: a modern introduction.* Cambridge Tracts in Mathematics, vol. 108. Cambridge University Press.
- Chikuse, Y. 2003. *Statistics on special manifolds.* Lecture Notes in Statistics, vol. 174. Springer.
- Chiuso, A., Picci, G., & Soatto, S. 2008. Wide-Sense Estimation on the Special Orthogonal Group. *Communications in Information & Systems*, **8**(3), 185–200.
- Cohn, H., & Kumar, A. 2007. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, **20**(1), 99–148.
- Conn, A.R., Gould, N.I.M., & Toint, P.L. 2000. *Trust-region methods.* MPS-SIAM Series on Optimization, vol. 1. Society for Industrial Mathematics.

- Cucuringu, M. 2013. *Synchronization over Z_2 and community detection in bipartite multiplex networks*. Accessed on the author's personal home page.
- Cucuringu, M., Singer, A., & Cowburn, D. 2012a. Eigenvector synchronization, graph rigidity and the molecule problem. *Information and Inference: A Journal of the IMA*, **1**(1), 21–67.
- Cucuringu, M., Lipman, Y., & Singer, A. 2012b. Sensor network localization by eigenvector synchronization over the Euclidean group. *ACM Transactions on Sensor Networks*, **8**(3), 19:1–19:42.
- Dai, W., Milenkovic, O., & Kerman, E. 2011. Subspace evolution and transfer (SET) for low-rank matrix completion. *Signal Processing, IEEE Transactions on*, **59**(7), 3120–3132.
- Dai, W., Kerman, E., & Milenkovic, O. 2012. A Geometric Approach to Low-Rank Matrix Completion. *Information Theory, IEEE Transactions on*, **58**(1), 237–247.
- Demanet, L., & Jugnon, V. 2013. Convex recovery from interferometric measurements. *arXiv preprint arXiv:1307.6864*.
- Diaconis, P., & Shahshahani, M. 1987. The subgroup algorithm for generating uniform random variables. *Probability in the Engineering and Informational Sciences*, **1**(1), 15–32.
- Ding, X., & Jiang, T. 2010. Spectral distributions of adjacency and Laplacian matrices of random graphs. *The Annals of Applied Probability*, **20**(6), 2086–2117.
- Dirr, G., Helmke, U., & Lageman, C. 2007. Nonsmooth Riemannian optimization with applications to sphere packing and grasping. *Pages 29–45 of: Lagrangian and Hamiltonian Methods for Nonlinear Control 2006*, vol. 366. Springer.
- do Carmo, M.P. 1992. *Riemannian geometry*. Mathematics: Theory & Applications. Boston, MA: Birkhäuser Boston Inc. Translated from the second Portuguese edition by Francis Flaherty.
- Doyle, PG, & Snell, JL. 2000. Random walks and electric networks. *arXiv preprint math.PR/0001057*.
- Edelman, A., Arias, T.A., & Smith, S.T. 1998. The geometry of algorithms with orthogonality constraints. *SIAM journal on Matrix Analysis and Applications*, **20**(2), 303–353.

- Fornasier, M., Rauhut, H., & Ward, R. 2011. Low-rank Matrix Recovery via Iteratively Reweighted Least Squares Minimization. *SIAM Journal on Optimization*, **21**(4), 1614–1640.
- Gallot, S., Hulin, D., & LaFontaine, J. 2004. *Riemannian geometry*. Springer Verlag.
- Gilbert, J.C., & Nocedal, J. 1992. Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on Optimization*, **2**(1), 21–42.
- Gillis, N., & Glineur, F. 2011. Low-Rank Matrix Approximation with Weights or Missing Data Is NP-Hard. *SIAM Journal on Matrix Analysis and Applications*, **32**(4), 1149–1165.
- Girko, V. 1995. A Matrix Equation for Resolvents of Random Matrices with Independent Blocks. *Theory of Probability & Its Applications*, **40**(4), 635–644.
- Goemans, M.X., & Williamson, D.P. 1995. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, **42**(6), 1115–1145.
- Gorman, J.D., & Hero, A.O. 1990. Lower bounds for parametric estimation with constraints. *Information Theory, IEEE Transactions on*, **36**(6), 1285–1301.
- Grubišić, I., & Pietersz, R. 2007. Efficient rank reduction of correlation matrices. *Linear algebra and its applications*, **422**(2), 629–653.
- Hager, W.W., & Zhang, H. 2006. A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, **2**(1), 35–58.
- Hartley, R., Aftab, K., & Trumpf, J. 2011. L1 rotation averaging using the Weiszfeld algorithm. *Pages 3041–3048 of: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE.
- Hartley, R., Trumpf, J., Dai, Y., & Li, H. 2013. Rotation Averaging. *International Journal of Computer Vision*, **103**(3), 267–305.
- Ho, N.-D., & Van Dooren, P. 2005. On the pseudo-inverse of the Laplacian of a bipartite graph. *Applied Mathematics Letters*, **18**(8), 917–922.
- Hoff, P.D. 2009. Simulation of the Matrix Bingham–von Mises–Fisher distribution, with applications to multivariate and relational data. *Journal of Computational and Graphical Statistics*, **18**(2), 438–456.

- Hoory, S., Linial, N., & Wigderson, A. 2006. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, **43**(4), 439–562.
- Howard, S.D., Cochran, D., Moran, W., & Cohen, F.R. 2010. Estimation and Registration on Graphs. *Arxiv preprint arXiv:1010.2983*.
- Huang, Q.X., & Guibas, L. 2013. Consistent shape maps via semidefinite programming. *Pages 177–186 of: Computer Graphics Forum*, vol. 32. Wiley Online Library.
- Jamakovic, A., & Uhlig, S. 2007. On the relationship between the algebraic connectivity and graph's robustness to node and link failures. *Pages 96–102 of: Next Generation Internet Networks, 3rd EuroNGI Conference on*. IEEE.
- Journée, M., Nesterov, Y., Richtárik, P., & Sepulchre, R. 2010a. Generalized Power Method for Sparse Principal Component Analysis. *The Journal of Machine Learning Research*, **11**, 517–553.
- Journée, M., Bach, F., Absil, P.-A., & Sepulchre, R. 2010b. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM Journal on Optimization*, **20**(5), 2327–2351.
- Karcher, H. 1977. Riemannian center of mass and mollifier smoothing. *Communications on pure and applied mathematics*, **30**(5), 509–541.
- Keshavan, R.H., & Montanari, A. 2010. Regularization for matrix completion. *Pages 1503–1507 of: Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. IEEE.
- Keshavan, R.H., & Oh, S. 2009. OptSpace: A gradient descent algorithm on the Grassman manifold for matrix completion. *Arxiv preprint arXiv:0910.5260 v2*.
- Keshavan, R.H., Montanari, A., & Oh, S. 2009. Low-rank matrix completion with noisy observations: a quantitative comparison. *Pages 1216–1222 of: Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE.
- Keshavan, R.H., Montanari, A., & Oh, S. 2010. Matrix completion from noisy entries. *The Journal of Machine Learning Research*, **99**, 2057–2078.
- Koren, Y. 2009. *The BellKor solution to the Netflix grand prize*.
- Kressner, D., Steinlechner, M., & Vandereycken, B. 2013. *Low-rank tensor completion by Riemannian optimization*. Tech. rept. École polytechnique fédérale de Lausanne.

- Krishnan, S., Lee, P.Y., Moore, J.B., & Venkatasubramanian, S. 2007. Optimisation-on-a-manifold for global registration of multiple 3D point sets. *International Journal of Intelligent Systems Technologies and Applications*, **3**(3), 319–340.
- Lee, J.M. 1997. *Riemannian manifolds: An introduction to curvature*. Graduate Texts in Mathematics, vol. 176. Springer.
- Lee, K., & Bresler, Y. 2010. ADMiRA: Atomic decomposition for minimum rank approximation. *Information Theory, IEEE Transactions on*, **56**(9), 4402–4416.
- Leichtweiss, K. 1961. Zur Riemannschen Geometrie in Grassmannschen Mannigfaltigkeiten. *Math. Z.*, **76**, 334–366.
- Lin, L., Lu, J., Ying, L., Car, R., et al. 2009. Fast algorithm for extracting the diagonal of the inverse matrix with application to the electronic structure analysis of metallic systems. *Communications in Mathematical Sciences*, **7**(3), 755–777.
- Luo, Z., Ma, W., So, AMC, Ye, Y., & Zhang, S. 2010. Semidefinite relaxation of quadratic optimization problems. *Signal Processing Magazine, IEEE*, **27**(3), 20–34.
- Mackey, L., Talwalkar, A., & Jordan, M.I. 2011. Divide-and-conquer matrix factorization. *arXiv preprint arXiv:1107.0789*.
- Mardia, K.V., & Jupp, P.E. 2000. *Directional statistics*. John Wiley & Sons Inc.
- Markley, F Landis. 1988. Attitude determination using vector observations and the singular value decomposition. *The Journal of the Astronautical Sciences*, **36**(3), 245–258.
- Meyer, G., Bonnabel, S., & Sepulchre, R. 2011a. Linear regression under fixed-rank constraints: a Riemannian approach. In: *28th International Conference on Machine Learning*. ICML.
- Meyer, G., Bonnabel, S., & Sepulchre, R. 2011b. Regression on fixed-rank positive semidefinite matrices: a Riemannian approach. *The Journal of Machine Learning Research*, **12**, 593–625.
- Mezzadri, F. 2007. How to generate random matrices from the classical compact groups. *Notices of the AMS*, **54**(5), 592–604.

- Mishra, B., Meyer, G., & Sepulchre, R. 2011a. Low-rank optimization for distance matrix completion. *Pages 4455–4460 of: Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on.* IEEE.
- Mishra, B., Meyer, G., Bach, F., & Sepulchre, R. 2011b. Low-rank optimization with trace norm penalty. *Arxiv preprint arXiv:1112.2318.*
- Mishra, B., Meyer, G., Bonnabel, S., & Sepulchre, R. 2012a. Fixed-rank matrix factorizations and Riemannian low-rank optimization. *Arxiv preprint arXiv:1209.0430.*
- Mishra, B., Adithya Apuroop, K., & Sepulchre, R. 2012b. A Riemannian geometry for low-rank matrix completion. *Arxiv preprint arXiv:1211.1550.*
- Nesterov, Y. 2004. *Introductory lectures on convex optimization: A basic course.* Applied optimization, vol. 87. Springer.
- Ngo, T., & Saad, Y. 2012. Scaled gradients on Grassmann manifolds for matrix completion. *Pages 1421–1429 of: Advances in Neural Information Processing Systems 25.*
- Nocedal, J., & Wright, S.J. 1999. *Numerical optimization.* Springer Verlag.
- O’Neill, B. 1983. *Semi-Riemannian geometry: with applications to relativity.* Vol. 103. Academic Pr.
- Petersen, K.B., & Pedersen, M.S. 2006. *The matrix cookbook.*
- Rao, C.R. 1945. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society,* **37**(3), 81–91.
- Recht, B., Ré, C., & Wright, S.J. 2011. Parallel stochastic gradient algorithms for large-scale matrix completion. *Optimization Online.*
- Ring, W., & Wirth, B. 2012. Optimization methods on Riemannian manifolds and their application to shape space. *SIAM Journal on Optimization,* **22**(2), 596–627.
- Rusinkiewicz, S., & Levoy, M. 2001. Efficient variants of the ICP algorithm. *Pages 145–152 of: 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on.* IEEE.
- Russell, W.J., Klein, D.J., & Hespanha, J.P. 2011. Optimal estimation on the graph cycle space. *Signal Processing, IEEE Transactions on,* **59**(6), 2834–2846.

- Saerens, M., Fouss, F., Yen, L., & Dupont, P. 2004. The principal components analysis of a graph, and its relationships to spectral clustering. *Machine Learning: ECML 2004*, 371–383.
- Sarlette, A., & Sepulchre, R. 2009. Consensus optimization on manifolds. *SIAM J. Control and Optimization*, **48**(1), 56–76.
- Sato, H., & Iwai, T. 2013. A new, globally convergent Riemannian conjugate gradient method. *Optimization*, 1–21.
- Shalit, U., Weinshall, D., & Chechik, G. 2012. Online learning in the embedded manifold of low-rank matrices. *The Journal of Machine Learning Research*, **13**, 429–458.
- Singer, A. 2011. Angular synchronization by eigenvectors and semidefinite programming. *Applied and Computational Harmonic Analysis*, **30**(1), 20–36.
- Singer, A., & Shkolnisky, Y. 2011. Three-Dimensional Structure Determination from Common Lines in Cryo-EM by Eigenvectors and Semidefinite Programming. *SIAM Journal on Imaging Sciences*, **4**(2), 543–572.
- Smith, S.T. 2005. Covariance, subspace, and intrinsic Cramér-Rao bounds. *Signal Processing, IEEE Transactions on*, **53**(5), 1610–1630.
- Sonday, B., Singer, A., & Kevrekidis, I.G. 2013. Noisy dynamic simulations in the presence of symmetry: Data alignment and model reduction. *Computers and Mathematics with Applications*, **65**(10), 1535–1557.
- Stoica, P., & Marzetta, T.L. 2001. Parameter estimation problems with singular information matrices. *Signal Processing, IEEE Transactions on*, **49**(1), 87–90.
- Stoica, P., & Ng, B.C. 1998. On the Cramér-Rao bound under parametric constraints. *Signal Processing Letters, IEEE*, **5**(7), 177–179.
- Tang, G., & Nehorai, A. 2011a. Constrained Cramér–Rao Bound on Robust Principal Component Analysis. *Signal Processing, IEEE Transactions on*, **59**(10), 5070–5076.
- Tang, G., & Nehorai, A. 2011b. Lower bounds on the mean-squared error of low-rank matrix reconstruction. *Signal Processing, IEEE Transactions on*, **59**(10), 4559–4571.
- Tao, M., & Yuan, X. 2011. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, **21**(1), 57–81.

- Theis, F.J., Cason, T.P., & Absil, P.-A. 2009. Soft dimension reduction for ICA by joint diagonalization on the Stiefel manifold. *Pages 354–361 of: Independent Component Analysis and Signal Separation*. Springer.
- Toh, K.C., & Yun, S. 2010. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, **6**(15), 615–640.
- Trefethen, L.N., & Bau, D. 1997. *Numerical linear algebra*. Society for Industrial Mathematics.
- Tron, R., & Vidal, R. 2009. Distributed image-based 3D localization of camera sensor networks. *Pages 901–908 of: Decision and Control, held jointly with the 28th Chinese Control Conference. Proceedings of the 48th IEEE Conference on*. IEEE.
- Tzveneva, T., Singer, A., & Rusinkiewicz, S. 2011. *Global Alignment of Multiple 3-D Scans Using Eigenvector Synchronization (Bachelor thesis)*. Tech. rept. Princeton University.
- Vandereycken, B. 2013. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, **23**(2), 1214–1236.
- Waldspurger, I., d'Aspremont, A., & Mallat, S. 2012. Phase recovery, maxcut and complex semidefinite programming. *Arxiv preprint arXiv:1206.0102*.
- Wang, L., & Singer, A. 2013. *Exact and stable recovery of rotations for robust synchronization*. to appear in *Information and Inference: A Journal of the IMA*.
- Wang, L., Singer, A., & Wen, Z. 2013. Orientation Determination of Cryo-EM Images Using Least Unsquared Deviations. *SIAM Journal on Imaging Sciences*, **6**(4), 2450–2483.
- Wen, Z., Goldfarb, D., & Yin, W. 2010. Alternating direction augmented Lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, **2**(3-4), 203–230.
- 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.
- Wolfram. 2001. *Modified Bessel function of the first kind: Integral representations*, <http://functions.wolfram.com/03.02.07.0007.01>.

- Xavier, J., & Barroso, V. 2004. The Riemannian geometry of certain parameter estimation problems with singular Fisher information matrices. *Pages 1021–1024 of: Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 2. IEEE.
- Xavier, J., & Barroso, V. 2005. Intrinsic Variance Lower Bound (IVLB): an extension of the Cramér-Rao bound to Riemannian manifolds. *Pages 1033–1036 of: Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, vol. 5. IEEE.
- Yu, S.X. 2009. Angular embedding: From jarring intensity differences to perceived luminance. *Pages 2302–2309 of: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*.
- Yu, S.X. 2012. Angular Embedding: A Robust Quadratic Criterion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **34**(1), 158–173.