A discrete regression method on manifolds and its application to data on $\mathrm{SO}(n)$ ⋆

Nicolas Boumal * **P.-A. Absil**

* *Center for Systems Engineering and Applied Mechanics (CESAME) and ICTEAM Institute, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium. (Tel: (+32) 10 47 80 10; e-mail: nicolasboumal@gmail.com).*

Abstract: The regression problem of fitting a "smooth", discrete curve to data points on a Riemannian manifold is formulated here as an unconstrained, finite-dimensional optimization problem. Smoothness of a discrete curve, seen as a sequence of close points on the manifold, is assessed and encouraged by a regularity term in the objective function. This term is built upon a generalization of finite differences to manifolds introduced in this work. Tuning of the balance between fitting and regularity (or energy-efficiency) is achieved by adjusting two parameters. The proposed framework is described in detail and is then applied to the special orthogonal group $\mathrm{SO}(n)$, i.e., the set of rotations in $\mathbb{R}^n$. A Riemannian version of the nonlinear conjugate gradient method is used to minimize the resulting objective. To this end, an explicit formula for the derivative of the matrix logarithm is derived, yielding explicit formulas for the gradient of the objective. Numerical results are presented and show that smooth curves in $\mathrm{SO}(n)$ can be obtained in a few hundred iterations with the proposed algorithm.

Keywords: differential geometric methods; non-parametric regression; discrete time; finite differences; conjugate gradient methods; interpolation algorithms; least-squares approximation; rotation.

1. INTRODUCTION

Fitting a regression curve to data points can serve at least two purposes: it may help reduce measurement noise and it can fill gaps in the data. This paper presents a framework and an algorithm to define and solve a type of discrete regression problem across time-labeled data points on manifolds. Unlike regression in Euclidean spaces, only little work has been done in this area.

Let $p_1, p_2, \ldots, p_N$ be N data points in the Euclidean space $\mathbb{R}^n$ with time labels $t_1 \leq t_2 \leq \cdots \leq t_N$. Continuous regression consists in searching for a curve $\gamma : [t_1, t_N] \to \mathbb{R}^n$ that simultaneously (i) reasonably fits the data and (ii) is sufficiently "smooth". One common way of formalizing this loose description of a natural need is to express γ as the minimizer of an objective function such as

$$E_{\mathrm{c}}(\gamma) = \frac{1}{2}\sum_{i=1}^{N} \|p_i - \gamma(t_i)\|^2 + \frac{\lambda}{2}\int_{t_1}^{t_N} \|\dot{\gamma}(t)\|^2 \mathrm{d}t + \frac{\mu}{2}\int_{t_1}^{t_N} \|\ddot{\gamma}(t)\|^2 \mathrm{d}t, \quad (1)$$

defined over some suitable curve space Γ_c. When speed and acceleration have physical importance, in terms of energy for example, this formulation is especially adequate. The tuning parameters λ and μ enable the user to tune the balance between the conflicting goals of fitting and smoothness. It is well known that (i) when $\lambda > 0$, $\mu = 0$, the optimal γ is piecewise affine—see [Machado et al., 2006, Th. 3.2]—and (ii) when $\lambda = 0$, $\mu > 0$, the optimal γ is an approximating cubic spline (provided Γ_c contains these curves)—see [Machado and Silva Leite, 2006, Prop. 4.5, 4.6].

In this paper, the N data points p_i are allowed to lie on a Riemannian manifold $\mathcal{M}$ and the objective is discretized. Doing so, one avoids having to formulate and solve a variational problem on manifolds. A general framework is developed and an example is given for the case where $\mathcal{M}$ is the special orthogonal group $\mathrm{SO}(n)$, i.e., the set of n-by-n orthogonal matrices of determinant $+1$. $\mathrm{SO}(n)$ is in one-to-one correspondence with rotations in $\mathbb{R}^n$, hence its importance.

A number of authors have studied the problem of fitting smooth curves to data on manifolds. A central algorithm for spline construction in Euclidean spaces is the de Casteljau algorithm. The latter was modified then generalized to manifolds by Jakubiak et al. [2006]. Related papers referenced in that work are mainly focusing on continuous interpolation problems. A few of these are specific to SO(3). Jakubiak et al. emphasize the industrial importance of interpolation on SO(3) and hint that the first author focusing on it may be Shoemake [1985]. Shoemake used quaternions, hence the algorithm is very specific to the manifold at hand.

⋆ This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization), funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office. The scientific responsibility rests with its authors. The first author is a F.R.S.-FNRS Research Fellow.

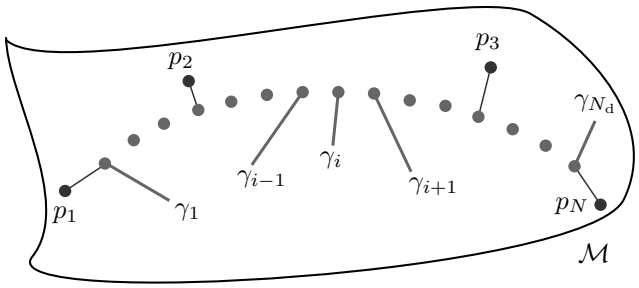

Fig. 1. Our algorithms fit a "smooth", discrete curve $(\gamma_1, \ldots, \gamma_{N_\mathrm{d}})$ to N data points p_i on a manifold $\mathcal{M}$.

Machado et al. [2006] show how (1) can be extended to manifolds for the special, simpler case $\mu = 0$. They show that, for suitable manifolds, the solutions are broken geodesics approximating the data. Camarinha et al. [1995] elaborate a definition of splines of class $\mathcal{C}^k$ on manifolds. Samir et al. [2011] present a steepest descent algorithm to minimize the extended version of (1) on manifolds and demonstrate it on the sphere. The optimal curves they obtain are cubic splines in the sense of Camarinha et al.

Most efforts have been focused on building continuous, interpolating splines for data on manifolds. Some of the proposed methods are constructive by nature and it may be unclear whether they are optimal in some sense or not. Furthermore, it may be excessively costly to solve variational problems on manifolds when, often times, only a few points of the computed curve will be useful such as, e.g., in MPC-like control or when resampling the data. In practical situations, the data will often be affected by noise, in which case it may be advantageous to soften the interpolation constraint in exchange for a smoother curve. For these reasons, we propose an algorithm for discrete regression on manifolds.

In this work, the curve space is reduced to the set of sequences of N_d points on the manifold, $\Gamma = \mathcal{M} \times \ldots \times \mathcal{M}$ (N_d copies of $\mathcal{M}$). For a discrete curve $\gamma = (\gamma_1, \ldots, \gamma_{N_\mathrm{d}})$ in Γ, each γ_i is associated to a fixed time τ_i such that $t_1 = \tau_1 < \tau_2 < \cdots < \tau_{N_\mathrm{d}} = t_N$. The problem is pictured in Figure 1. In the case $\mathcal{M} = \mathbb{R}^n$, the objective function we propose reduces to the following discretization of (1):

$$E(\gamma) = \frac{1}{2}\sum_{i=1}^{N} \|p_i - \gamma_{s_i}\|^2 + \frac{\lambda}{2}\sum_{i=1}^{N_\mathrm{d}} \alpha_i \|v_i\|^2 + \frac{\mu}{2}\sum_{i=1}^{N_\mathrm{d}} \beta_i \|a_i\|^2. \quad (2)$$

Here, the indices s_i are chosen such that τ_{s_i} is closest (ideally equal) to t_i. The weights α_i and β_i are chosen based on a suitable numerical integration scheme such as, e.g., the trapezium method, such that the sums effectively discretize the integrals present in (1). The vectors v_i and a_i, intuitively rooted at γ_i, approximate velocity and acceleration along the discrete curve γ at γ_i. Still considering $\mathcal{M} = \mathbb{R}^n$, the vectors v_i and a_i can be defined using finite differences. A bit of algebra shows that minimizing E comes down to solving a band linear system.

For $\mathcal{M}$ a nonlinear manifold, classical finite differences no longer make sense. Finite differences are generalized to manifolds in this work, which permits the introduction of a generalization of (2). The details are covered in Section 3.

The main advantage of this approach is that the problem reduces to a finite dimensional optimization problem on a manifold. Efficient algorithms of first and second order for these problems are covered by Absil et al. [2008], with an emphasis on matrix manifolds. A geometric, nonlinear conjugate gradient method is shown in Section 4. It has the advantage of needing only first order derivatives, like the steepest descent method, while often being superior to it according to our experiments.

Finally, the proposed framework is specialized to $\mathrm{SO}(n)$ in Section 5.

2. GEOMETRIC TOOLS

A reminder of essential elements of Riemannian geometry—mainly based on Boothby [1986] and Absil et al. [2008]—is given here. The emphasis is on giving the intuition rather than being technically precise.

Intuitively, manifolds are sets that can be locally identified with patches of $\mathbb{R}^n$. A simple example of a smooth manifold is the sphere, $\mathbb{S}^2$. The sphere is easy to picture and shares fundamental properties with $\mathrm{SO}(n)$: they are both compact Riemannian submanifolds of a Euclidean space, respectively $\mathbb{R}^3$ and $\mathbb{R}^{n\times n}$. The framework developed to state and solve regression problems on manifolds is well defined for all smooth Riemannian manifolds, including quotient spaces. Nevertheless, focusing on submanifolds of Euclidean spaces greatly simplifies definitions.

Let $\mathcal{M}$ be a smooth manifold. A smooth curve on $\mathcal{M}$ is a smooth function $c : I \to \mathcal{M}$, where I is an interval of $\mathbb{R}$. At each point $x \in \mathcal{M}$, one can define a tangent vector space noted $T_x\mathcal{M}$.

Definition 1. (tangent space). 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 vector subspace of $\mathbb{R}^n$ defined by:

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

In this definition, $c'(t)$ stands for the usual derivative of c.

Since tangent spaces are vector spaces, they can be endowed with inner products. In loose terms, when the mapping that maps points on $\mathcal{M}$ to inner products is continuous in some sense, $\mathcal{M}$ is a Riemannian manifold. Since, for all $x \in \mathcal{M}$, $T_x\mathcal{M}$ is a vector subspace of the embedding space $\mathbb{R}^n$, a natural inner product on $T_x\mathcal{M}$, noted $\langle \cdot, \cdot \rangle_x$, is obtained by restricting the inner product of $\mathbb{R}^n$ to $T_x\mathcal{M}$. If this is done for all the tangent spaces, $\mathcal{M}$ is a Riemannian submanifold of $\mathbb{R}^n$. The associated norm is $\|v\|_x = \sqrt{\langle v, v \rangle_x}$. When it is clear from the context, the subscript x is often omitted. $\mathrm{SO}(n)$ is a submanifold of $\mathbb{R}^{n\times n}$ endowed with the usual matrix inner product $\langle A, B \rangle = \mathrm{trace}\left(A^\top B\right)$.

The objective function constructed in this paper is a real-valued function defined over a manifold, i.e., a scalar field. Attempting to minimize this function will raise the need for an adequate concept of gradient.

Definition 2. (gradient). Let f be a scalar field on a smooth, finite-dimensional Riemannian manifold $\mathcal{M}$. The

gradient of f at x, denoted by $\operatorname{grad} f(x)$, is defined as the unique element of $T_x\mathcal{M}$ satisfying:

$$\mathrm{D}f(x)[v] = \langle \operatorname{grad} f(x), v \rangle_x , \quad \forall v \in T_x\mathcal{M},$$

where $\mathrm{D}f(x)[v]$ stands for the directional derivative of f at x along v.

For a scalar field f on a Euclidean space, $\operatorname{grad} f$ is the usual gradient.

The availability of inner products yields a natural definition of distance on manifolds. Given $x, y \in \mathcal{M}$, consider a smooth curve $c : [0, 1] \to \mathcal{M}$ such that $c(0) = x$ and $c(1) = y$. The length of the path c between x and y is

$$L(c) = \int_0^1 \|c'(t)\|_{c(t)} \mathrm{d}t.$$

The Riemannian (or geodesic) distance between x and y, written $\operatorname{dist}(x, y)$, is the length of a shortest such path. According to the Hopf-Rinow theorem, such a path is guaranteed to exist when $\mathcal{M}$ is complete, such as $\mathrm{SO}(n)$ for example.

In a Euclidean space, the shortest path is a straight line. When parameterized by arclength, such paths have zero acceleration. This leads to the definition of geodesic curves on manifolds, which generalize straight lines.

Definition 3. (geodesic curve). A curve $c : \mathbb{R} \to \mathcal{M}$ is a *geodesic curve* if it has zero acceleration, i.e., if $c''(t) \perp T_{c(t)}\mathcal{M}, \forall t$.

Geodesics are shortest paths between close points.

Given a point $x \in \mathbb{R}^n$ and a vector $v \in \mathbb{R}^n$, $x + v$ is the point in $\mathbb{R}^n$ one reaches by traveling from x in the direction v for a distance $\|v\|$ along a straight path. The exponential map generalizes this notion to manifolds.

Definition 4. (exponential map). Let $\mathcal{M}$ be a Riemannian manifold and $x \in \mathcal{M}$. For every $v \in T_x\mathcal{M}$, there exists an open interval $I \ni 0$ and a unique geodesic $\gamma(\cdot; x, v) : I \to \mathcal{M}$ such that $\gamma(0; x, v) = x$ and $\dot{\gamma}(0; x, v) = v$. Moreover, $\gamma(t; x, av) = \gamma(at; x, v)$ (homogeneity property). The mapping

$$\operatorname{Exp}_x : T_x\mathcal{M} \to \mathcal{M} : v \mapsto \operatorname{Exp}_x(v) = \gamma(1; x, v)$$

is called the *exponential map at x*.

Of course, when $\mathcal{M}$ is a Euclidean space, $\operatorname{Exp}_x(v) = x + v$.

The inverse mapping, called the logarithmic map and defined hereafter, will be crucial in generalizing finite differences. Given a root point x and a target point y, the logarithmic map returns a tangent vector at x, pointing toward y, of length $\operatorname{dist}(x, y)$.

Definition 5. (logarithmic map). Let $\mathcal{M}$ be a Riemannian manifold. The *logarithmic map* at $x \in \mathcal{M}$ is

$$\operatorname{Log}_x : \mathcal{M} \to T_x\mathcal{M} : y \mapsto \operatorname{Log}_x(y) = v,$$
$$\text{such that } \operatorname{Exp}_x(v) = y \text{ and } \|v\|_x = \operatorname{dist}(x, y).$$

When $\mathcal{M}$ is a Euclidean space, $\operatorname{Log}_x(y) = y - x$. As is, Definition 5 is not perfect. There might indeed be more than one eligible v, as discussed by Samir et al. [2011]. As long as x and y are not "too far apart", the above definition is satisfactory.

In Euclidean spaces, it is natural to compare vectors rooted at different points in space, so much so 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 combined immediately. One needs a mathematical tool capable of *transporting* vectors between tangent spaces while retaining the information they contain. This motivates the definition of vector transports T such as introduced in [Absil et al., 2008 §8.1]. In particular, for $x, y \in \mathcal{M}$ and $u = \operatorname{Log}_x(y)$, $\mathsf{T}_u : T_x\mathcal{M} \to T_y\mathcal{M}$ is a linear map and T_0 is the identity.

3. GENERALIZED OBJECTIVE

In order to generalize (2) to manifolds, two steps need to be taken. Both originate from the loss of a vector space structure. First, it is natural to replace the Euclidean distance $\|p_i - \gamma_{s_i}\|$ with the Riemannian distance $\operatorname{dist}(p_i, \gamma_{s_i})$. Then, in the next subsection, we propose geometric formulas for the velocity and acceleration vectors v_i and a_i.

3.1 Geometric finite differences

For ease of notation, assume the discretization times τ_i are homogeneously spaced, with spacing $\Delta\tau$. It is not challenging to get rid of this assumption. When $\mathcal{M}$ is a Euclidean space, a first order, forward finite difference is a reasonable formula for v_i:

$$v_i = \frac{\gamma_{i+1} - \gamma_i}{\Delta\tau}.$$

However, the difference appearing at the numerator does not, in general, make sense if $\mathcal{M}$ lacks a vector space structure. It can nevertheless be interpreted: $\gamma_{i+1} - \gamma_i$ is a vector rooted at γ_i and pointing toward γ_{i+1}. Furthermore, the length of the vector is the distance between γ_i and γ_{i+1}. Using the logarithmic map associated to the manifold $\mathcal{M}$, consider the following, more general, formula for v_i:

$$v_i = \frac{\operatorname{Log}_{\gamma_i}(\gamma_{i+1})}{\Delta\tau}. \tag{3}$$

v_i is now a vector in the tangent space $T_{\gamma_i}\mathcal{M}$ pointing toward γ_{i+1} and such that $\|v_i\| = \operatorname{dist}(\gamma_i, \gamma_{i+1}) / \Delta\tau$. When $\mathcal{M}$ is a Euclidean space, (3) reduces to the original finite difference formula.

The same trick can be used for second order derivatives, by exhibiting differences between close points. The classic formula

$$a_i = \frac{\gamma_{i+1} - 2\gamma_i + \gamma_{i-1}}{\Delta\tau^2} = \frac{(\gamma_{i+1} - \gamma_i) + (\gamma_{i-1} - \gamma_i)}{\Delta\tau^2}$$

becomes

$$a_i = \frac{\operatorname{Log}_{\gamma_i}(\gamma_{i+1}) + \operatorname{Log}_{\gamma_i}(\gamma_{i-1})}{\Delta\tau^2}.$$

This formula exhibits several desirable properties. In particular, a_i is zero if and only if there exists a geodesic $\gamma(t)$ such that $\gamma(\tau_j) = \gamma_j$, $j = i - 1, i, i + 1$.

It is easy to construct similar formulas for unevenly spaced discretization times, unilateral differences (for the end points) and higher order derivatives.

3.2 Objective function

The complete objective function we propose for the discrete regression problem on manifolds is $E : \Gamma = \mathcal{M}^{N_\mathrm{d}} \to \mathbb{R}$, with

$$E(\gamma) = \frac{1}{2}\sum_{i=1}^{N} \operatorname{dist}^2(p_i, \gamma_{s_i}) + \frac{\lambda}{2}\sum_{i=1}^{N_\mathrm{d}-1} \alpha_i \left\| \frac{\operatorname{Log}_{\gamma_i}(\gamma_{i+1})}{\Delta\tau} \right\|_{\gamma_i}^2 + \frac{\mu}{2}\sum_{i=2}^{N_\mathrm{d}-1} \beta_i \left\| \frac{\operatorname{Log}_{\gamma_i}(\gamma_{i+1}) + \operatorname{Log}_{\gamma_i}(\gamma_{i-1})}{\Delta\tau^2} \right\|_{\gamma_i}^2 . \quad (4)$$

For ease of notation, we have set $\alpha_{N_\mathrm{d}}, \beta_1$ and β_{N_d} to 0. The simplest choice for the other integration weights is $\alpha_i = \Delta\tau$, $i = 1 \dots N_\mathrm{d} - 1$ and $\beta_i = \Delta\tau$, $i = 2 \dots N_\mathrm{d} - 1$.

4. GEOMETRIC CONJUGATE GRADIENT

Solving the regression problem on $\mathcal{M}$ comes down to minimizing E (4) over Γ, a manifold of dimension $N_\mathrm{d} \dim(\mathcal{M})$. Constraining the points γ_i to lie on $\mathcal{M}$ can be difficult with standard optimization software, since for some manifolds the constraints may take the form of nonlinear equalities. It is therefore advisable to use optimization algorithms that exploit the rich structure of the constraints. For completeness, a geometric version of the nonlinear conjugate gradient method adapted from Absil et al. [2008] is given in Algorithm 1. Under certain conditions, notably on the step size selection algorithm, the output sequence will converge toward a critical point. For more information, the reader is referred to Absil et al. [2008].

Algorithm 1. (Geometric conjugate gradient method).
Input: A scalar field $f : \mathcal{N} \to \mathbb{R}$, its gradient $\operatorname{grad} f$ and an initial guess $x_0 \in \mathcal{N}$, where $\mathcal{N}$ is a Riemannian manifold equipped with a vector transport T.
Output: A sequence $x_1, x_2, \dots$ in $\mathcal{N}$.

$p_0 := -\operatorname{grad} f(x_0)$
$k := 0$
while $\operatorname{grad} f(x_k) \neq 0$ **do**
  $\alpha_k :=$ choose step size (classical line-search)
  $x_{k+1} := \operatorname{Exp}_{x_k}(\alpha_k p_k)$
  $\beta_{k+1} := \|\operatorname{grad} f(x_{k+1})\|^2 / \|\operatorname{grad} f(x_k)\|^2$
  $p_{k+1} := -\operatorname{grad} f(x_{k+1}) + \beta_{k+1} \mathsf{T}_{\alpha_k p_k}(p_k)$
  $k := k + 1$
end while

Notice that setting β_k to zero yields a steepest descent-like method. For our purpose, Algorithm 1 is applied to the objective function $f = E$ over the manifold $\mathcal{N} = \Gamma$.

5. APPLICATION: DENOISING AND RESAMPLING OF DATA ON SO(n)

The orientation of a rigid body in $\mathbb{R}^3$ is characterized by an axis and an angle of rotation. The associated rotation can be represented by a 3-by-3 orthogonal matrix of unit determinant. The set of all these matrices is called the special orthogonal group of dimension 3, SO(3). It is also a Riemannian manifold. In this section, denoising and resampling of data on SO(n) is demonstrated using the framework proposed in this paper.

Set:	$\operatorname{SO}(n) = \{A \in \mathbb{R}^{n\times n} \ : \ A^\top A = I, \det(A) = 1\}$
Tangent spaces:	$T_A \operatorname{SO}(n) = \{H \in \mathbb{R}^{n\times n} \ : \ A^\top H + H^\top A = 0\}$
Inner product:	$\langle H_1, H_2 \rangle_A = \operatorname{trace}\left(H_1^\top H_2\right)$
Vector norm:	$\|H\|_A = \sqrt{\langle H, H \rangle_A}$
Distance:	$\operatorname{dist}(A, B) = \|\operatorname{Log}_A(B)\| = \|\log(A^\top B)\|$
Exponential:	$\operatorname{Exp}_A(H) = A \operatorname{Exp}_I\left(A^\top H\right) = A \exp(A^\top H)$
Logarithm:	$\operatorname{Log}_A(B) = A \log(A^\top B)$
Transport:	$\mathsf{T}_{H_1}(H_2) = B A^\top H_2$, with $H_1, H_2 \in T_A \operatorname{SO}(n)$ and $H_1 = \operatorname{Log}_A(B)$.

Table 1. Toolbox for SO(n).

5.1 Toolbox

All the important geometric functions on SO(n) have nice, closed-form expressions, as shown in Table 1—see, e.g., Boothby [1986] for some treatment of Lie groups. This toolbox is sufficient to compute the objective function (4). In order to apply Algorithm 1 to it, the gradient of (4) is needed too. This will require some work.

5.2 Objective

Specializing (4) to SO(n) yields:

$$E(\gamma) = \frac{1}{2}\sum_{i=1}^{N} \left\|\log(p_i^\top \gamma_{s_i})\right\|^2 + \frac{\lambda}{2}\sum_{i=1}^{N_\mathrm{d}-1} \alpha_i \left\| \frac{\log(\gamma_i^\top \gamma_{i+1})}{\Delta\tau} \right\|^2 + \frac{\mu}{2}\sum_{i=2}^{N_\mathrm{d}-1} \beta_i \left\| \frac{\log(\gamma_i^\top \gamma_{i+1}) + \log(\gamma_i^\top \gamma_{i-1})}{\Delta\tau^2} \right\|^2 , \quad (5)$$

where log is the matrix logarithm (8). Focusing on the last term, note that

$$\left\|\log(\gamma_i^\top \gamma_{i+1}) + \log(\gamma_i^\top \gamma_{i-1})\right\|^2 = \|\log(\gamma_i^\top \gamma_{i+1})\|^2 + \|\log(\gamma_i^\top \gamma_{i-1})\|^2 + 2\left\langle \log(\gamma_i^\top \gamma_{i+1}), \log(\gamma_i^\top \gamma_{i-1}) \right\rangle ,$$

where $\langle \cdot, \cdot \rangle$ stands for the usual real matrix inner product $\langle A, B \rangle = \operatorname{trace}\left(A^\top B\right)$. Two scalar functions f and g are central to the definition of E:

$$f(A, B) = \|\log(A^\top B)\|^2, \quad (6)$$

$$g(A, B, C) = \left\langle \log(A^\top B), \log(A^\top C) \right\rangle . \quad (7)$$

5.3 Gradient of the objective

The objective E (5) is a linear combination of the functions f and g evaluated at discretization points γ_i and one (or two) of their neighbors. Hence, in order to compute the gradient of E, it is sufficient to have formulas for the gradients of f and g in each of their variables.

A general result mentioned in [Samir et al., 2011, Theorem 3.1] states that

$$\operatorname{grad}(X \mapsto f(A, X))(B) = -2 \operatorname{Log}_B(A) .$$

Since $f(A, B) = f(B, A)$, it remains to derive formulas for the gradients of g. That is the purpose of this subsection and will require a significant amount of algebra.

The function g is expressed in terms of the geometric mapping Log. Log is expressed in terms of the matrix logarithm, itself defined as a series:

$$\log(A) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1}}{k}(A-I)^k. \tag{8}$$

The series is derived from the real series $\log(1+x) = x - x^2/2 + x^3/3 - \ldots$, which is convergent if $|x| < 1$ (and divergent if $|x| > 1$). One thus expects the matrix counterpart to be convergent when $\rho(A-I) < 1$, where $\rho(X)$ is the spectral radius of X.

This is indeed the case. On the other hand, $\log(x)$ is well defined for all $x > 0$. According to [Higham, 2008, problem 11.1], a Taylor series can be used to define the principal logarithm of any matrix having no eigenvalue on $\mathbb{R}^-$. A hint to this is that, using the identity

$$\log(A) = s\log(A^{1/s}),$$

it is always possible, with s large enough, to bring the eigenvalues of $A^{1/s}$ into a circle centered at 1 (in the complex plane) and with radius strictly smaller than 1. Consequently, the matrix $A^{1/s} - I$ has spectral radius less than 1 and the Taylor series (8) is convergent.

Working out an explicit formula for the gradients of g requires formulas for the directional derivatives of the matrix logarithm. A generic way of computing derivatives of functions of diagonalizable matrices is demonstrated next. For it to apply to logarithms, it is crucial that the matrix logarithm may be expressed as a convergent Taylor series, as hinted earlier. The way the material in this section is derived is inspired by [Fillard et al., 2007, appendix] and Bhatia [2007].

Let A be an n-by-n diagonalizable matrix, U an invertible matrix and $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$, such that $A = UDU^{-1}$. Let z be a smooth, real function defined by a Taylor series

$$z(x) = \sum_{k=0}^{\infty} a_k x^k.$$

Generalized to matrices, this yields

$$z(A) = \sum_{k=0}^{\infty} a_k A^k = U \operatorname{diag}(z(\lambda_1), \ldots, z(\lambda_n))\, U^{-1}.$$

One would like to compute $\mathrm{D}z(A)[H]$, the directional derivative of z at A in the direction H, by differentiating the series term by term. To this end, note that:

$$\begin{aligned}\mathrm{D}(X \mapsto X^k)(A)[H] &= \lim_{h\to 0} \frac{(A+hH)^k - A^k}{h} \\ &= \sum_{l=1}^{k} A^{l-1} H A^{k-l}.\end{aligned}$$

Then,

$$\begin{aligned}\mathrm{D}z(A)[H] &= \sum_{k=1}^{\infty} a_k \sum_{l=1}^{k} A^{l-1} H A^{k-l} \\ &= U\left[\sum_{k=1}^{\infty} a_k \sum_{l=1}^{k} D^{l-1} U^{-1} H U D^{k-l}\right] U^{-1} \\ &= U \cdot \mathrm{D}z(D)[U^{-1}HU] \cdot U^{-1},\end{aligned}$$

which is a simpler object to compute because D is diagonal. Define $\tilde{H} = U^{-1}HU$ and $M = \mathrm{D}z(D)[\tilde{H}]$. Looking at M entry-wise yields:

$$\begin{aligned}M_{ij} &= \sum_{k=1}^{\infty} a_k \sum_{l=1}^{k} (D^{l-1}\tilde{H}D^{k-l})_{ij} \\ &= \sum_{k=1}^{\infty} a_k \sum_{l=1}^{k} \lambda_i^{l-1}\lambda_j^{k-l}\tilde{H}_{ij} \\ &= \tilde{H}_{ij} \sum_{k=1}^{\infty} a_k \frac{\lambda_j^k}{\lambda_i} \sum_{l=1}^{k} \left(\frac{\lambda_i}{\lambda_j}\right)^l.\end{aligned}$$

Using the identity $\sum_{l=1}^{k} x^k = x\frac{1-x^k}{1-x}$, valid for $x \neq 1$, one can distinguish two cases:

$$\frac{\lambda_j^k}{\lambda_i}\sum_{l=1}^{k}\left(\frac{\lambda_i}{\lambda_j}\right)^l = \begin{cases}\frac{\lambda_i^k - \lambda_j^k}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j, \\ k\lambda_i^{k-1} & \text{if } \lambda_i = \lambda_j.\end{cases}$$

Hence:

$$M_{ij} = \tilde{H}_{ij}\tilde{z}(\lambda_i, \lambda_j),$$

with:

$$\tilde{z}(\lambda_i, \lambda_j) = \begin{cases}\frac{z(\lambda_i) - z(\lambda_j)}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j, \\ z'(\lambda_i) & \text{if } \lambda_i = \lambda_j.\end{cases}$$

The coefficients $\tilde{z}(\lambda_i, \lambda_j)$ are termed the *first divided differences* (fdd) in [Bhatia, 2007, p. 60]. Consider the matrix $\tilde{Z}$ such that $\tilde{Z}_{ij} = \tilde{z}(\lambda_i, \lambda_j)$. Then,

$$M = \tilde{H} \odot \tilde{Z},$$

where $\odot$ stands for entry-wise multiplication (Hadamard's product). Setting $z = \log$, Algorithm 2 computes the matrix $\mathrm{Dlog}\,(A)\,[H]$ and takes advantage of the fact that orthogonal matrices are diagonalized by unitary matrices. Hereafter, $\bar{A}$ denotes the conjugate of A and A^* is the conjugate transpose of A.

Algorithm 2. (directional derivative of log).
Input: $A \in \mathrm{SO}(n)$, $H \in \mathbb{R}^{n\times n}$.
Output: $\mathrm{Dlog}\,(A)\,[H]$.
Diagonalize A: $A = UDU^*$, $D = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$.
Compute $\tilde{H} = U^*HU$.
Compute $\tilde{Z}$ with $\tilde{Z}_{ij} = \begin{cases}\frac{\log(\lambda_i) - \log(\lambda_j)}{\lambda_i - \lambda_j} & \text{if } \lambda_i \neq \lambda_j, \\ \frac{1}{\lambda_i} & \text{if } \lambda_i = \lambda_j.\end{cases}$
return $\mathrm{Dlog}\,(A)\,[H] = U(\tilde{H} \odot \tilde{Z})U^*$.

We now state three useful properties without proof, due to lack of space. The first two refer to the complex inner product $\langle X, Y\rangle = \operatorname{trace}(Y^*X)$.

$$\langle P, QRS\rangle = \langle Q^*PS^*, R\rangle, \quad \forall\, P, Q, R, S \in \mathbb{C}^{n\times n}. \tag{9}$$

$$\langle P, Q \odot R\rangle = \left\langle P \odot \bar{R}, Q\right\rangle, \qquad \forall\, P, Q, R \in \mathbb{C}^{n\times n}. \tag{10}$$

$$A\mathrm{Dlog}\,(A)\,[H] = \mathrm{Dlog}\,(A)\,[AH] \in T_I\,\mathrm{SO}(n), \quad \forall\, A \in \mathrm{SO}(n), H \in T_I\,\mathrm{SO}(n). \tag{11}$$

Collecting Algorithm 2 and identities (9)–(11):

$$\begin{aligned}&\mathrm{D}\left(X \mapsto g(A, B, X)\right)(C)[H] \\ &= \left\langle \log(A^\top B), \mathrm{D}\left(X \mapsto \log(A^\top X)\right)(C)[H]\right\rangle \\ &= \left\langle \log(A^\top B), \mathrm{Dlog}\left(A^\top C\right)\left[A^\top H\right]\right\rangle \\ &\text{Diagonalize } A^\top C = UDU^*, \text{ set } \tilde{H} = U^*A^\top HU \\ &\text{and define } \tilde{Z}, \text{ fdd's of } \lambda(A^\top C) \text{ w.r.t. } \log:\end{aligned}$$

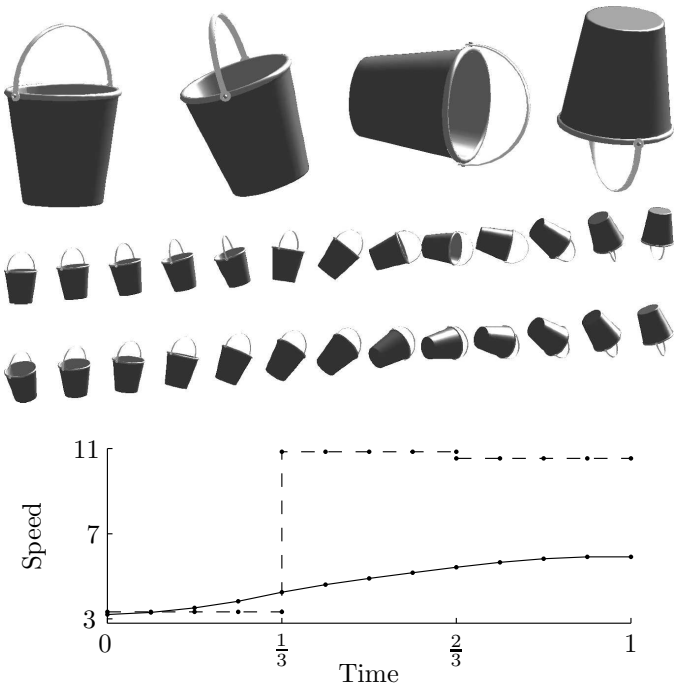

Fig. 2. Regression on SO(3). Line 1: data points (random orientations). Line 2: piecewise geodesic interpolation. Line 3: regression with $\lambda = 0$ and $\mu = 10^{-2}$. Plot: the dashed and continuous lines are the speed profiles of (resp.) Line 2 and Line 3, which is much smoother.

$$= \left\langle \log(A^\top B), U(\tilde{H} \odot \tilde{Z})U^* \right\rangle$$

Successively use (9), (10) and (9) :

$$= \left\langle U^* \log(A^\top B)U, \tilde{H} \odot \tilde{Z} \right\rangle$$

$$= \left\langle (U^* \log(A^\top B)U) \odot \tilde{\tilde{Z}}, U^* A^\top H U \right\rangle$$

$$= \left\langle AU \left(\left(U^* \log(A^\top B)U\right) \odot \tilde{\tilde{Z}} \right) U^*, H \right\rangle$$

Identify with Algorithm 2 and use (11) :

$$= \left\langle A\mathrm{Dlog}\left(C^\top A\right)\left[\log(A^\top B)\right], H \right\rangle$$

$$= \left\langle C\mathrm{Dlog}\left(C^\top A\right)\left[C^\top A \log(A^\top B)\right], H \right\rangle.$$

Identification with Definition 2 yields an explicit formula for the gradient of g in its third and, by symmetry, second argument. Similar developments yield a formula for the gradient of g in its first argument. Summing up:

$$\mathrm{grad}\left(X \mapsto g(X,B,C)\right)(A) = A\mathrm{Dlog}\left(A^\top B\right)\left[A^\top B \log(C^\top A)\right] + A\mathrm{Dlog}\left(A^\top C\right)\left[A^\top C \log(B^\top A)\right] \in T_A \mathrm{SO}(n),$$

$$\mathrm{grad}\left(X \mapsto g(A,X,C)\right)(B) = B\mathrm{Dlog}\left(B^\top A\right)\left[B^\top A \log(A^\top C)\right] \in T_B \mathrm{SO}(n),$$

$$\mathrm{grad}\left(X \mapsto g(A,B,X)\right)(C) = C\mathrm{Dlog}\left(C^\top A\right)\left[C^\top A \log(A^\top B)\right] \in T_C \mathrm{SO}(n).$$

Using (11), it can be checked that these vectors belong to the appropriate tangent spaces. One now has all the necessary tools to minimize E over Γ, given N data points $p_i \in \mathrm{SO}(n)$ and an initial guess $\gamma \in \Gamma$. The next subsection illustrates the results for $n = 3$.

5.4 Results

Figure 2 demonstrates the effect of regression through $N = 4$ random data points on SO(3) ($t_i = (i-1)/3,\ i = 1 \dots 4$) with $N_\mathrm{d} = 13$ discretization points ($\tau_i = (i-1)/12,\ i = 1 \dots 13$), in comparison with piecewise geodesic interpolation. The latter was fed to Algorithm 1 as initial guess to produce the former. Regression yields a significantly smoother speed profile. In practice, the algorithm operates by building a coarse γ (small N_d) first, then iteratively refining and reoptimizing the results. Algorithm 1 was run for 628 iterations in total. The norm of the gradient was reduced by about five orders of magnitude between the initial guess (line 2) and the final result (line 3).

6. CONCLUSIONS AND PERSPECTIVES

We proposed a framework and algorithm for discrete regression on manifolds and demonstrated it on SO(3). In other works, we have applied the same method on the sphere and on the set of positive-definite matrices. The major challenge now is to apply higher order optimization methods to the objective function, to improve speed of convergence. This will require the computation of second order derivatives, which might prove complicated since eigenvalue decompositions are involved in the gradient expression. In future work, we will explore approximations of the logarithmic map to circumvent this difficulty.

REFERENCES

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

R. Bhatia. *Positive definite matrices*. Princeton University Press, 2007.

W.M. Boothby. *An introduction to differentiable manifolds and Riemannian geometry*. Elsevier, 1986.

M. Camarinha, F. Silva Leite, and P. Crouch. Splines of class C^k on non-Euclidean spaces. *IMA J. Math. Control Inform.*, 12(4):399–410, 1995. ISSN 0265-0754.

P. Fillard, X. Pennec, V. Arsigny, and N. Ayache. Clinical DT-MRI estimation, smoothing, and fiber tracking with log-Euclidean metrics. *IEEE Transactions on Medical Imaging*, 26(11):1472–1482, 2007.

N. J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. ISBN 978-0-898716-46-7.

J. Jakubiak, F. Silva Leite, and R. C. Rodrigues. A two-step algorithm of smooth spline generation on Riemannian manifolds. *J. Comput. Appl. Math.*, 194(2):177–191, 2006. ISSN 0377-0427.

L. Machado and F. Silva Leite. Fitting smooth paths on Riemannian manifolds. *Int. J. Appl. Math. Stat.*, 4(J06): 25–53, 2006.

L. Machado, F. Silva Leite, and K. Hüper. Riemannian means as solutions of variational problems. *LMS J. of Comput. Math.*, 9:86–103, 2006. ISSN 1461-1570.

C. Samir, P.-A. Absil, A. Srivastava, and E. Klassen. A gradient-descent method for curve fitting on Riemannian manifolds, 2011. Accepted for publication in Foundations of Computational Mathematics.

K. Shoemake. Animating rotation with quaternion curves. *ACM SIGGRAPH*, 19(3):245–254, 1985.