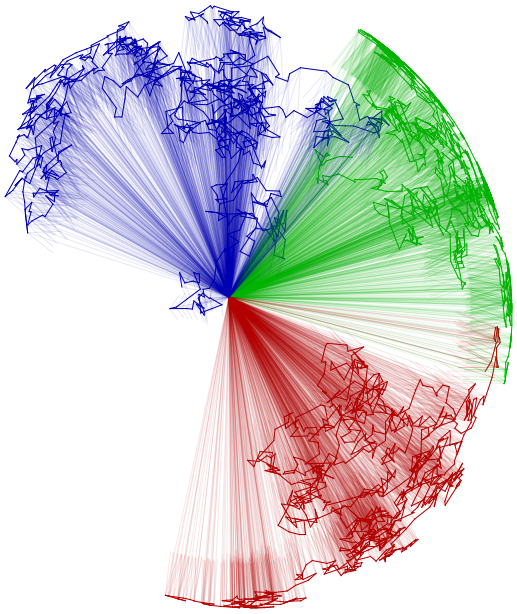
UNIVERSITY OF COPENHAGEN
FACULTY OF SCIENCE



PhD thesis

Line Kühnel

Stochastic Modelling on Manifolds

Department of Computer Science

Advisors: Stefan Sommer, Mads Nielsen

Handed in: December 30, 2018

University of Copenhagen
Faculty of Science
Department of Computer Science

Stochastic Modelling on Manifolds

Line Kühnel

Supervisor: Stefan Sommer
Second Supervisor: Mads Nielsen

This thesis has been submitted to the PhD School of The Faculty of Science,
University of Copenhagen, on December 30, 2018

Abstract

The constant improvement of data collection techniques increases the complexity of observed data objects. Cameras and scanning technologies make it possible to retrieve detailed images of everything from microscopic structures of a cell to 3D images of anatomical objects. No matter if data are curve outlines of a shape, medical images, or a collection of landmark points for an object, such complex data structures lack vector space properties and will hence challenge the well-known statistical theory for data in Euclidean space. All things considered, new generalised statistical methods have to be developed for analysing non-linear data samples. This thesis focus on two topics: The incorporation of uncertainty estimation in generalised statistical models, and the use of symbolic software packages for concise implementation of non-linear statistical methods.

Obtaining closed-form expressions for probability density functions of distributions on manifolds is difficult and has only been successfully deduced in simple cases. Alternatively, the presented work uses stochastic theory to describe uncertainty and variation in distributions on manifolds. A regression model is presented for estimating the relation between multiple Euclidean covariates and a manifold-valued response variable. The data spaces are connected by transportation of a Euclidean semi-martingale to the response manifold. The transported semi-martingale can model non-geodesic regression curves and include uncertainty in the estimated relation.

Furthermore, we present two different methods for describing the uncertainty in populations of image data. The first method is a mixed-effects model which separates uncertainty in images into a deformation effect and a spatial intensity effect. Deformations are modelled as displacement fields on a discretised lattice of the image domain. Based on a maximum likelihood procedure, parameters for the uncertainty effects are estimated along with a fixed template image. The second model represents uncertainty by stochastic deformation of images. The stochastic deformations are modelled as a stochastic flow of diffeomorphisms based on the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework. Estimation of parameters, which define noise correlations and local variability in images, is performed by matching data moments against moments of the stochastic deformation.

Numerical frameworks developed for Deep Learning tasks are generally computationally optimised and able to incorporate computations on GPUs, parallel computing and symbolic calculations. The second topic of the dissertation focuses on the use of such numerical frameworks for concise implementation of concepts from differential geometry and non-linear statistics.

Resumé

Den konstante udvikling af data indsamlingsteknikker øger kompleksiteten af observerede dataobjekter. Kamera og scannings teknologier gør det muligt at opnå detaljerede billeder af alt lige fra mikroskopiske strukturer af celler til 3D billeder af anatomiske objekter. Lige meget om data er kurver der repræsenterer konturen af en shape, medicinske billeder, eller punkter der markerer landemærker for objekter, kan sådanne komplekse datastrukturer mangle vektorrums egenskaber og derved udfordre eksisterende statistiske metoder udviklet til analyse af Euklidisk data. Alt i alt, skal der udvikles nye generaliserede statistiske metoder for at analysere ikke-lineære data objekter. Afhandlingen omhandler to emner: Inkorporering af usikkerhedsestimation i generaliserede statistiske modeller, og brugen af symbolske software pakker for koncis implementation af ikke-lineære statistiske metoder.

At opnå et lukket udtryk for sandsynlighedstæthedsfunktioner af fordelinger på mangfoldigheder er besværligt og kun udledt i simple tilfælde. De præsenterede metoder vil i stedet bruge stokastisk teori til at beskrive usikkerhed og variation af fordelinger på mangfoldigheder. Vi definerer en regressionsmodel til estimation af relationen mellem multiple Euklidiske kovariater og en respons variabel, der tager værdier på en mangfoldighed. De to data rum er forbundede via transport af en Euklidisk semi-martingal til mangfoldigheden. Denne transporterede semi-martingal kan modellere ikke-geodætiske regressions kurver og inkludere usikkerhed i den estimerede relation.

Derudover, introducerer vi to forskellige metoder til estimation af usikkerhed i populationer af billed data. Den første metode er en varianskomponentmodel, der separerer usikkerhed i billeder i en warp effekt og en spatiel intensitets effekt. Deformationer er defineret ved forskydningsfelter på et diskretiseret gitter af billed domænet. Baseret på maksimum-likelihood optimering er parametre for usikkerhedseffekterne estimeret simultant med et billed atlas. Den anden model beskriver usikkerhed ved stokastisk deformation af billeder. Den stokastiske deformation er modelleret som en stokastisk bevægelse i rummet af diffeomorfier inden for de givne rammer af LDDMM. Estimation af parametre, der beskriver støj-korrelation og lokal variation af billeder, er udført ved at matche data momenter mod momenterne for den stokastiske deformation.

Software pakker udviklet primært til Deep Learning opgaver er generelt beregningsmæssigt optimeret og kan inkorporere beregninger på GPU'er, parallelle udregninger og indeholde muligheden for symbolske kalkulationer. En del af arbejdet i denne afhandling viser, hvordan disse software pakker kan bruges til koncis implementering af koncepter fra differential geometri og ikke-lineær statistik.

Acknowledgements

The past three years have been some of the most eventful years of my life so far. I have had the great opportunity to travel around the world and meeting some of the most dominant people in my field. Working in the cross-section of statistics, mathematics and computer science has been fascinating and at times challenging. A lot of people have contributed at making my PhD time memorable and helped me through the tough times of the process.

First and foremost I would like to thank my supervisor, Stefan Sommer, for guiding me through three years of ups and downs. You have pushed me into delivering my best, and your office door was always open whenever I needed your supervision. This thesis would not have been possible without your constant support and advice.

I am genuinely grateful to Tom Fletcher for letting me visit his group at the University of Utah. It was five months of great experiences and where I learned a lot from being part of a large research group. Thank you for sharing your knowledge and for being a mentor during my stay in Utah.

Moreover, I would like to thank Kris Campbell for a ton of interesting theoretical discussions and a helping hand whenever needed, Sarang Joshi for discussing your ideas and thoughts with me particularly during my stay in Utah, and Alexis Arnaudon for multiple great collaborations and for letting me visit you in London. I wish that more collaborations will be possible in the future.

This thesis would not have been a realisation without the support of CSGB Centre for Stochastic Geometry and Advanced Bioimaging funded by a grant from the Villum Foundation. Being part of a diverse research group, helped me gain perspective on the work I conducted.

Going through the process of a PhD thesis is hard and can be difficult to describe. Therefore, it has been incredible to have so many great and fun colleagues and officemates who understand the rollercoaster ride the PhD life can be. Thank you for all the parties, social events and for just being great friends whenever needed. You made the long work days enjoyable. Thank you Yova and Mareike for proof reading parts of my thesis and giving useful comments for improvements, and especially thank you to Jacob and not least Mathias for helping me through the final days of my PhD.

As a last point, I would like to say the biggest thank you to my family. Thank you for the support and for being available whenever I needed an encouraging chat. You have always believed in me also when I stopped doing it myself, and for that, I am most grateful.

Contents

List of Figures

CHAPTER 1

Introduction and Background

1.1 Introduction

Until recent years statistical data has mainly been elements of Euclidean spaces. However, the fast improvement of data collecting techniques increases the complexity of data and challenges the well-founded methodology of statistical analysis. Several statistical methods rely on vector space structure of the considered data space. With the new technology, it is, however, no longer a consequence that data obey the vector space properties, e.g. being closed under the normal operations such as addition. Examples of non-linear data include Medical images, 3D constructions of shapes, curves, or landmark representation of objects, where either contours or special features are represented by landmark points. Examples of data objects are shown in Fig. 1.1.

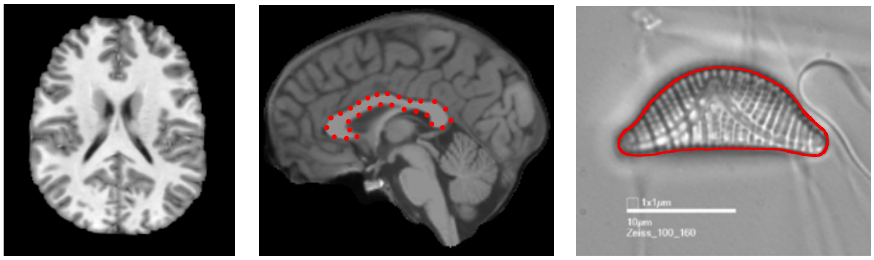


Figure 1.1: Examples of non-linear data: (left) An MR image of a human brain [71]. (middle) A landmark representation of a corpus callosum shape. The MR image was taken from the ADNI database (adni.loni.usc.edu). (right) A curve representation of the shape of a diatom [54].

The thesis aims at presenting generalised statistical methods capable of modelling uncertainty and variation in non-linear data populations. The lacking vector space structure of the data space makes the task of performing

statistical analysis hard as the generalised methods have to incorporate theory from differential geometry. Analysing non-linear data, therefore, combines concepts from statistics, differential geometry, and computer science. Working at the intersection of these fields has been both fascinating and challenging at times. The three years of the PhD have resulted in work which naturally partitions into 2 main topics:

1. Uncertainty and variation estimation in complex data structures.

2. Concise implementation of non-linear statistical methods using symbolic software packages.

The first topic is the main objective of the thesis and includes the work presented in Chapter 2-5. The common goal of the papers is to introduce uncertainty estimation in generalised statistical methods. The considered methods are regression models and longitudinal models describing the evolution of data objects over time. Estimating the variation in data populations is an essential part of statistical data analyses. By knowing the variation in data distributions, we are able to perform statistical inference of the difference between populations and to categorise individuals in different classes, e.g. sick and healthy. Generalising uncertainty to complex data without the vector space structure is non-trivial as even distributions are hard to define on such non-linear data spaces. In Section 1.4 and 1.7 we describe different approaches for defining distributions on non-linear data spaces.

The second topic concerns the problem of implementing non-linear statistical methods. We show how implementation tools, primarily developed for Deep Learning tasks, can be naturally used for concise implementation of concepts from differential geometry and non-linear statistics. The papers enclosed in Chapter 6 and 7 present a software library in the deep learning framework Theano. The library contains implementations of the basic theory of differential geometry and non-linear statistics. Additionally, the manuscript in Chapter 8 describe how non-linear statistical methods can be used to perform analysis of high-dimensional data, by representing data objects in a lower dimensional non-linear latent space trained by a Variational Autoencoder [13].

In the following sections, we give an introduction to the background theory and methodology inspiring the work presented in Chapter 2-8. This cover a brief presentation of assumptions made for the non-linear data spaces, a generalisation of distributions, linear regression generalisations, and stochastic dynamics as a method for modelling uncertainty in non-linear data populations. The following sections describe methods relevant for the purpose of the presented work and do not contain an extensive background study. For more references on each subject see the corresponding chapters.

1.2 Why Non-linear Statistics?

So what does it mean for data to be non-linear? Data objects are called non-linear whenever the data space lacks the usual vector space properties, e.g. being closed under addition. A common example of a non-linear space is the sphere S^2.

To perform statistical analysis on non-linear data, we need general assumptions on the properties of the data space. A common assumption is that data are elements of a manifold. A d-dimensional manifold, $\mathcal{M}$, is a potentially non-linear space which is locally homeomorphic to the Euclidean space $\mathbb{R}^d$. Manifolds behave like Euclidean spaces in a local neighbourhood of a point $p \in \mathcal{M}$, however, globally it can be highly non-linear. Examples of simple non-trivial manifolds are the circle S^1, the sphere S^2, or the space of landmark representations of Corpus Callosum shown in Fig. 1.1. We equip $\mathcal{M}$ with a metric g to define the concept of distance on the data space. The pair $(\mathcal{M}, g)$ is called a Riemannian manifold, where g denotes the Riemannian metric. In this thesis, we restrict attention to data on Riemannian manifolds.

Even though manifolds can be approximated by a vector space in a local neighbourhood of any point $p \in \mathcal{M}$, applying Euclidean statistical methods to non-linear data could result in biased estimates, which potentially escape the data space. As simple examples, consider the data samples on the unit sphere, S^2, visualised in Fig. 1.2. Without knowing the structure of the data space, a natural procedure would be to treat the sample points as elements of the ambient space $\mathbb{R}^3$. Estimating the mean of the data sample by the regular average estimator (left of equation of (1.1)) results in estimated mean values which are not elements of the data space (see Fig. 1.2).

$$\bar{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \bar{\mu}_{\text{FM}} = \operatorname*{arg\,min}_{y \in \mathcal{M}} \sum_{i=1}^{n} dist(y, x_i)^2 \tag{1.1}$$

Instead, we seek an intrinsic mean which is an element of the data space and contains similar properties as the observed data sample. An intrinsic mean estimator is the Fréchet mean, named after Maurice Fréchet, presented in [38]. The Fréchet mean is based on a distance function $dist\colon \mathcal{M} \to \mathbb{R}$ on the data space $\mathcal{M}$ and is defined as the set of points $\mu \in \mathcal{M}$ minimising the total distance to the data sample. The empirical estimator for the Fréchet means is given as the right equation of (1.1). Notice that existence and uniqueness of the Fréchet mean is not ensured as it is a solution to a minimisation problem. For an example where the Fréchet mean is not unique consider the case in the left plot of Fig. 1.2. For data in $\mathbb{R}^n$ equipped with the usual Euclidean distance, the Fréchet mean coincides with the traditional mean on vector spaces.

Defining the concept of a mean value on manifolds is just the first step in fitting distributions to data, but equally important is it to describe variation

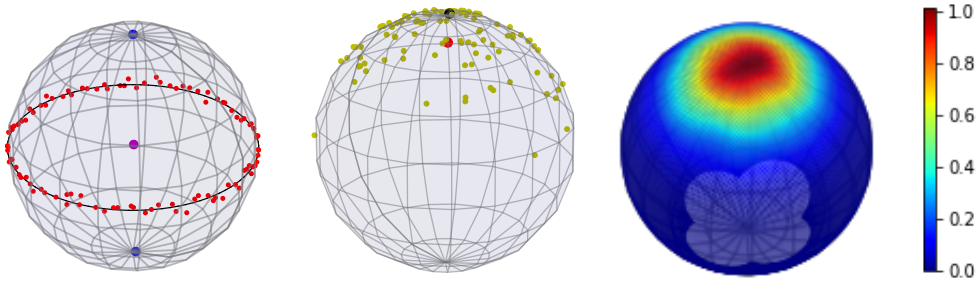


Figure 1.2: (left) A data sample on the unit sphere S^2 shown in red. The magenta point is the normal Euclidean average as defined in (1.1) left, while the blue points denotes the intrinsic emperical Fréchet mean set (see (1.1) right). (middle) A data sample on the upper hemisphere with mean $\mu = (0, 0, 1)$ generated by Brownian motions on S^2. The red point denotes the Euclidean average and the black the estimated emperical Fréchet mean. (right) An estimated distribution on S^2.

in data samples. Modelling variation in manifold-valued data is not an easy task due to the possible curvature of the data space. In most of the methods presented in this work, we propose to apply stochastic theory to describe uncertainty in data samples. Stochastic processes can be defined chart free and intrinsically on manifolds making it possible to obtain global distributions on the non-linear space. Fig 1.2 shows an example of a distribution on the unit sphere, S^2.Section 1.7 gives a brief introduction to stochastic processes on manifolds and how these define intrinsic distributions on the non-linear data spaces.

1.3 Riemannian Manifolds

The following section is based on [76, 22]. As described above, we make the general assumption that the considered data spaces are Riemannian manifolds, $(\mathcal{M}, g)$. The metric g is defined for any element $p \in \mathcal{M}$ and changes smoothly between tangent spaces of $\mathcal{M}$. Let $\mathfrak{X}(\mathcal{M})$ denote the space of smooth vector fields on $\mathcal{M}$. The manifold is endowed with a connection, $\nabla \colon \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M})$, sending a pair of smooth vector fields to a smooth vector field. A connection ∇ describes the transportation of tangent vectors along a curve γ on $\mathcal{M}$, hence connecting nearby tangent spaces. The transportation defines an isomorphism between tangent spaces called parallel transport. A connection preserving the metric across tangent spaces is called a metric connection. Throughout the thesis, we consider a metric connection called the Levi-Civita connection. Let $x \in \mathcal{M}$ and consider a chart (U, ϕ) around x, with

local coordinates $\phi(x) = (x_1, \ldots, x_d) \in \mathbb{R}^d$. The corresponding coordinate basis vectors are $\partial_1, \ldots, \partial_d$, for $\partial_i = \frac{\partial}{\partial x_i}$, representing a basis for the tangent spaces in U. If the coordinate representation of the metric g is given by the matrix $(g_{ij})_{ij}$, where $g_{ij} = \langle \partial_i, \partial_j \rangle$, the connection ∇ satisfies the equation,

$$\nabla_{\partial_i} \partial_j = \Gamma_{ij}^k \partial_k, \tag{1.2}$$

for the Christoffel symbols, Γ_{ij}^k. The Levi-Civita connection is the affine metric connection with the Christoffel symbols,

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl} (\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}), \tag{1.3}$$

where g^{kl} denotes the coordinate representation of the inverse of the metric.

A curve, γ_t, on $\mathcal{M}$ is a called a geodesic curve if the vector field along γ is parallel to the curve it self with respect to the connection ∇, i.e. $\nabla_{\dot{\gamma}_t} \dot{\gamma}_t = 0$. As in Euclidean space, $\mathbb{R}^n$, geodesics are length minimising and with zero acceleration. A curve $\gamma_t = (\gamma_t^i)_{i=1}^d$ represented by local coordinates in the chart (U, ϕ) with $\gamma_0 \in \mathcal{M}$ and $\dot{\gamma}_0 = v_0 \in T_{\gamma_0}\mathcal{M}$ is a geodesic if and only if it solves the geodesic equation,

$$\ddot{\gamma}_t^k + \dot{\gamma}_t^i \dot{\gamma}_t^j \Gamma_{ij}^k(\gamma_t) = 0. \tag{1.4}$$

For a point $p \in \mathcal{M}$, the exponential map, $\mathrm{Exp}_p \colon T_p\mathcal{M} \to \mathcal{M}$, is defined by $\mathrm{Exp}_p(v) = \gamma_1(v)$ for a geodesic γ_t with $\gamma_0 = p$ and $\dot{\gamma}_0 = v$. The exponential map, Exp_p, is only invertible in a neighbourhood U of p. Consider the subset $V \subseteq T_p\mathcal{M}$ of tangent vectors v making $\mathrm{Exp}_p(tv)$ a minimising geodesic for all $t \in [0, 1]$, but not on the extended interval $t \in [0, 1 + \varepsilon)$, for some $\varepsilon > 0$. The set $\mathcal{M} \setminus \mathrm{Exp}_p(V)$ is called the cut locus of p and denoted $C_{\mathcal{M}}(p)$. Let $U = \mathcal{M} \setminus C_{\mathcal{M}}(p)$. The inverse of the exponential map, $\mathrm{Log}_p \colon U \to T_p\mathcal{M}$, is called the logarithm map. For a point $q \in U$, $\mathrm{Log}_p(q)$ returns the unique tangent vector v s.t. $\mathrm{Exp}_p(v) = q$. The Riemannian distance on U is then defined by

$$dist(p, q) = \|\mathrm{Log}_p(q)\|_p^2,$$

where the norm is induced by the metric g. Chapter 6 and 7 provide a more detailed description of differential geometry and describe simultaneously how the Deep learning framework Theano can be used to make implementation of the mathematical theory.

1.4 Random Variables on Manifolds

When analysing data samples in Euclidean spaces, there is a wide range of well-known distribution families available. These include Normal, Poisson, Beta, and Uniform. A common feature of these distributions is that there exists a closed form expression of the density function with respect to the

Lebesgue measure. Defining similar standard distribution families on manifolds is non-trivial, and it is in many cases impossible to describe distributions on $\mathcal{M}$ by a closed-form expression of a density function. However, to even talk about distributions and random variables on manifolds, we first need to define a probabilistic setting. The section is based on [22, 100].

Consider a probability space $(\Omega, \mathcal{F}, P)$ and a measurable space $(\mathcal{M}, \mathcal{A})$, where $\mathcal{F}$ and $\mathcal{A}$ are Borel σ-algebras generated by open sets. In Euclidean space, the Lebesgue measure is a common choice of a standard measure in $\mathbb{R}^n$ as it describes the area or volume of a set $A \subseteq \mathbb{R}^n$. Additionally, several distributions are absolute continuous with respect to the Lebesgue measure making it possible to describe distributions by a closed-form expression of the density functions. The natural generalisation of the Lebesgue measure on manifolds is the Volume measure, which can be described in local coordinates. Consider a local finite atlas (U_α, ϕ_α) with coordinate basis $\partial_i^\alpha = \frac{\partial}{\partial x_i^\alpha}$, $i = 1, \ldots, d$. We denote the coordinate representation of the metric, g, in the chart (U_α, ϕ_α), by the matrix $G^\alpha = (g_{ij}^\alpha)_{i,j=1,\ldots,d} = (g(\partial_i^\alpha, \partial_j^\alpha))_{i,j}$. Given a partition of unity $\{\rho_\alpha\}$ the volume measure of A is hence,

$$\begin{aligned} \mathrm{Vol}(A) &= \sum_\alpha \int_{\phi_\alpha(A \cap U_\alpha)} \left(\rho_\alpha \sqrt{\det(G^\alpha)}\right) \circ \phi_\alpha^{-1} \, dx_1^\alpha \cdots dx_d^\alpha \qquad (1.5) \\ &= \sum_\alpha \int_{\phi_\alpha(A \cap U_\alpha)} \rho_\alpha(\phi_\alpha^{-1}(x_1^\alpha, \ldots, x_d^\alpha)) \sqrt{\det(G^\alpha(\phi_\alpha^{-1}(x_1^\alpha, \ldots, x_d^\alpha)))} \, dx_1^\alpha \cdots dx_d^\alpha, \end{aligned}$$

where $dx_1^\alpha \cdots dx_d^\alpha$ denotes the Lebesgue measure of $\mathbb{R}^d$. The volume measure, therefore, maps the set $A \subset \mathcal{M}$ to $\mathbb{R}^d$ and measures the volume in the local chart weighted by the metric.

Based on the volume measure we can define integration of real-valued functions. Let $f \colon \mathcal{M} \to \mathbb{R}$ be a compactly supported continuous function over a compact subset $A \subset \mathcal{M}$. The integral of f with respect to the Volume measure is given as,

$$\int_A f \, d\mathrm{Vol} = \sum_\alpha \int_{\phi_\alpha(A \cap U_\alpha)} (f \circ \phi_\alpha^{-1}) \left(\rho_\alpha \sqrt{\det(G^\alpha)}\right) \circ \phi_\alpha^{-1} \, dx_1^\alpha \cdots dx_d^\alpha. \quad (1.6)$$

Intuitively, this integral corresponds to regarding f, via $f \circ \phi^{-1} \colon \mathbb{R}^d \to \mathbb{R}$, as a function from a subset of $\mathbb{R}^d$ to $\mathbb{R}$ and measure it with respect to the Lebesgue measure on $\mathbb{R}^d$.

A random variable X on $\mathcal{M}$ is a measurable function between the probability space $(\Omega, \mathcal{F}, P)$ and the measurable space $(\mathcal{M}, \mathcal{A})$. The measure on $\mathcal{M}$ under the random variable X is the transformation $X(P)$, which define the distribution of X. The measure of a set $A \in \mathcal{A}$ under $X(P)$ is written as $P(X \in A)$. Based on measures on $\mathcal{M}$, assume that the random variable X on $\mathcal{M}$ is integrable and that the distribution of X is absolutely continuous with respect to a measure ν on $\mathcal{M}$ defining a density function p of $X(P)$. The

mean of the transformation of X under a real-valued function f is defined by

$$\mathbb{E}[f(X)] = \int_{\Omega} f(X)\, dP = \int_{\mathcal{M}} f(x)\, dX(P)(x) = \int_{\mathcal{M}} f(x)\, p(x)\, d\nu(x). \quad (1.7)$$

The above mean is an element of $\mathbb{R}$ and does therefore not describe mean values of data distributions on $\mathcal{M}$. As a consequence, other methods have to be taken into account to define means of distributions on manifolds. One example of a generalisation of a mean on $\mathcal{M}$ is the Fréchet mean briefly described in Section 1.2. The Fréchet mean is the set of points minimising the variance [100],

$$\text{Var}_X(y) = \mathbb{E}[dist(y, X)^2], \quad (1.8)$$

where the mean value is well-defined as an integral over the real-valued function $dist(y, X)$. The problem with the Fréchet mean is that it is not unique and does not always exist as a global minimum. In [58] it was proposed to consider local minima of the variance (1.8), instead of the global minima defining the Fréchet mean set. This mean is called the Karcher mean and has been shown to be unique for a sufficiently locally defined distribution [58, 60]. For a more detailed discussion see also [100].

One way to describe the notion of variation of a random variable X is by defining covariance in the tangent space of the mean. In the following, we assume the existence of a unique mean μ. The covariance is defined in the tangent space $T_\mu\mathcal{M}$ by the Log_μ map presented above. It is a straightforward generalisation of the Euclidean notion of covariance, and obtained as the integral [100],

$$\Sigma(X) = \mathbb{E}[\text{Log}_\mu(X)\text{Log}_\mu(X)^T] \quad (1.9)$$

Later in Section 1.7, we present a way to determine the probability density function of the normal distribution family based on this definition of covariance.

In Euclidean space, another way of estimating the mean of a distribution is by maximum likelihood estimation. The likelihood function is based on the density function of a data distribution. Above we mentioned that determining the density function for a data distribution on a manifold is hard and that there does not always exist a closed form expression for the density function. However, [120] presented a method for obtaining an approximation of the likelihood function based on stochastic processes. The idea about using stochastic theory to describe distributions and variation in data samples is the primary objective of the thesis. We use stochastic processes to express uncertainty in regression models on manifolds and evolution models for medical images. In the rest of this chapter, we, therefore, introduce the deterministic version of the models which inspired the work presented in Chapter 2-5.

1.5 Geodesic Regression

In this section, we give a short description of the geodesic regression model [35] and the extension [48] which introduce a non-geodesic relation between data variables. There have been several methods for defining regression models on manifolds, these include kernel-based, extrinsic and intrinsic, non-parametric and parametric methods. For references see Chapter 2 and 3.

Regular linear regression for which both covariate variables x and the response y are elements of the same vector space is modelled by the linear relation,

$$y = \alpha + X\beta + \varepsilon, \tag{1.10}$$

for an intercept $\alpha \in \mathbb{R}$, a design matrix X with slope vector β and iid. normally distributed noise ε. Modelling the relation between x and y in vector spaces is straightforward as the space is closed under addition. However, when modelling the relation between variables with either the covariates or the response defined on a manifold $\mathcal{M}$, the data space is not closed under addition, and the regular linear regression model is no longer applicable.

Geodesic regression [35] makes a natural generalisation of the Euclidean linear regression model. Based on a single covariate x, geodesic regression models the relation to the response y via geodesic curves,

$$y_i = \mathrm{Exp}_{\mathrm{Exp}_p(x_i v)}(\varepsilon_i), \tag{1.11}$$

for a random tangent vector $\varepsilon_i \in T_{\mathrm{Exp}_p(x_i v)}\mathcal{M}$. The tangent vector v is equivalent to the slope of the linear regression, and the point $p \in \mathcal{M}$ describes the intercept.

An extension of geodesic regression was presented in [48]. Here, the polynomial regression model was generalised to manifolds by applying Riemannian polynomials. Polynomial regression is able to describe more flexible relations between covariates and response but is restricted to incorporate a single covariate variable, e.g. time.

These methods formed the base of the stochastic development regression presented in the papers of Chapter 2 and 3. The aim of the generalised regression model, from Chapter 2 and 3, is to define an intrinsic parametric regression model able to include multiple covariates, to model non-geodesic relations, and to incorporate both fixed and random effects.

1.6 Evolution of Shapes and the LDDMM Framework

This section describes the setup for analysing deformation of shapes, which is the topic of the papers presented in Chapter 4 and 5. In this respect, we introduce the Large Deformation Diffeomorphic Metric Mapping framework [12]

(LDDMM), which inspired the work presented in Chapter 5. The content of the section was based on [12] and [87].

Consider a manifold Ω consisting of for example curves, images, or landmark points (see Fig. 1.1). We define the deformation of an element $q \in \Omega$ as an action, $\phi.q$. The deformation of the object q can be defined in multiple ways, depending on the choice of regularisation. In the large deformation, framework deformations are defined as diffeomorphisms. However, deformations are not restricted to diffeomorphisms in general and in Chapter 4 we give an example of small deformations modelled as displacement fields on a discretised lattice.

The shape examples used throughout the thesis are landmark representations of shapes and deformation of images. The action of ϕ on an m-dimensional landmark representation, $q = (x_1, \dots, x_m) \in \mathbb{R}^{d \times m}$, is an action on each landmark, i.e. $\phi.q = (\phi(x_1), \dots, \phi(x_m))$. The action of a deformation on an image I is defined as the composition $\phi.I = I \circ \phi^{-1}$. Examples of deformation of shape objects are given in Fig. 1.3.

Let q_0, q_1 be two shapes in Ω. Analysing the difference between shapes can be based on analysis of the deformation ϕ matching q_0 to q_1, i.e. $\phi.q_0 = q_1$. However, in many situations, e.g. image registration, there does not exist an exact matching of shapes. Instead, consider inexact matching for which a deformation ϕ is obtained by minimising the energy,

$$E(\phi) = R(\phi) + S(\phi.q_0, q_1), \tag{1.12}$$

for a regularisation R and a similarity measure S.

Consider the space $\mathrm{Diff}(\Omega)$ of automorphisms of the smooth manifold Ω. The LDDMM framework defines deformations, ϕ, based on a time-dependent family of elements of the Lie group $\mathrm{Diff}(\Omega)$. Deformations are modelled as the endpoint of a flow of automorphisms solving the ordinary differential equation (ODE),

$$\frac{\partial}{\partial t}\phi_t = \phi_t \circ v_t. \tag{1.13}$$

In (1.13), $v_t \in \mathfrak{X}(\Omega)$, is a time-varying smooth velocity field over the domain Ω. The Lie algebra $\mathfrak{g} = T_{Id}\mathrm{Diff}(\Omega)$ of the automorphism group is the space of smooth velocity fields. The smoothness of the flow ϕ_t is uniquely determined by the time-varying velocity field v_t. To obtain a geodesic flow of automorphisms, a special class of velocity fields have to be considered, which we describe in the following.

Assume that the space of velocity fields $\mathfrak{X}(\Omega)$ is a Reproducing Kernel Hilbert space, with kernel K defining the metric, g_L, on the Lie algebra, $\mathfrak{g}$. Let $\phi \in \mathrm{Diff}(\Omega)$ be given and define the right-multiplication $R_\phi \colon \mathrm{Diff}(\Omega) \to \mathrm{Diff}(\Omega)$, $R_\phi(\psi) = \psi \circ \phi$. By right translating the metric g_L under the differential of the right-multiplication, a global metric can be defined on $\mathrm{Diff}(\Omega)$. The

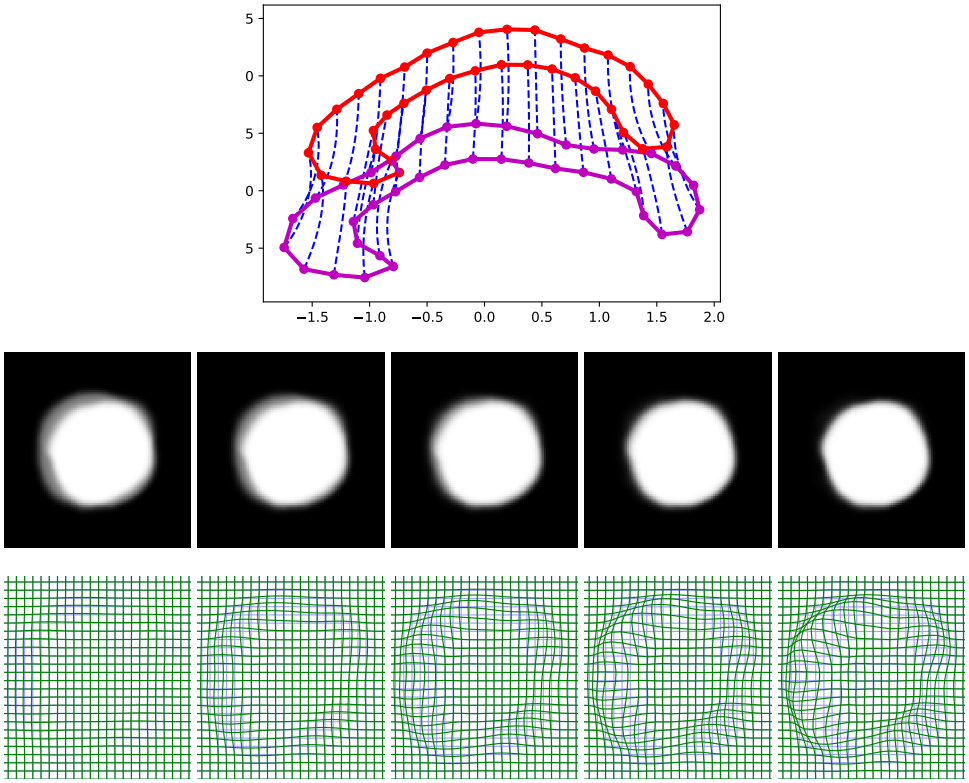


Figure 1.3: (1. row) Deformation of Corpus Callosum shape in landmark representation. (2. row) Matching of two shapes. The gray shape is matched into the white circle. (3. row) The corresponding deformation of the underlying grid. The matching was conducted by applying the FLASH code described in [140].

resulting right-invariant metric give rise to a simplification of the geodesic equations on Diff(Ω) called the Euler-Poincaré equations for diffeomorphisms (EPDiff). For an initial velocity field $v_0 \in \mathfrak{g}$, the EPDiff equation become,

$$\frac{d}{dt} m_t = -ad^*_{v_t} m_t, \tag{1.14}$$

for the dual $m_t = K^{-1} v_t$ and with ad^* denoting an adjoint operator. Solving (1.13) based on the time-varying velocity field v_t, solution to (1.14), result in a geodesic flow of automorphisms ϕ_t.

The deformation ϕ, a solution to the geodesic flow in the LDDMM framework, describes a deterministic deformation of a shape q_0. However, when considering the time evolution of, for example, anatomical objects, modelling the evolution as a deterministic path would not take into account the variation and uncertainty of the deformation. Including stochastic variation in the

LDDMM framework makes it possible to describe uncertainty in data samples over time. This is the topic of the paper presented in Chapter 5.

The above sections introduced different generalisations of statistical concepts to manifolds. Including estimation of uncertainty and variation to generalised models is non-trivial and often solved by modelling uncertainty in the tangent bundle. In this thesis, we suggest using stochastic theory to describe variation in data distributions on manifolds. The following section presents two ways of considering distributions on manifolds and gives an overview of different definitions for Brownian motions on $\mathcal{M}$.

1.7 Stochastic Dynamics on Manifolds

Defining standard distribution families, such as Normal, Beta, and Gamma distributions, on manifolds is non-trivial. Probability density functions are often defined based on desirable properties, e.g. by a decreasing probability mass when the distance to the mean increases. However, determining such suitable globally well-defined properties for distributions on manifolds is hard.

In [100], a generalisation of the normal distribution to manifolds was presented as the distribution defined by the density function minimising the negative entropy I,

$$I[X] = \mathbb{E}[\log(p(X))] = \int_{\mathcal{M}} \log(p(x)) dX(P)(x). \tag{1.15}$$

The minimisation is conditioned on knowing the mean $\mu \in \mathcal{M}$, assumed to be unique, and the covariance Σ, defined in (1.9), of the distribution of X. In the above integral, p denotes the density function for the distribution of the random variable X wrt. the lebesgue measure ν.

Under certain constraints on the minimisation of (1.15), described in [100], the normal distribution obtain the form,

$$p(x) = k \exp\left(-\frac{(\mathrm{Log}_\mu x)^T \Lambda \mathrm{Log}_\mu x}{2}\right), \tag{1.16}$$

for the symmetric concentration matrix Λ and with the normalization constant satisfying,

$$k^{-1} = \int_{\mathcal{M}} \exp\left(-\frac{(\mathrm{Log}_\mu x)^T \Lambda \mathrm{Log}_\mu x}{2}\right) d\nu(x). \tag{1.17}$$

The above definition of normal distributions result in a closed expression for the probability density function which resembles the probability density function for a normal distribution in $\mathbb{R}^d$.

An alternative method for generalisation of distributions, can be based on stochastic theory. An example is the limiting distribution of a vector, B_t of d independent standard Brownian motions $B_t^1, \dots, B_t^d$ for $t \in [0, 1]$ which resemble a standard normal distribution in $\mathbb{R}^d$. A Brownian motion, B_t in $\mathbb{R}$, is an almost surely continuous stochastic process with independent increments and $B_t - B_s \sim \mathcal{N}(0, t - s)$ for any time points $t > s \geq 0$. The limit distribution of B_t is the distribution at $t = 1$. A general normal distribution with mean μ and covariance matrix Σ can be obtained by the limit distribution of a random variable solution to the stochastic differential equation,

$$dX_t = \mu dt + \Sigma^{\frac{1}{2}} dB_t, \tag{1.18}$$

i.e. for $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d \times d}$, a d-dimensional vector B_t of d independent standard Brownian motions $B_t^1, \dots, B_t^d$ in $\mathbb{R}$, and X_t solving (1.18), then $X_1 \sim \mathcal{N}(\mu, \Sigma)$ [72]. Due to the potential non-linearity of the manifold, it is in general not possible to take long steps on the space. Moving around a manifold is instead defined by infinitesimal steps based on tangent vectors at the current point of location. The transition distribution of a stochastic process solution to a SDE is obtained by infinitesimal steps in tangent spaces. Hence, defining distributions on manifolds in this way is a natural approach and will be a focal point of this dissertation.

To talk about normal distributions on $\mathcal{M}$, obtained via stochastic differential equations, we first need to define Brownian motions on manifolds.

1.7.1 Brownian Motions on Riemannian Manifolds

There are multiple ways to define Brownian motions on a Riemannian manifold $(\mathcal{M}, g)$. In this section, we present a subset of these methods. The content of the section is based on [52].

Brownian motions in Euclidean space is defined as the stochastic process generated by the Laplace Beltrami operator. The Laplace Beltrami operator is defined as the divergence of the gradient of a function f,

$$\Delta f = \text{div} \nabla f. \tag{1.19}$$

There exist a natural generalisation of the Laplace Beltrami operator in Euclidean space to a Laplace Beltrami operator on a manifold $\mathcal{M}$, $\Delta_{\mathcal{M}}$. This operator is given as (1.19) based on the divergence and gradient of f on manifolds. For a metric g and local coordinates $\partial_1, \dots, \partial_d$ the Laplace Beltrami operator in local coordinates is given by,

$$\Delta_{\mathcal{M}} f = \frac{1}{\sqrt{G}} \frac{\partial}{\partial x_j} \left(\sqrt{G} g^{ij} \partial_i f \right) = g^{ij} \frac{\partial^2}{\partial x_i \partial x_j} f + g^{jk} \Gamma^i_{jk} \frac{\partial}{\partial x_i} f. \tag{1.20}$$

Based on the generalised Laplace Beltrami operator Brownian motion on manifolds can be defined as the diffusion process generated by $\frac{1}{2}\Delta_{\mathcal{M}}$. From

the local representation of $\frac{1}{2}\Delta_{\mathcal{M}}$, a Brownian motion on $\mathcal{M}$ can be described in local coordinates as an Itô SDE,

$$dW_t^i = -\frac{1}{2}g^{lk}\Gamma_{kl}^i(W_t)dt + \sqrt{g^{-1}}_k^i dB_t^k. \tag{1.21}$$

Here, g^{-1} denotes the inverse metric matrix $g^{-1} = (g^{ij})_{ij}$, Γ_{kl}^i the Cristoffel symbols of the connection, and B_t^k a real-valued Brownian motion. The presented Itô SDE is described in local coordinates, and is therefore dependent on the chosen chart and not globally defined.

To define a global, chart free representation of a Brownian motion on $\mathcal{M}$, we use the following result [52]. Let V_i, V_0 be vector fields on the tangent bundle $T\mathcal{M}$ and consider a Hörmander operator L on the form,

$$L = \frac{1}{2}\sum_{i=1}^{d} V_i^2 + V_0. \tag{1.22}$$

The stochastic process solving the Stratonovich SDE

$$dX_t = V_0(X_t)dt + V_i(X_t) \circ_S dB_t^i, \tag{1.23}$$

is generated by the operator L. This means that if it is possible to write the Laplace Beltrami operator $\Delta_{\mathcal{M}}$ as in (1.22), then we obtain a Stratonovich representation of a globally defined, chart free Brownian motion on $\mathcal{M}$. However, $\Delta_{\mathcal{M}}$ cannot be described in this form. Consider instead the frame bundle $\mathcal{FM}$. Elements of $\mathcal{FM}$ is a tuple, (x, u), of a point x on the manifold $\mathcal{M}$ and a linear isomorphism $u\colon \mathbb{R}^d \to T_x\mathcal{M}$, defining a basis on the tangent space. An example of a basis of $T_x\mathcal{M}$ is the collection of tangent vectors ue_i, $i = 1, \ldots, d$, where e_i denotes the i'th canonical basis vector of $\mathbb{R}^d$. The canonical projection $\pi\colon \mathcal{FM} \to \mathcal{M}$ is defined by $\pi(x, u) = x$. The connection allows the splitting of the tangent space of the frame bundle into a horisontal part describing changes in the base point $x \in \mathcal{M}$, and a vertical part explaining changes in the basis u, i.e. $T\mathcal{FM} = H\mathcal{FM} \oplus V\mathcal{FM}$. For a visualisation of the frame bundle see Fig. 1.4. The canonical projection give rise to a one to one correspondance between the horizontal tangent space $T_{\pi u}\mathcal{M}$ and $H_u\mathcal{FM}$ through the lift $\pi^*\colon T_{\pi u}\mathcal{M} \to H_u\mathcal{FM}$. Considering the basis ue_i, $i = 1, \ldots, d$ described above, a basis for the horizontal tangent space $H_u\mathcal{FM}$ is generated by lifting the basis vectors ue_i to the horizontal tangent space, i.e. $H_i(u) = \pi^*(ue_i)$, $i = 1, \ldots, d$ denotes a basis for $H_u\mathcal{FM}$.

For Riemannian manifolds we can restrict attention to a subspace of the frame bundle called the Orthonormal frame bundle $\mathcal{OM}$. The orthonormal frame bundle consists of all points (x, u) where $x \in \mathcal{M}$ and u is an orthonormal basis of the tangent space $T_x\mathcal{M}$. The Laplace Beltrami operator on $\mathcal{M}$ can be lifted to the orthonormal frame bundle resulting in a Laplace Beltrami

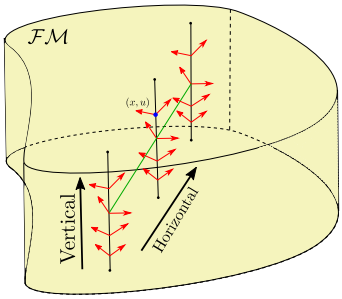


Figure 1.4: A visualisation of the frame bundle $\mathcal{FM}$. Each point (x, u) is a pair $x \in \mathcal{M}$ and a basis u of $T_x\mathcal{M}$. Walking horizontal in $\mathcal{FM}$ is equivalent with walking on $\mathcal{M}$ and parallel transporting u along. Walking vertically in $\mathcal{FM}$ fixes $x \in \mathcal{M}$ while smoothly moving across frames u of $T_x\mathcal{M}$.

operator $\Delta_{\mathcal{OM}}$ on $\mathcal{OM}$. The Laplace Beltrami operator $\Delta_{\mathcal{OM}}$ can be written as in (1.22) based on the horizontal basis vectors H_i, i.e.

$$\Delta_{\mathcal{OM}} = \sum_{i=1}^{d} H_i^2. \tag{1.24}$$

An intrinsic globally defined Brownian motion on $\mathcal{OM}$ can hence be obtained by solving the Stratonovich SDE,

$$dU_t = \sum_{i=1}^{d} H_i(U_t) \circ dB_t^i, \tag{1.25}$$

for a d-dimensional real valued Brownian motion B_t, and an initial point $U_0 = u_0 \in \mathcal{OM}$. A Brownian motion, W_t, on $\mathcal{M}$ is then defined by the canonical projection of U_t, i.e. $W_t = \pi(U_t)$. Notice that if the horizontal vector fields H_i are described in local coordinates, the local representation of $\pi(U_t)$ would be the same as the one presented in (1.21).

So why bother presenting this theory on Brownian motions on manifolds? The reason is that noise or uncertainty of data in Euclidean spaces is most often assumed to follow a normal distribution. Brownian motions on a time interval $t \in [0, 1]$ generates a standard normal distribution at $t = 1$. Hence modelling uncertainty by the limit distribution of a scaled Brownian motion on $\mathcal{M}$ results in uncertainty being independent and identically normally distributed. Some of the included work in this dissertation considers uncertainty defined based on Brownian motions in one of the described versions. As an example, the stochastic development definition is used in Chapter 2 and 3 to generalise regression models to $\mathcal{M}$. Chapter 4 and 5 model random deformations of images, wherein Chapter 5, the random deformation is based on a Stratonovich SDE on the group of automorphisms.

CHAPTER 2

Stochastic Development Regression on Non-Linear Manifolds

The following paper was accepted for IPMI 2017 and published in the conference proceedings,

- L. Kühnel and S. Sommer. Stochastic development regression on non-linear manifolds. In *Information Processing in Medical Imaging*, pages 53–64, Cham, 2017. Springer International Publishing

The work was conducted in collaboration with Stefan Sommer and presents a generalisation of linear regression on manifolds which is able to model flexible relations and include multiple covariate variables in the regression model.

Stochastic Development Regression on Non-Linear Manifolds

Line Kühnel and Stefan Sommer

Department of Computer Science (DIKU), University of Copenhagen, Denmark

Abstract

We introduce a regression model for data on non-linear manifolds. The model describes the relation between a set of manifold valued observations, such as shapes of anatomical objects, and Euclidean explanatory variables. The approach is based on stochastic development of Euclidean diffusion processes to the manifold. Defining the data distribution as the transition distribution of the mapped stochastic process, parameters of the model, the non-linear analogue of design matrix and intercept, are found via maximum likelihood. The model is intrinsically related to the geometry encoded in the connection of the manifold. We propose an estimation procedure which applies the Laplace approximation of the likelihood function. A simulation study of the performance of the model is performed and the model is applied to a real dataset of Corpus Callosum shapes.

Keywords: Regression, Statistics on Manifolds, Non-linear Statistics, Frame Bundle, Stochastic Development

2.1 Introduction

A main focus in computational anatomy is to study the shape of anatomical objects. Performing statistical analysis of anatomical objects is however challenging due to the non-linear nature of shape spaces. The established statistical theory for Euclidean data does not directly allow us to answer questions like: How does a treatment affect the deformation of an organ? or: Is it possible to categorize sick and healthy patients based on the shape of the subject's organs?

Shape spaces are typically non-linear and often equipped with manifold structure. Examples of manifold-valued shape data include landmarks, curves, surfaces, and images with warp variation. The lack of vector space structure for manifold-valued data implies that addition and scalar multiplication are not defined. Several concepts in statistics rely on addition and scalar

multiplication, these including mean value, variance, and regression models. Hence, in order to make inference on manifold-valued data, generalization of Euclidean statistical theory is necessary.

This paper focuses on generalization of regression models to manifolds. The aim is to model the relation between Euclidean explanatory variables and a manifold-valued response. The regression model has, as an example, applications in computational anatomy [136]. The proposed model can for example be used to analyze how age affects the shape of Corpus Callosum [35].

Several approaches have previously been proposed for defining normal distributions on manifolds [101, 118]. In [118], the distribution is defined based on Brownian motions in $\mathbb{R}^m$ and the fact that normal distributions on $\mathbb{R}^m$ can be defined as transition distributions of Brownian motions. The normal distribution on the manifold is then defined as the transition distribution of the stochastic development of the Euclidean Brownian motion [52]. The proposed regression model will be defined in a similar manner. The construction can be considered intrinsic as it only depends on the connection of the manifold, e.g. the Levi-Civita connection of a Riemannian manifold. It does not rely on linearization of the manifold, and it naturally includes the effect of curvature in the mapping of the stochastic processes.

In Euclidean linear regression, the relation between explanatory variables, $\boldsymbol{X}$, and a response variable, $\boldsymbol{y}$, is modeled by an affine function of $\boldsymbol{X}$,

$$\boldsymbol{y} = \boldsymbol{a} + \boldsymbol{X}\boldsymbol{b} + \boldsymbol{\varepsilon}. \tag{2.1}$$

Due to the lack of vector space structure, alternatives for modeling relations between the given variables, $\boldsymbol{X}$ and $\boldsymbol{y}$, are needed in the non-linear situation. Several ideas have previously been introduced and a selection of these will be described in Section 2.2.

In this paper, the regression model is considered as a transported linear regression defined in $\mathbb{R}^m$. This approach is inspired by the transport of normal distributions defined in [118]. Notice that the linear regression model (2.1) can be generalized to situations in which several observations are observed over time,

$$\boldsymbol{y}_t = \boldsymbol{a}_t + \boldsymbol{X}_t\boldsymbol{b} + \boldsymbol{\varepsilon}_t, \quad \text{for } t \in [t_1, t_2]. \tag{2.2}$$

Our approach suggests to define the regression model by transportation of stochastic processes, $Z_t = \boldsymbol{a}_t + \boldsymbol{X}_t\boldsymbol{b} + \boldsymbol{\varepsilon}_t$, in $\mathbb{R}^m$ on to the manifold in order to obtain the relation to the response variable, $\boldsymbol{y}$ (see Figure 2.1).

The paper will be structured as follows. In Section 2.2, we give a discussion on previous methods developed for regression on manifolds. Section 2.3 presents a short description of development of stochastic paths from a Euclidean space to the manifold. Section 2.4 introduces the proposed model,

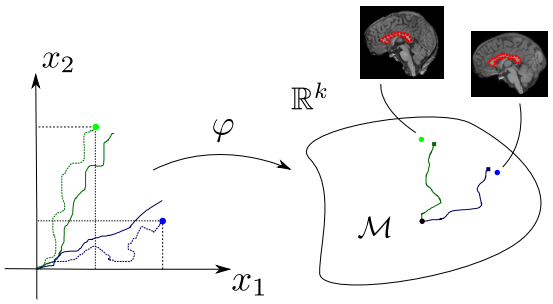


Figure 2.1: The idea behind the proposed regression model. Stochastic processes in $\mathbb{R}^m$ is transported to $\mathcal{M}$, by stohcastic development φ, to model the relation between the explanatory variables and the response $y \in \mathcal{M}$.

followed by a description of the estimation procedure in Section 2.5. In Section 2.6 and 2.7, illustrative examples are considered for the application and performance of the model. The paper is ended by a discussion of the defined model in Section 2.8.

2.2 Background

Multiple approaches have been proposed for generalizing regression models to non-linear manifolds. The methods consider the regression problem in different situations. In this paper we will consider the case of Euclidean exaplanatory variables and a manifold-valued response. There have been several works describing regression models for manifold-valued data in other situations [25, 9, 80, 124].

Regression models for describing the relation between a manifold-valued response and Euclidean explanatory variables have also previously been introduced. Examples include [78] in which an extrinsic regression model is introduced, and [115], which defines an intrinsic regression model where the parameter vector is estimated by minimizing the total sum of squares based on the Riemannian manifold distance. Another example is the geodesic regression model introduced in [35], which is a generalization of the linear regression model in Euclidean spaces. The relation is here modeled by a geodesic described by an initial velocity dependent on an explanatory variable and a starting point on the manifold.

In this paper, we will take a different view on how to relate the response and explanatory variables. Instead of considering the relation as being modeled by geodesics on the manifold as in [35], we will describe the relation by stochastic paths transported from the space of explanatory variables to the manifold. By defining the regression model using stochastic paths, we are able to model non-geodesic relations, incorporate several explanatory vari-

ables, and consider random effects in the model. Non-geodesic relations have been considered by others before. An example is [117] in which the geodesic regression model from [35] is generalized in order to model more complex shape changes. The regression function is in this case fitted by piecewise cubic splines that describes the variation of one explanatory variable. In [51], a regression model is introduced, in which the non-geodesic relation is obtained by time-warping. Others have proposed to model the non-geodesic relation by either a generalized polynomial regression model or by non-linear kernel-based regression [48, 137, 11, 10, 31]. On the contrary, [116] introduces the Hierarchical Geodesic Model which are able to consider several explanatory variables including random variables, but assumes nested observations and does only consider geodesic relations. A regression model, which incorporates both a non-geodesic relation and several explanatory variables, is proposed in [28]. This work defines an intrinsic regression model on Riemannian symmetric spaces, in which the regression function is obtained by minimizing the conditional mean of residuals defined by the log-map.

In addition to describing the proposed model, we perform estimation of model parameters by maximum likelihood using the transition density on the manifold. The model does not linearize the manifold as in many of the local regression models, but instead take into account the curvature of the manifold at each point as encoded in the connection through the mapping of the stochastic process.

2.3 Stochastic Development

In this section we give a brief description of stochastic development of curves in $\mathbb{R}^m$ to the manifold. The reader is referred to [52, 119, 122] for a deeper description of this concept.

Let $\mathcal{M}$ be a d-dimensional manifold provided with a connection ∇ and metric g. The connection is necessary for transportation of tangent vectors along curves on the manifold. A frequently used connection is the Levi-Civita connection coming from a Riemannian structure on $\mathcal{M}$. Let ∂_i for $i = 1, \ldots, d$ denote a coordinate frame on $\mathcal{M}$ and let dx^i be the corresponding dual frame. A connection ∇ is given in terms of its Christoffel symbols defined by $\nabla_{\partial_i}\partial_j = \Gamma_{ij}^k \partial_k$. For the Levi-Civita connection, the Christoffel symbols are given by

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl}(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}) \tag{2.3}$$

in which g_{ij} is the components of g in the coordinate basis, i.e. $g = g_{ij} dx^i dx^j$, and g^{ij} is the inverse components.

Consider the frame bundle $\mathcal{FM}$ being the set of tuples (y, ν) in which $y \in \mathcal{M}$ and ν is a frame for the tangent space $T_y\mathcal{M}$. Let $\pi \colon \mathcal{FM} \to \mathcal{M}$ be the

projection map given by $\pi(y, \nu) = y$ for $(y, \nu) \in \mathcal{FM}$. A smooth curve U_t on $\mathcal{FM}$ is a smooth selection of frames, i.e. for every $t \in I$, $U_t = (y_t, \nu_t)$ in which $\nu_t \colon \mathbb{R}^d \to T_{\pi(U_t)}\mathcal{M}$ is a frame.

Given a connection ∇, the tangent space of the frame bundle, $T\mathcal{FM}$, splits into a horizontal and a vertical part, $T\mathcal{FM} = H\mathcal{FM} \oplus V\mathcal{FM}$. The horizontal subspace explains infinitesimal changes of the base point on the manifold. On the other hand, tangent vectors in $V\mathcal{FM}$ describe changes of the frame ν keeping the base point fixed. Given a tangent vector $v \in T_y\mathcal{M}$ and a frame ν, a vector in $H_{(y,\nu)}\mathcal{FM}$ can be defined by horizontal lift. The horizontal lift of a tangent vector v is the unique horizontal vector $w \in H_{(y,\nu)}\mathcal{FM}$, satisfying $\pi_\star w = v$, where $\pi_\star \colon H_{(y,\nu)}\mathcal{FM} \to T_y\mathcal{M}$ is induced by the projection π. The horizontal lift of v will be denoted $h_l(v)$.

Consider a probability space $(\Omega, \mathcal{F}, P)$ and a stochastic process $X_t \colon \Omega \to \mathcal{W}(\mathbb{R}^m)$, where $\mathcal{W}(\mathbb{R}^m)$ denotes the path space of $\mathbb{R}^m$. The stochastic development of X_t to $\mathcal{FM}$ can be defined as a solution, U_t, of the Stratonovich stochastic differential equation,

$$dU_t = \sum_{i=1}^{d} H_i(U_t) \circ dX_t^i, \tag{2.4}$$

where $\circ$ symbolizes a Stratonovich stochastic differential equation. The vector fields $H_1, \ldots, H_d$ denotes a basis for the horizontal subspace of $T\mathcal{FM}$. Given a point $u = (y, \nu) \in \mathcal{FM}$, H_i are defined as $H_i(u) = h_l(\nu(e_i))$, $i = 1, \ldots, d$, where $e_1, \ldots, e_d$ is the canonical basis for $\mathbb{R}^d$. A path Y_t on the manifold $\mathcal{M}$ can then be obtained by the projection of U_t onto $\mathcal{M}$ by the projection map π, i.e. $Y_t = \pi(U_t)$.

Consider two processes X_t^1, X_t^2 in $\mathbb{R}^m$, $t \in [0, T]$ for $T > 0$, for which $X_0^1 = X_0^2 = \boldsymbol{x}_0$ and $X_T^1 = X_T^2$. If Y_t^1, Y_t^2 denotes the stochastic development of X_t^1 and X_t^2 respectively on $\mathcal{M}$, then it does not in general hold that $Y_T^1 = Y_T^2$ on $\mathcal{M}$ due to the curvature of the manifold.

2.4 Model

Let $\mathcal{M}$ be a d-dimensional manifold embedded in the ambient space $\mathbb{R}^k$ for some $k \geq d$ and consider a response variable y in $\mathcal{M}$. Let $\nu_{y_0} \colon \mathbb{R}^d \to T_{y_0}\mathcal{M}$ be a frame for the tangent space at a reference point $y_0 \in \mathcal{M}$. Assume that $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n \in \mathbb{R}^k$ are n realizations of $y \in \mathcal{M}$ and let $\boldsymbol{x}_i = (x_i^1, \ldots, x_i^m) \in \mathbb{R}^m$ denote the vector of explanatory variables for the i'th observation. Notice that the realizations of y are assumed to lie in the ambient space $\mathbb{R}^k$ and not required to be in $\mathcal{M}$. This construction allows for observations measured with noise which are not necessarily observed as elements of $\mathcal{M}$.

The strategy of the proposed model is to define stochastic processes according to the generalized linear regression in (2.2) and transport these to the

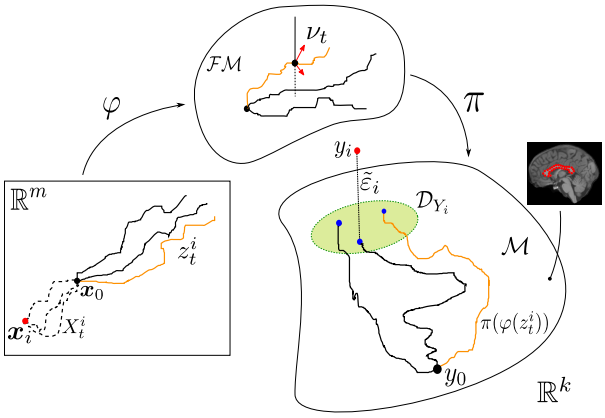


Figure 2.2: Illustration of the regression model. Stochastic processes z_t^i, defined in (2.5), are transported through the frame bundle $\mathcal{F}\mathcal{M}$ to $\mathcal{M}$, with stochastic development, φ. Each observation y_i is then modelled as a noisy member of the endpoint distribution of the transported z_t^i processes. The model supports cases where the endpoint noise $\tilde{\varepsilon}$ perturbes y_i in the ambient space $\mathbb{R}^k$ in which $\mathcal{M}$ is embedded.

manifold by stochastic development. All stochastic processes are defined for $t \in [0, T]$ for a $T > 0$. Consider for each observation i the stochastic process $z_t^i \colon \Omega \to \mathcal{W}(\mathbb{R}^m)$, solution to the stochastic differential equation,

$$dz_t^i = \beta dt + \tilde{W} dX_t^i + d\varepsilon_t. \tag{2.5}$$

The first term, βdt, is a fixed drift for $\beta \in \mathbb{R}^m$. $\tilde{W} dX_t^i$ is the dependence of the explanatory variables with $X_t^i \colon \Omega \to \mathcal{W}(\mathbb{R}^m)$ being a stochastic process satisfying $X_0^i(\omega) = 0$ and $X_T^i(\omega) = \boldsymbol{x}_i$ for $\omega \in \Omega$. The matrix $\tilde{W}$ is a $m \times m$-dimensional matrix with columns relating to the basis vectors of the frame ν_{y_0} on $\mathcal{M}$. Consider the matrix W with columns consisting of basis vectors of ν_{y_0}. If $\mathcal{M}$ has a Riemannian metric, then $W = U\tilde{W}$, in which U denotes a $d \times m$ orthonormal matrix with respect to the metric. Notice that this model can incorporate both fixed and random explanatory variables. If the j'th explanatory variable, x_i^j, is a random effect, X_t^{ij} is modeled as a Brownian bridge, while it for fixed effects are modeled as a constant drift. The random error, ε_t, is modeled as a multidimensional Brownian motion on $\mathbb{R}^m$.

The i'th observation y_i is modeled as a noisy endpoint of the stochastic development of z_t^i. If $m < d$ only a reduced frame $\tilde{\nu}_{y_0}$ is used for the stochastic development of z_t^i. The reduced frame is considered as we are only interested in the effect of frame vectors associated to the explanatory variables. The basis vectors of $\tilde{\nu}_{y_0}$ corresponds to the columns of W. Given the reference point $y_0 \in \mathcal{M}$, define stochastic processes Y_t^i as the stochastic development of z_t^i. Let $\mathcal{Y}_i^T \colon \Omega \to \mathcal{M}$ be a random variable following the distribution of

endpoints of the stochastic development Y_t^i. Then

$$y_i = \mathcal{Y}_i^T + \tilde{\varepsilon}_i, \tag{2.6}$$

where $\tilde{\varepsilon}_i \sim \mathcal{N}(\mathbf{0}, \tau^2 \mathbb{I}_d)$ represents the random measurement error that pulls the realization, y_i, from the manifold. In Figure 2.2, the two steps of the model are illustrated. First, the stochastic development of z_t^i are defined on the frame bundle and finally, this stochastic development is projected to the manifold.

Notice that in the case $\mathcal{M} = \mathbb{R}^k$ with the standard connection on $\mathbb{R}^k$, the proposed model reduces to the regular regression model for data in $\mathbb{R}^k$. Assume $y \in \mathbb{R}^k$ and that X_t^i is a vector from 0 to $\boldsymbol{x}_i$. Then β and y_0 relates to the intercept, W is the matrix of regression coefficients and ε_t and $\tilde{\varepsilon}$ the iid. random noise.

2.5 Estimation

The reference point y_0, the matrix W, the drift β, and the variance parameter τ^2 are the parameters of the model. These parameters can be estimated in several ways. This section describes a Laplace approximation of the marginal likelihood function which are used for finding optimal parameter estimates. We could alternatively use a Monte Carlo EM based procedure using simulations of the missing data, Y_t^i for $t \in [0, T]$, to optimize the complete data likelihood. This will be considered in future works.

Laplace approximation can be used to determine a linear approximation of a non-linear likelihood function [59]. Let θ denote the vector of parameters, and $d\boldsymbol{x}_t$ a discretization of the process X_t at $n_s + 1$ time-points. Hence $d\boldsymbol{x}_t$ is a vector of length $n \cdot m \cdot n_s$, in which n_s denotes the number of time steps, n the number of observations, and m the number of explanatory variables. Let $f(y|\theta)$ be the conditional density of the response $y \in \mathcal{M}$ given θ and $p(d\boldsymbol{x}_t|\theta)$ the density of the discretization of X_t given θ. To find the optimal parameter vector, θ, the following likelihood has to be optimized,

$$L(\theta; \boldsymbol{y}) = f(y|\theta) = \int f(y|d\boldsymbol{x}_t, \theta) p(d\boldsymbol{x}_t|\theta) d(d\boldsymbol{x}_t) = \int e^{-nh(d\boldsymbol{x}_t)} \, d(d\boldsymbol{x}_t), \tag{2.7}$$

where $h(d\boldsymbol{x}_t) = -\frac{1}{n} \log f(y|d\boldsymbol{x}_t, \theta) - \frac{1}{n} \log p(d\boldsymbol{x}_t|\theta)$. The Laplace approximation of L is then given by

$$L(\theta; \boldsymbol{y}) \approx f(y|d\boldsymbol{x}_t^o, \theta) p(d\boldsymbol{x}_t^o|\theta) (2\pi)^{\frac{mn_s}{2}} |\Sigma|^{\frac{1}{2}} n^{-\frac{mn_s}{2}}, \tag{2.8}$$

in which $d\boldsymbol{x}_t^o = \text{argmax}_{d\boldsymbol{x}_t}\{-h(d\boldsymbol{x}_t)\}$ and $\Sigma = \left(D^2 h(d\boldsymbol{x}_t)\right)^{-1}$, the inverse of the Hessian of $h(d\boldsymbol{x}_t)$. The approximated likelihood is then optimized wrt. θ to obtain the estimated parameters. In the following simulation study, the Laplace approximation is used for parameter estimation. The code for

the estimation algorithm as well as the simulation study below was implemented in Theano [127]. The code is available at https://bitbucket.org/stefansommer/theanodiffgeom.

2.6 Simulation Study

This section investigates properties of the model on simulated synthetic data. Two setups will be introduced, both considering landmark representations of shapes. The data are assumed to lie in a manifold defined in the LDDMM (Large Deformation Diffeomorphic Metric Mapping) framework [135].

In the LDDMM framework, deformations of shapes are modeled as smooth flows which are solutions to ordinary differential equations defined by vector fields. A point $q \in \mathcal{M}$ is a finite number of landmarks, $q = (x_1^1, x_1^2, \ldots, x_{n_l}^1, x_{n_l}^2)$. The metric on $\mathcal{M}$ is given by $g(v, w) = \sum_{i,j}^{n_l} vK^{-1}(x_i, x_j)w$, where K^{-1} denotes the inverse of a kernel K. In this simulation study K is the Gaussian kernel with standard deviation, $\sigma = 0.5$. Based on this metric the Levi-Civita connection can be obtained by calculating the Christoffel symbols defined in (2.3).

To begin with, we consider estimation of $\tilde{W}$ and y_0 and investigate the performance of the estimation procedure. The shapes that will be considered consists of 8 landmarks generated from the unit circle with landmarks located at $0, \frac{\pi}{4}, \frac{\pi}{2}, \ldots, \frac{3\pi}{2}, \frac{7\pi}{4}$ radians. The center plot of Figure 2.3 shows the unit circle with the chosen frame for each landmark. The number of explanatory variables are set to $m = 2$ and the variables are drawn from a normal distribution with mean 0 and standard deviation 2. The other parameters are set to

$$\tilde{W} = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix}, \ \tau = 0.1 \tag{2.9}$$

In Figure 2.3 is shown an example of simulated observations as well as the sample paths X_t^i. A total of 50 datasets were sampled, in which each consisted of 20 observations. For each simulated dataset, the $\tilde{W}$ matrix was estimated. Each of the estimated distrubtions for the entries of $\tilde{W}$ are shown in Figure 2.4. By the results, we conclude that the estimated parameters are fairly stable between the different simulations and that the true values are well centered in each distribution. For this simulation, the estimation procedure is thus able to estimate the true $\tilde{W}$ parameters that were specified in the model.

Three similar datasets, as explained above, were sampled with different number of observations, 20, 60 and 100 respectively. The matrix $\tilde{W}$ as well as the reference point y_0 were estimated for each of the three datasets. In this

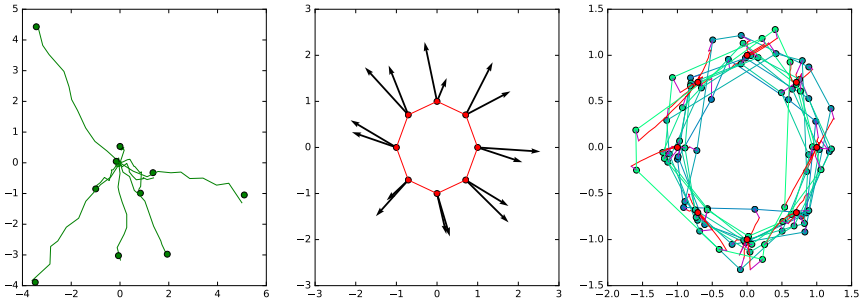


Figure 2.3: The figures show the simulation of a dataset. (left) The stochastic paths in $\mathbb{R}^m$ are shown, where the vector of explanatory variables for each observation i is represented by a green dot. (center) The true frame for the simulated data as well as the reference shape are plotted. (right) The simulated observations are shown, with the stochastic developments as the red processes.

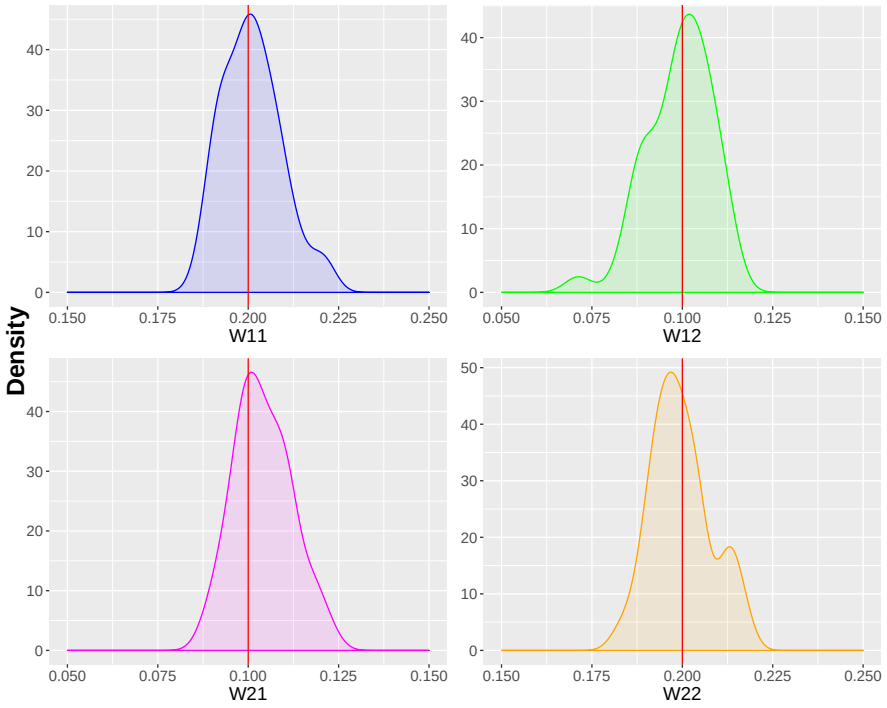


Figure 2.4: The distribution of the estimated $\tilde{W}$ parameters. The red horizontal lines show the true parameters given in (2.9).

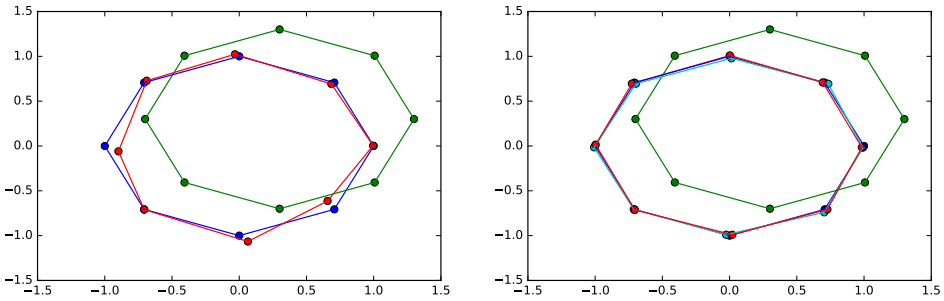


Figure 2.5: (left) The estimated reference point y_0 (red) for the dataset with 20 observations. (right) The estimated y_0 for 60 (cyan) and 100 (red) observations. In both plots, the initial (green) and the true reference circle (blue) are shown.

case, the estimated $\tilde{W}$ matrix was found to be

$$\hat{W}_{20} = \begin{pmatrix} 0.206 & 0.136 \\ 0.147 & 0.322 \end{pmatrix}, \hat{W}_{60} = \begin{pmatrix} 0.22 & 0.11 \\ 0.11 & 0.21 \end{pmatrix}, \hat{W}_{100} = \begin{pmatrix} 0.205 & 0.104 \\ 0.115 & 0.214 \end{pmatrix}$$

while the estimated reference points are shown in Figure 2.5. By increasing the number of observations, we conclude that the estimated parameters $\tilde{W}$ and y_0 converge towards the true parameters.

In the second study, we consider the problem of estimating the frame matrix U. In this case, each observation consists of 3 landmarks that were generated from a setup shown in Figure 2.6. We only consider one explanatory variable, meaning that only one frame vector has to be estimated for each landmark. The true frame vectors for each landmark was set to a vertical unit vector. In the estimation procedure, the frame vectors were initialized with the Euclidean linear regression estimate. In Figure 2.6 is shown the true (red), the initial (green) and the estimated frame (blue) for each landmark. The estimation procedure converges to a good estimate of the true frame. Estimation of the initial frame was considered for different number of observations, but the estimated frame did not seem to converge for increasing number of observations. The difference in the parameter estimates might therefore be a result of either the linear approximation of the likelihood or that the optimal solution of the initial frame is not unique.

2.7 Data Example

We now apply the model to a real dataset consisting of landmark representations of Corpus Callosum (CC) shapes. The model is used to describe the effect of age on CC shapes. The manifold considered is the same as that introduced in Section 2.6, but in this case $\sigma = 0.1$. Again the Levi-Civita connection is used.

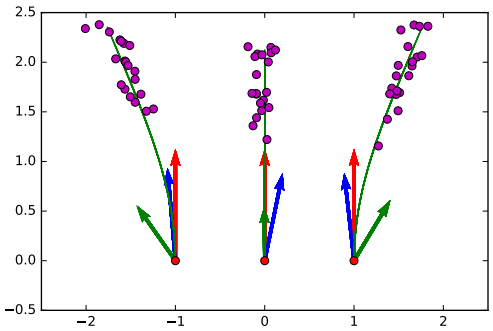


Figure 2.6: Comparison of the estimated (blue), initial (green) and true frame vectors (red).

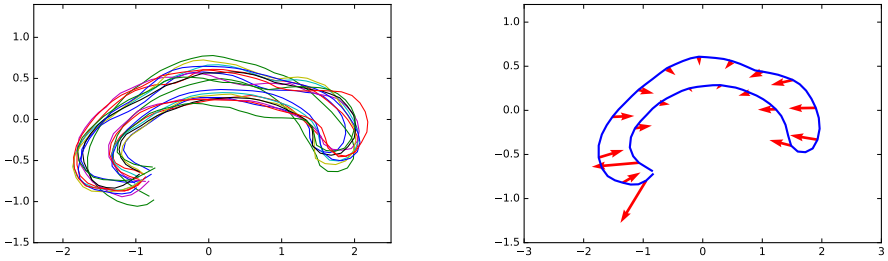


Figure 2.7: (left) A subset of the Corpus Callosum data. (right) The mean shape with the estimated frame for the 20 landmarks used in the model fitting.

A subset of the CC dataset is plotted in Figure 2.7. For model fitting, a dataset of 20 CC shapes was considered with age values ranging from 22 to 78. The model was fitted to CC shapes represented by a subset of 20 landmarks. We did not incorporate a drift term in the model, and only the frame and $\tilde{W}$ has been estimated. The refrence point was set to the mean shape (Figure 2.7) and $\tau = 0.1$.

The estimated frame for the 20 landmarks are shown in Figure 2.7 on top of the mean shape. The weight matrix was estimated as $\tilde{W} = -0.0002$. Given the low estimate of $\tilde{W}$ and hence a small frame matrix W, the result of this experiment suggests a low age effect on CC for these data.

2.8 Discussion

A method was proposed for modeling the relation between a manifold-valued response and Euclidean explanatory variables. The relation was modeled by transport of stochastic paths from $\mathbb{R}^m$ to the manifold. The stochastic paths defined on $\mathbb{R}^m$ was given as solutions to a stochastic differential equa-

tion with a contribution from a fixed drift, a stochastic process related to the explanatory variables, and a random noise assumed to follow a multidimensional Brownian motion. The response variable was then modeled as a noisy observation of a stochastic variable following the distribution of the endpoints of the transported process. The proposed model is intrinsic and based on a connection on the manifold without making linearization of the non-linear space. Moreover, a likelihood based estimation procedure were described using Laplace approximation of the marginal likelihood. We experimentally illustrated the model and the parameter estimation using a simulation study and a real data example.

Other procedures could be used for estimation of parameters. As an example, the Monte Carlo EM procedure could be used to optimize the complete data likelihood based on simulations of the missing data. Another example is to approximate the distribution of the response by moment matching.

An interesting problem to investigate is how to make variable selection in the model. As the contribution from the explanatory variables is defined in comparison with the frame basis vectors, one idea is to exclude those explanatory variables which corresponds to frame vectors parallel to the curve. These frame vectors will not contribute to the stochastic development and hence will not be important for explaining the relation to the response variable.

An important assumption of the manifold considered, is that the manifold is equipped with a connection. In this paper, the Levi-Civita connection was used, but several other connections could have been chosen. It would be interesting to explore how the choice of connection affects the model.

As it is possible to transport stochastic paths from a manifold to a Euclidean space, the model could be generalized to handle situations in which a Euclidean response variable is compared to manifold-valued explanatory variables. Based on such a model, one might be able to make categorization of individuals based on manifold-valued shapes.

CHAPTER 3

Stochastic Development Regression using Method of Moments

In the previous paper, a Laplace approximation was used to approximate the likelihood function in order to infer model parameters for the stochastic development regression. This required calculation of a high-dimensional Hessian matrix, making the optimisation procedure computationally infeasible. The paper in this chapter presents the same stochastic development regression, but where we investigate the application of the method of moments procedure for retrieving good estimates of the model parameters. The paper was joint work with Stefan Sommer and accepted for GSI 2017. The paper can be found in the conference proceedings,

- L. Kühnel and S. Sommer. Stochastic development regression using method of moments. In *International Conference on Geometric Science of Information*, pages 3–11. Springer, Cham, 2017

Notice that Section 3.7 was not part of the original paper, but has been included in the thesis to present an additional example on S^2 and adress further thoughts concerning the stochastic development regression.

Stochastic Development Regression using Method of Moments

Line Kühnel and Stefan Sommer

Department of Computer Science (DIKU), University of Copenhagen, Denmark

Abstract

This paper considers the estimation problem arising when inferring parameters in the stochastic development regression model for manifold valued non-linear data. Stochastic development regression captures the relation between manifold-valued response and Euclidean covariate variables using the stochastic development construction. It is thereby able to incorporate several covariate variables and random effects. The model is intrinsically defined using the connection of the manifold, and the use of stochastic development avoids linearizing the geometry. We propose to infer parameters using the Method of Moments procedure that matches known constraints on moments of the observations conditional on the latent variables. The performance of the model is investigated in a simulation example using data on finite dimensional landmark manifolds.

Keywords: Frame Bundle, Non-linear Statistics, Regression, Statistics on Manifolds, Stochastic Development.

3.1 Introduction

There is a growing interest for statistical analysis of non-linear data such as shape data arising in medical imaging and computational anatomy. Non-linear data spaces lack vector space structure, and traditional Euclidean statistical theory is therefore not sufficient to analyze non-linear data. This paper considers parameter inference for the stochastic development regression (SDR) model introduced in [66] that generalizes Euclidean regression models to non-linear spaces. The focus of this paper is to introduce an alternative estimation procedure which is simple and computationally tractable.

Stochastic development regression is used to model the relation between a manifold-valued response and Euclidean covariate variables. Similar to Brownian motions on a manifold, $\mathcal{M}$, defined as the transport of a Euclidean

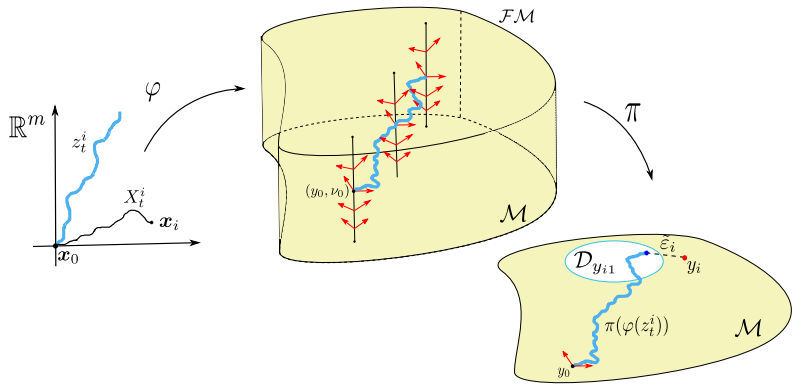


Figure 3.1: The idea behind the model. Normal linear regression process z_t^i defined in (3.1) is transported to the manifold through stochastic development, φ. Here $\mathcal{FM}$ is the frame bundle, π a projection map, and $\mathcal{D}_{y_{i1}}$ the transition distribution of $y_{it} = \pi(\varphi(z_t^i))$. The tangent bundle of $\mathcal{FM}$ can be split in a horizontal and vertical subspace. Changes on $\mathcal{FM}$ in the vertical direction corresponds to fixing a point $y \in \mathcal{M}$ while changing the frame, ν, of the tangent space, $T_y\mathcal{M}$. Changes in the horizontal direction is fixing the frame for the tangent space and changing the point on the manifold. The frame is in this case parallel transported to the new tangent space.

Brownian motion from $\mathbb{R}^n$ to $\mathcal{M}$, the SDR model is defined as the transport of a Euclidean regression model. A Euclidean regression model can be regarded as a time dependent model in which, potentially, several observations have been observed over time. Given a response variable $y_t \in \mathbb{R}^d$ and covariate vector $\boldsymbol{x}_t = (x_t^1, \dots, x_t^m) \in \mathbb{R}^m$, the Euclidean regression model can be written as

$$y_t = \alpha_t + \beta_t \boldsymbol{x}_t + \varepsilon_t, \quad t \in [0, 1], \tag{3.1}$$

where $\alpha_t \in \mathbb{R}^d$ and $\beta_t \in \mathbb{R}^{d \times m}$. A regression model can hence be defined as a stochastic process with drift α_t, covariate dependency through $\beta_t \boldsymbol{x}_t$, and a brownian noise ε_t. The SDR model is then defined as the transport of a regression model of the form (3.1), from $\mathbb{R}^d$ to the manifold $\mathcal{M}$. The trasnportation is performed by stochastic development described in Section 3.2. Fig. 3.1 visualizes the idea behind the model.

In [66], Laplace approximation was applied for estimation of the parameter vector. However, this method was computational expensive and it was difficult to obtain results for detailed shapes. Alternatively, a Monte Carlo Expectation Maximization (MCEM) method has been considered, but, with this method, high probability samples were hard to obtain, which led to an unstable objective function. As a consequence, this paper examines the Method of Moments (MM) procedure for parameter estimation. The MM procedure

is easy to apply and not as computationally expensive as the Laplace approximation. It is a well-known method for estimation in Euclidean statistics (see for example [97, 45, 29]), where it has been proven in general to provide consistent parameter estimates.

Several versions of the generalized regression model have been proposed in the case of manifold-valued response and Euclidean covariate variables. Local regression is considered in [137, 78]. The former defines an intrinsic local regression model, while [78] constructs an extrinsic model. For global regression models, [35, 93, 116] consider geodesic regression, which is a generalization of the Euclidean linear regression model. There have been several approaches for defining non-geodesic regression models on manifolds. An example is kernel based regression models, in which the model function is estimated by a kernel representation [10, 31, 94]. In [51, 48, 117], the non-geodesic relation is modelled by a polynomial or piecewise cubic spline function. Moreover, [115, 28] propose estimation of a parametric link function by minimization of the total residual sum of squares and the generalized method of moments procedure respectively.

The paper will be structured as follows. Section 3.2 gives a brief description of stochastic development and the frame bundle $\mathcal{FM}$. Section 3.3 introduces the SDR model and Section 3.4 describes the estimation procedure, Method of Moments. At the end, a simulation example is performed in Section 3.5.

3.2 Stochastic Development

This section gives a brief introduction to frame bundle and stochastic development. For a more detailed description and a reference for the following see [52]. Consider a d-dimensional Riemannian manifold $(\mathcal{M}, g)$ and a probability space $(\Omega, \mathcal{F}, P)$. Stochastic development is a method for transportation of stochastic processes in $\mathbb{R}^d$ to stochastic processes on $\mathcal{M}$. Let $z_t \colon \Omega \to \mathbb{R}^d$ denote a stochastic process for $t \in [0, 1]$. In order to define the stochastic development of z_t it is necessary to consider a connection on $\mathcal{M}$. A connection, ∇, defines transportation of vectors along curves on the manifold, such that tangent vectors in different tangent spaces can be compared. A frequently used connection, which will also be used in this paper, is the Levi-Civita connection of a Riemannian metric. Consider a point $q \in \mathcal{M}$ and let ∂_i for $i = 1, \ldots, d$ denote a coordinate frame at q, i.e. an ordered basis for $T_q\mathcal{M}$, with dual frame dx^i. A connection ∇ is locally determined by the Christoffel symbols defined by $\nabla_{\partial_i}\partial_j = \Gamma_{ij}^k \partial_k$. The Christoffel symbols for the Levi-Civita connection are given by $\Gamma_{ij}^k = \frac{1}{2} g^{kl} \left(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}\right)$, where g_{ij} denotes the coefficients of the metric g in the dual frame dx^i, i.e. $g = g_{ij} dx^i dx^j$, and g^{ij} are the inverse coefficients.

Stochastic development uses the frame bundle, $\mathcal{FM}$, defined as the fiber

bundle of tuples (y, ν), $y \in \mathcal{M}$ with $\nu \colon \mathbb{R}^d \to T_y\mathcal{M}$ being the frame for the tangent space $T_y\mathcal{M}$. Given a connection on $\mathcal{FM}$, the tangent bundle of the frame bundle, $T\mathcal{FM}$, can be split into a horizontal, $H\mathcal{FM}$, and vertical, $V\mathcal{FM}$, subspace, i.e. $T\mathcal{FM} = H\mathcal{FM} \oplus V\mathcal{FM}$. Fig. 3.1 shows a visualization of the frame bundle and the horizontal and vertical tangent spaces. The horizontal subspace determines changes in $y \in \mathcal{M}$ while fixing the frame ν, while $V\mathcal{FM}$ fixes $y \in \mathcal{M}$ and describes the change in the frame for $T_y\mathcal{M}$. Given the split of the tangent bundle $T\mathcal{FM}$, an isomorphism $\pi_{\star,(y,\nu)} \colon H_{(y,\nu)}\mathcal{FM} \to T_y\mathcal{M}$ can be defined. The inverse map $\pi^{\star}_{(y,\nu)}$ is called the horizontal lift and pulls a tangent vector in $T_y\mathcal{M}$ to $H_{(y,\nu)}\mathcal{FM}$. The horizontal lift of $v \in T_y\mathcal{M}$ is here denoted $v^{\star} \in H_{(y,\nu)}\mathcal{FM}$.

Let $e_1, \ldots, e_d$ be the canonical basis of $\mathbb{R}^d$ and consider a point $(y, \nu) \in \mathcal{FM}$. Define the horizontal vector fields, $H_1, \ldots, H_d$, by $H_i(\nu) = (\nu e_i)^{\star}$. The vector fields $H_1, \ldots, H_d$ then form a basis for the subspace $H\mathcal{FM}$. Given this basis for $H\mathcal{FM}$, the stochastic development of a Euclidean stochastic process, z_t, to the frame bundle $\mathcal{FM}$ can be found by the solution to the Stratonovich differential equation $dU_t = H_i(U_t) \circ dz_t^i$, where Einsteins summation notation is used and $\circ$ specifies that it is a Stratonovich differential equation. The stochastic development of a process $z_t \in \mathbb{R}^d$ with reference point (y, ν) will be denoted $\varphi_{(y,\nu)}(z_t)$. A stochastic process on $\mathcal{M}$ can then be obtained by the projection of U_t to $\mathcal{M}$ by the projection map $\pi \colon \mathcal{FM} \to \mathcal{M}$.

3.3 Model

Consider a d-dimensional manifold $\mathcal{M}$ equipped with a connection ∇ and let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be n realizations of the response $y \in \mathcal{M}$. Notice that the realizations are assumed to be measured with additive noise, which might pull the observations to an ambient space of $\mathcal{M}$. An example of such additive noise for landmark data is given in Section 3.5. Denote for each observation $i = 1, \ldots, n$, $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{im}) \in \mathbb{R}^m$ the covariate vector of $m \leq d$ covariate variables. The SDR model is defined as a stochastic process on $\mathcal{M}$ based on the definition of Euclidean regression models regarded as stochastic processes (see (3.1)). Assume therefore that the response $y \in \mathcal{M}$ is the endpoint of a stochastic process y_t in $\mathcal{M}$ and the covariates, $\boldsymbol{x}_i$, the endpoint of a stochastic process $X_t = (X_{1t}, \ldots, X_{mt})$ in $\mathbb{R}^m$. The process X_{jt} is for random covariate variables assumed to be a Brownian motion in $\mathbb{R}$, while for fixed covariate effects it is modelled as a fixed drift. The process y_{it} for each observation $i = 1, \ldots, n$ is defined as the stochastic development of a Euclidean model on $\mathbb{R}^m$. Consider the stochastic process, z_{it}, in $\mathbb{R}^m$ defined by the stochastic differential equation equivalent to the Euclidean regression model defined in (3.1),

$$dz_{it} = \alpha dt + W dX_{it} + d\varepsilon_{it}, \quad t \in [0, 1]. \tag{3.2}$$

Here αdt is a fixed drift, W the $m \times m$ coefficient matrix and ε_{it} the random error modelled as a Brownian motion in $\mathbb{R}^m$. The response process y_{it} is then given as the stochastic development of z_{it}, i.e. $y_{it} = \varphi_{(\boldsymbol{y}_0,\nu_0)}(z_{it})$ for a reference point $\boldsymbol{y}_0$ and frame $\nu_0 \in T_{\boldsymbol{y}_0}\mathcal{M}$ (see Fig. 3.1). The realizations are modelled as noisy observations of the endpoints of y_{it}, $\boldsymbol{y}_i = y_{i1} + \tilde{\boldsymbol{\varepsilon}}_i$ in which $\tilde{\boldsymbol{\varepsilon}}_i \sim \mathcal{N}(0, \tau^2 I)$ denotes iid. additive noise. There is a natural relation between W and the frame ν_0. If ν_0 is assumed to be an orthonormal basis and U the $d \times m$-matrix with columns of basis vectors of ν_0, then the matrix $\tilde{W} = UW$ explains the gathered effect of W and ν_0 through U. However, this decomposition is not unique and hence the $\tilde{W}$ matrix is estimated instead of U and W individually.

3.4 Method of Moments

In this section the MM procedure is introduced for the estimation of the parameters in the regression model. The MM procedure uses known moment conditions to define a set of equations which can be optimized to find the true parameter vector $\theta = (\tau, \alpha, \tilde{W}, \boldsymbol{y}_0)$, see [97, 45, 29]. Here τ^2 is the additive noise variance, α the drift, $\tilde{W}$ combined effect of covariates and ν_0, and $\boldsymbol{y}_0$ the initial point on $\mathcal{M}$.

In the SDR model the known moment conditions are based on the moments of the additive noise $\tilde{\boldsymbol{\varepsilon}}_i$ and the fact that $\tilde{\boldsymbol{\varepsilon}}_i$ is independent of the covariate variables x_{ik} for each $k = 1, \ldots, m$. Hence, the moment conditions are,

$$\mathbb{E}\left[\tilde{\varepsilon}_{ij}\right] = 0,\ \mathbb{E}\left[\tilde{\varepsilon}_{ij} x_{ik}\right] = 0,\ \mathbb{E}\left[\tilde{\varepsilon}_{ij}^2\right] = \tau^2\ \forall j = 1, \ldots, d, \text{ and } k = 1, \ldots, m.$$

Known consistent estimators for these moments are the sample means. Consider the residuals, $\hat{\varepsilon}_{ij} = y_{ij} - \hat{y}_{ij}$, in which the dependency of the parameter vector, θ, lies in the predictions, $\hat{y}_{ij}$ for $i = 1, \ldots, n,\ j = 1, \ldots, d$. For a proper choice of parameter vector θ, the sample means will approach the true moments. Therefore, the set of equations used to optimize the parameter vector θ are,

$$\frac{1}{n}\sum_{i=1}^{n} \hat{\varepsilon}_{ij} = 0, \quad \frac{1}{n}\sum_{i=1}^{n} x_{ik}\hat{\varepsilon}_{ij} = 0, \quad \text{and} \quad \frac{1}{n-2}\sum_{i=1}^{n} \hat{\varepsilon}_{ij}^2 = \hat{\tau}^2,$$

for all $j = 1, \ldots, d$ and $k = 1, \ldots, m$ and where $\hat{\tau}^2$ is the estimated variance. In Euclidean statistics, the method of moments is known to provide consistent estimators, but these estimators might be biased.

The cost function considered for optimization with respect to θ is,

$$f(\theta) = \frac{1}{d}\sum_j \left(\frac{1}{n}\sum_{i=1}^n \hat{\varepsilon}_{ij}\right)^2 + \frac{1}{dm}\sum_{j,k}\left(\frac{1}{n}\sum_{i=1}^n x_{ik}\hat{\varepsilon}_{ij}\right)^2 + \frac{1}{d}\sum_j \left(\frac{1}{n-2}\sum_{i=1}^n \hat{\varepsilon}_{ij}^2 - \hat{\tau}^2\right)^2. \quad (3.3)$$

This cost function depends on predictions from the model based on the given parameter vector in each iteration. In order for the objective function to be stable it has to be evaluated for several predictions. Therefore, the function has been averaged for several predictions to obtain a more stable gradient descent optimization procedure.

The initial value of θ can in practice be chosen as parameters estimated from a Euclidean multivariate linear regression model. Here, the estimated covariance matrix would resemble the $\tilde{W}$ effect and the intercept the initial point $\boldsymbol{y}_0$.

3.5 Simulation Example

The performance of the estimation procedure will be evaluated using simulated data. We will generate landmark data on Riemannian landmark manifolds as defined in the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework [135], and use the Levi-Civita connection. Shapes in the landmark manifold $\mathcal{M}$ are defined by a finite landmark representation, i.e. $q \in \mathcal{M}$, $q = (x_1^1, x_1^2, \ldots, x_{n_l}^1, x_{n_l}^2)$, where n_l denotes the number of landmarks. The dimension of $\mathcal{M}$ is hence $d = 2n_l$. Using a kernel K, the Riemannian metric on $\mathcal{M}$ is defined as $g(v,w) = \sum_{i,j}^{n_l} vK^{-1}(\boldsymbol{x}_i, \boldsymbol{x}_j)w$ with K^{-1} denoting the inverse of the kernel matrix. In the following, we use a Gaussian kernel for K with standard deviation $\sigma = 0.1$.

We will consider a single covariate variable $x \in \mathbb{R}$ drawn from $\mathcal{N}(0, 36)$ and model the relation to two response variables either with 1 or 3 landmarks. The response variables are simulated from a model with parameters given in Table 3.1 and Fig. 3.2 for $n_l = 3$. Examples of simulated data for $n_l = 1$ and 3 are shown in Fig. 3.2. The additive noise is in this case normally distributed iid. random noise added to each coordinate of landmarks. In this example we consider a simplification of the model, as the random error in z_{it}, given in (3.2), will be disregarded. Estimation of parameters is examined for three different models: one without additive noise and drift, one without drift, and at last the full model. For $n_l = 3$ only estimation of the two first models is studied, and estimation in the model with no drift has been considered for $n = 70$ and $n = 150$.

By the results shown in Table 3.1 and Fig. 3.2, the procedure makes a good estimate of the frame matrix $\tilde{W}$ in every situation. For the model with no

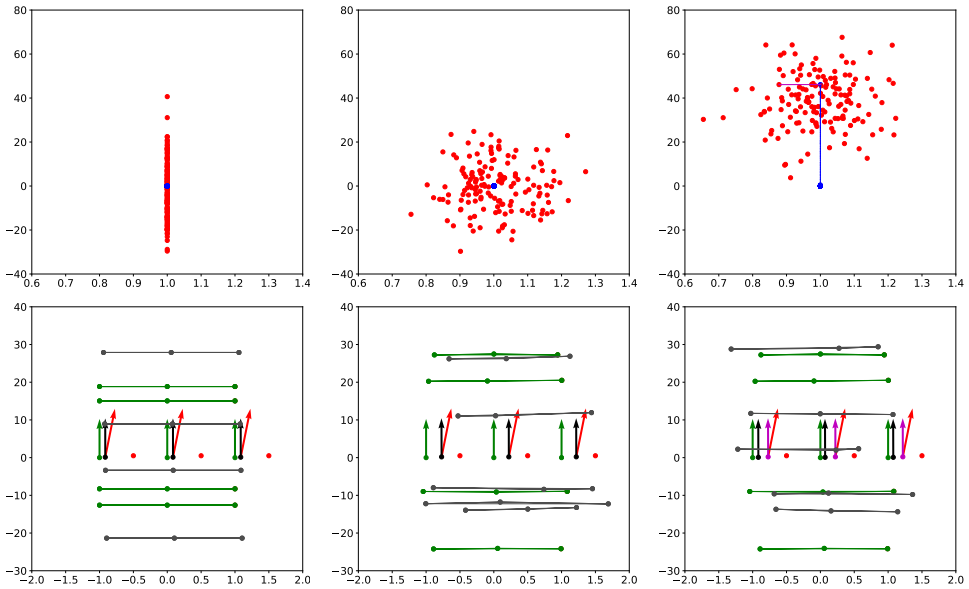


Figure 3.2: (upper left) Sample drawn from model without additive noise and drift. (upper center) Sample drawn with additive noise, but no drift. (upper right) Sample drawn from the full model. The vertical lines are the stochastic development of z_{it} and the horizontal corresponds to the additive noise, the blue point is the reference point. (lower left) Model without drift and variance for $n_l = 3$, $n = 70$. (lower center) Model without drift and $n = 70$. (lower right) Model without drift and $n = 150$. These plots show the estimated results. (red) initial, (green) true, and (black) estimated reference point and frame. The gray samples are predicted from the estimated model while the green are a subset of the simulated data. Lower right plot does also show the difference in the estimated parameters for $n = 70$, $n = 150$ for the model with no drift. The magenta parameters in that plot is the estimated parameters for model without drift and $n = 70$, the corresponding black parameters in lower center plot.

additive noise and no drift, the procedure finds a reasonable estimate of $\boldsymbol{y}_0$. When noise is added, it is seen that a larger sample size is needed in order to get a good estimate of $\boldsymbol{y}_0$. On the contrary, the variance estimate seems biased in each case. For $n_l = 3$ the variance parameters estimated were $\hat{\tau} = 0.306$ for $n = 70$ and $\hat{\tau} = 0.231$ for $n = 150$. However, when drift is added to the model, the estimation procedure has a hard time recapture the true estimates of $\boldsymbol{y}_0$ and α. This difficulty can be explained by the relation between the variables. In normal linear regression, only one intercept variable is present in the model, but in the SDR this intercept variable is split between α and $\boldsymbol{y}_0$.

Table 3.1: Parameter estimates found with the MM procedure for 1 landmark. First column shows the true values and each column, estimated parameters in each model.

	True	excl. τ, α $n = 70$	excl. α $n = 70$	excl. α $n = 150$	full model $n = 150$
τ	0.1	- ($\tau = 0$)	0.256	0.226	0.207
α	40	- ($\alpha = 0$)	- ($\alpha = 0$)	- ($\alpha = 0$)	37.19
$\tilde{W}$	$(0, 2)$	$(0, 2.013)$	$(0.004, 1.996)$	$(0, 2.003)$	$(0, 2.004)$
$\boldsymbol{y}_0$	$(1, 0)$	$(1.064, 0.0438)$	$(1.158, 0.162)$	$(1.026, 0.0227)$	$(1.076, 2.708)$

3.6 Conclusion

Method of Moments procedure has been examined for parameter estimation in the stochastic development regression (SDR) model. The SDR model is a generalization of regression models on Euclidean space to manifold-valued data. This model analyzes the relation between manifold-valued response and Euclidean covariate variables. The performance of the estimation procedure was studied based on a simulation example. The Method of Moments procedure was easier to apply and less computationally expensive than the Laplace approximation considered in [66]. The estimates found for the frame parameters were reasonable, but the procedure had a hard time retrieving the reference point and drift parameter. This is due to a mis-specification of the model as the reference point and drift parameter jointly correspond to the intercept in normal Euclidean regression models and hence there is no unique split of these parameters.

For further investigation, it could be interesting to test the relation between the reference point and drift parameter to be able to retrieve good estimates of these parameters. In the Euclidean case, the Method of Moments procedure has been shown to provide consistent, but sometimes biased estimates. An interesting question for future work could also be, whether the parameter estimates in this model is consistent and biased.

Acknowledgements. This work was supported by the CSGB Centre for Stochastic Geometry and Advanced Bioimaging funded by a grant from the Villum foundation.

3.7 Example on S^2 and Further Thoughts

In this section, we present an example of the stochastic development regression (SDR) for data on S^2 and make suggestions for further improvement of the model.

The paper presented in this chapter discussed mis-specifications of the SDR model as different parameters described similar effects, e.g. the drift of the Euclidean semi-martingale, z_t, together with the initial point y_0, and the covariate weight matrix W with the initial frame ν_0. To get rid of the mis-specifications, this example consider a simplified semi-martingale,

$$dz_{it} = W dX_{it}. \tag{3.4}$$

The drift α and intrinsic noise ε are disregarded, resulting in the process z_{it} exclusively describing the covariate dependency on y. The response y is modelled as the stochastic development, ϕ, of z_t,

$$y_i = \phi_{(y_0,\nu_0)}(z_{it})\big|_{t=1} + \tilde{\boldsymbol{\varepsilon}}_i, \tag{3.5}$$

for the initial point $y_0 \in \mathcal{M}$, initial frame ν_0 of $T_{y_0}\mathcal{M}$ and additive noise $\tilde{\boldsymbol{\varepsilon}}_i \sim N(0, \tau^2\mathbb{I})$. To remove the problem of separating the effects of ν_0 and W, we model ν_0 as the orthonormal basis obtained from a Cholesky decomposition of the coordinate matrix of the metric g evaluated at y_0.

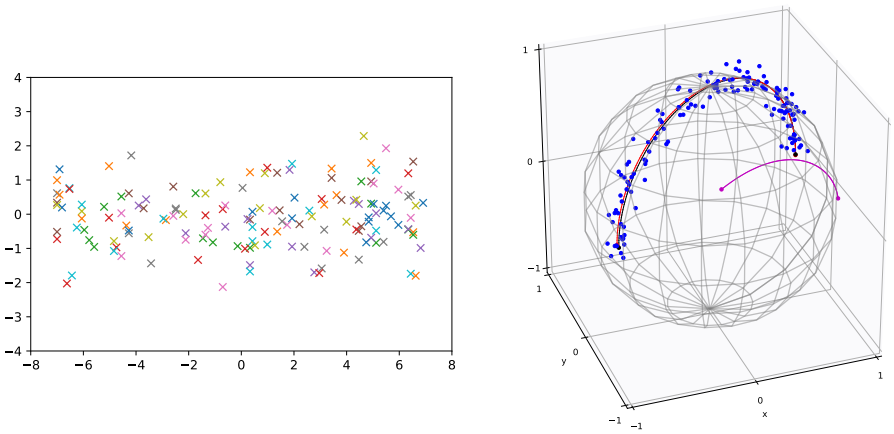


Figure 3.3: Example of stochastic development regression on S^2. (left) Simulated covariate variables. (right) Corresponding response variables. The red regression curve resembles the estimated model, the black curve (near the red) is the true model and the magenta curve is the initial regression curve.

A data sample of 150 observations was simulated on S^2 based on the two-dimensional covariate observations visualised in the left plot of Fig. 3.3. The

corresponding simulated response observations are shown in the right plot of Fig. 3.3 together with the initial (magenta), estimated (red), and true regression curve (black). The additive noise pulls the response observations to the ambient space $\mathbb{R}^3$. When excluding the drift and intrinsic noise ε, thewe see that the model is able to retrieve a good estimate of the regression curve on S^2 (see Fig. 3.3).

Notice that we exclusively included the extrinsic noise in the model to obtain a less stochastic objective function for the method of moments. An alternative optimisation procedure could be to define method of moments for the stochastic development process $\phi_{(y_0,\nu_0)}(z_t)$ instead of the extrinsic noise $\tilde{\varepsilon}$. This optimisation procedure was used in the paper presented in Chapter 5. The procedure determines the moments of a stochastic process, e.g. $\phi_{(y_0,\nu_0)}(z_t)$, by applying the Fokker-Planck equations. The obtained moments for the limit distribution of the stochastic process is then matched with the observed data moments. The method does not require model predictions and will hence be computationally more efficient and result in a stable optimisation procedure. Moreover, the procedure makes it possible to include intrinsic uncertainty on the data space via the Euclidean process z_t. Introducing the Fokker-Planck based method of moments to the stochastic development regression will be a focal point for future work.

CHAPTER 4

Most Likely Separation of Intensity and Warping Effects in Image Registration

The chapter contains the paper

- L. Kühnel, S. Sommer, A. Pai, and L. L. Raket. Most likely separation of intensity and warping effects in image registration. *SIAM Journal on Imaging Sciences*, 10(2):578–601, 2017

published in SIAM Journal on Imaging Sciences. The work is conducted in collaboration with Stefan Sommer, Akshay Pai, and Lars Lau Raket. We present a model for separating uncertainty in image data. Compared to the deformation model presented in Chapter 5, the considered deformations in this chapter is not required to be diffeomorphic, but are modelled as random displacement fields on a discretised lattice.

Most Likely Separation of Intensity and Warping Effects in Image Registration

Line Kühnel[1], Stefan Sommer[1], Akshay Pai[1], and Lars Lau Raket[2]

Department of Computer Science, University of Copenhagen, Denmark[1]

Department of Mathematical Sciences, University of Copenhagen, Denmark[2]

Abstract

This paper introduces a class of mixed-effects models for joint modeling of spatially correlated intensity variation and warping variation in 2D images. Spatially correlated intensity variation and warp variation are modeled as random effects, resulting in a nonlinear mixed-effects model that enables simultaneous estimation of template and model parameters by optimization of the likelihood function. We propose an algorithm for fitting the model which alternates estimation of variance parameters and image registration. This approach avoids the potential estimation bias in the template estimate that arises when treating registration as a preprocessing step. We apply the model to datasets of facial images and 2D brain magnetic resonance images to illustrate the simultaneous estimation and prediction of intensity and warp effects.

Keywords: template estimation, image registration, separation of phase and intensity variation, nonlinear mixed-effects model

4.1 Introduction

When analyzing collections of imaging data, a general goal is to quantify similarities and differences across images. In medical image analysis and computational anatomy, a common goal is to find patterns that can distinguish morphologies of healthy and diseased subjects aiding the understanding of the population epidemiology. Such distinguishing patterns are typically investigated by comparing single observations to a representative member of the underlying population, and statistical analyses are performed relative to this representation. In the context of medical imaging, it has been customary to choose the template from the observed data as a common image of the population. However, such an approach has been shown to be highly

dependent on the choice of the image. In more recent approaches, the templates are estimated using statistical methods that make use of the additional information provided by the observed data [81].

In order to quantify the differences between images, the dominant modes of variation in the data must be identified. Two major types of variability in a collection of comparable images are *intensity variation* and variation in *point-correspondences*. Point-correspondence or *warp* variation can be viewed as shape variability of an individual observation with respect to the template. Intensity variation is the variation that is left when the observations are compensated for the true warp variation. This typically includes noise artifacts like systematic error and sensor noise or anatomical variation such as tissue density or tissue texture. Typically one would assume that the intensity variation consists of both independent noise and spatially correlated effects.

In this work, we introduce a flexible class of mixed-effects models that explicitly model the template as a fixed effect and intensity and warping variation as random effects, see Figure 4.1. This simultaneous approach enables separation of the random variation effects in a data-driven fashion using alternating maximum-likelihood estimation and prediction. The resulting model will therefore choose the separation of intensity and warping effects that is most likely given the patterns of variation found in the data. From the model specification and estimates, we are able to denoise observations through linear prediction in the model under the maximum likelihood estimates. Estimation in the model is performed with successive linearization around the warp parameters enabling the use of linear mixed-effects predictors and avoiding the use of sampling techniques to account for nonlinear terms. We apply our method on datasets of face images and 2D brain MRIs to illustrate its ability to estimate templates for populations and predict warp and intensity effects.

4.1.1 Outline of the paper

The paper is structured as follows. In Section 4.2, we give an overview of previously introduced methods for analyzing image data with warp variation. Section 4.3 covers the mixed-effects model including a description of the estimation procedure (Section 4.3.1) and how to predict from the model (Section 4.3.2). In Section 4.4, we give an example of how to model spatially correlated variations with a tied-down Brownian sheet. We consider two applications of the mixed-effects model to real-life datasets in Section 4.5 and Section 4.6 contains a simulation study that is used for comparing the precision of the model to more conventional approaches.

Figure 4.1: Fixed and random effects: The template (θ: leftmost) pertubed by random warp ($\theta \circ v$: 2nd from left) and warp+spatially correlated intensity ($\theta \circ v + x$: 3rd from left) together with independent noise ϵ constitute the observation (y: 4th from left). Right: the warp field v that brings the observation into spatial correspondence with θ overlayed the template. Estimation of template and model hyperparameters are conducted simultaneously with prediction of random effects allowing separation of the different factors in the nonlinear model.

4.2 Background

The model introduced in this paper focuses on separately modelling the intensity and warp variation. Image registration conventionally only focuses on identifying warp differences between pairs of images. The intensity variation is not included in the model and possible removal of this effect is considered as a pre-or postprocessing step. The warp differences are often found by solving a variational problem of the form

$$E_{I_1,I_2}(\varphi) = R(\varphi) + \lambda S(I_1, I_2 \circ \varphi^{-1}), \tag{4.1}$$

see for example [123]. Here S measures the dissimilarity between the fixed image I_1 and the warped image $I_2 \circ \varphi^{-1}$, R is a regularization on the warp φ, and $\lambda > 0$ is a weight that is often chosen by ad-hoc methods. After registration, either the warp, captured in φ, or the intensity differences between I_1 and $I_2 \circ \varphi^{-1}$ can be analyzed [126]. Several works have defined methods that incorporate registration as part of the defined models. The approach described in this paper will also regard registration as a part of the proposed model and adress the following three problems that arise in image analysis: (a) being able to estimate model parameters such as λ in a data-driven fashion; (b) assuming a generative statistical model that gives explicit interpretation of the terms that corresponds to the dissimilarity S and penalization R; and (c) being simultaneous in the estimation of population-wide effects such as the mean or template image and individual per-image effects, such as the warp and intensity effects. These features are of fundamental importance in image registration and many works have addressed combinations of them. The main difference of our approach to state-of-the-art statistical registration

frameworks is that we propose a simultaneous random model for warp and intensity variation. As we will see, the combination of maximum likelihood estimation and the simultaneous random model for warp and intensity variation manifests itself in a trade-off where the uncertainty of both effects are taken into account simultaneously. As a result, when estimating fixed effects and predicting random effects in the model the most likely separation of the effects given the observed patterns of variation in the entire data material is used.

Methods for analyzing collections of image data, for example template estimation in medical imaging [57], with both intensity and warping effects can be divided into two categories, *two-step methods* and *simultaneous methods*. Two-step methods perform alignment as a preprocessing step before analyzing the aligned data. Such methods can be problematic because the data is modified and the uncertainty related to the modification is ignored in the subsequent analysis. This means that the effect of intensity variation is generally underestimated, which can introduce bias in the analysis, see [108] for the corresponding situation in 1D functional data analysis. Simultaneous methods, on the other hand, seek to analyze the images in a single step that includes the alignment procedure.

Conventional simultaneous methods typically use L^2 data terms to measure dissimilarity. Such dissimilarity measures are equivalent to the model assumption that the intensity variation in the image data consists solely of uncorrelated Gaussian noise. This approach is commonly used in image registration with the sum of squared differences (SSD) dissimilarity measure, and in atlas estimation [141]. Since the L^2 data term is very fragile to systematic deviations from the model assumption, for example contrast differences, the method can perform poorly. One solution to make the L^2 data term more robust against systematic intensity variation and in general to insufficient information in the data term is to add a strong penalty on the variation of the warping functions. This approach is however an implicit solution to the problem, since the gained robustness is a side effect of regularizing another model component. As a consequence, the effect on the estimates is very hard to quantify, and it is very hard to specify a suitable regularization for a specific type of intensity variation. This approach is, for example, taken in the variational formulations of the template estimation problem in [57]. An elegant instance of this strategy is the Bayesian model presented in [2] where the warping functions are modeled as latent Gaussian effects with an unknown covariance that is estimated in a data-driven fashion. Conversely, systematic intensity variation can be sought to be removed prior to the analysis, in a reversed two-step method, for example by using bias-correction techniques for MRI data [131]. The presence of warp variation can however influence the estimation of the intensity effects.

Analysis of images with systematic intensity differences can be improved using data dissimilarity measures that are robust or invariant to such system-

atic differences. However, robustness and invariance come at a cost in accuracy. By choosing a specific kind of invariance in the dissimilarity measure, the model is given a pre-specified recipe for separating intensity and warping effects; the warps should maximize the invariant part of the residual under the given model parameters. Examples of classical robust data terms include L^1-norm data terms [105], Charbonnier data terms [20], and Lorentzian data terms [17]. Robust data terms are often challenging to use, since they may not be differentiable (L^1-norms) or may not be convex (Lorentzian data term). A wide variety of invariant data terms have been proposed, and are useful when the invariances represent a dominant mode of variation in the data. Examples of classical data terms that are invariant to various linear and nonlinear photometric relationships are normalized cross-correlation, correlation-ratio and mutual information [82, 47, 110, 98]. Another approach for achieving robust or invariant data terms is to transform the data that is used in the data term. A classical idea is to match discretely computed gradients or other discretized derivative quantities [99]. A related idea is to construct invariant data terms based on discrete transformations. This type of approach has become increasingly popular in image matching in recent years. Examples include the rank transform and the census transform [138, 88, 43, 44], and more recently the complete rank transform [33]. While both robust and invariant data terms have been shown to give very good results in a wide array of applications, they induce a fixed measure of variation that does not directly model variation in the data. Thus, the general applicability of the method can come at the price of limited accuracy.

Several alternative approaches for analyzing warp and intensity simultaneously have been proposed [91, 56, 18, 134]. In [91] warps between images are considered as combination of two transformation fields, one representing the image motion (warp effect) and one describing the change of image brightness (intensity effect). Based on this definition warp and intensity variation can be modeled simultaneously. An alternative approach is considered in [56], where an invariant metric is used, which enables analysis of the dissimilarity in point correspondences between images disregarding the intensity variation. These methods are not statistical in the sense that they do not seek to model the random structures of the variation of the image data. A statistical model is presented in [18], where parameters for texture, shape variation (warp) and rendering are estimated using maximizing-a-posteriori estimation.

To overcome the mentioned limitations of conventional approaches, we propose to do statistical modeling of the sources of variation in data. By using a statistical model where we assume parametric covariance structures for the different types of observed variation, the variance parameters can be estimated from the data. The contribution of different types of variation is thus weighted differently in the data term. By using, for example, maximum-likelihood estimation, the most likely form of the variation given the data is

penalized the least. We emphasize that in contrast to previous mixed-effects models incorporating warp effects [2, 141], the goal here is to simultaneously model warp and intensity effects. These effects impose randomness relative to a template, the fixed-effect, that is estimated during the inference process.

The nonlinear mixed-effects models are a commonly used tool in statistics. These types of models can be computationally intensive to fit, and are rarely used for analyzing large data sizes such as image data. We formulate the proposed model as a nonlinear mixed-effects model and demonstrate how certain model choices can be used to make estimation in the model computationally feasible for large data sizes. The model incorporates random intensity and warping effects in a small-deformation setting: We do not require warping functions to produce diffeomorphisms. The geometric structure is therefore more straightforward than in for example the LDDMM model [135]. From a statistical perspective, the small-deformation setting is much easier to handle than the large-deformation setting where warping functions are restricted to produce diffeomorphisms.

Instead of requiring diffeomorphisms, we propose a class of models that will produce warping functions that in most cases do not fold. Another advantage of the small-deformation setting is that we can model the warping effects as latent Gaussian disparity vectors in the domain. Such direct modeling allows one to compute a high-quality approximation of the likelihood function by linearizing the model around the modes of the nonlinear latent random variables. The linearized model can be handled using conventional methods for linear mixed-effects models [103] which are very efficient compared to sampling-based estimation procedures.

In the large-deformation setting, the metamorphosis model [129, 130] extends the LDDMM framework for image registration [135] to include intensity change in images. Warp and intensity differences are modeled separately in metamorphosis with a Riemannian structure measuring infinitesimal variation in both warp and intensity. While this separation has similarities to the statistical model presented here, we are not aware of any work which have considered likelihood-based estimation of variables in metamorphosis models.

4.3 Statistical model

We consider spatial functional data defined on $\mathbb{R}^2$ taking values in $\mathbb{R}$. Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be n functional observations on a regular lattice with $m = m_1 m_2$ points (s_j, t_k), that is, $\boldsymbol{y}_i = (y_i(s_j, t_k))_{j,k}$ for $j = 1, \ldots, m_1$, $k = 1, \ldots, m_2$. Consider the model in the image space

$$y_i(s_j, t_k) = \theta(v_i(s_j, t_k)) + x_i(s_j, t_k) + \varepsilon_{ijk}, \tag{4.2}$$

for $i = 1, \ldots, n$, $j = 1, \ldots, m_1$ and $k = 1, \ldots, m_2$. Here $\theta \colon \mathbb{R}^2 \to \mathbb{R}$ denotes the template and $v_i \colon \mathbb{R}^2 \to \mathbb{R}^2$ is a warping function matching a point in y to a point in the template θ. Moreover x_i is the random spatially correlated intensity variation for which we assume that $\boldsymbol{x}_i = (x_i(s_j, t_k))_{j,k} \sim \mathcal{N}(0, \sigma^2 S)$ where the spatial correlation is determined by the covariance matrix S. The term $\varepsilon_{ijk} \sim \mathcal{N}(0, \sigma^2)$ models independent noise. The template θ is a fixed-effect while v_i, x_i, and ε_{ijk} are random.

We will consider warping functions of the form

$$v_i(s,t) = v(s,t,\boldsymbol{w}_i) = \begin{pmatrix} s \\ t \end{pmatrix} + \mathcal{E}_{\boldsymbol{w}_i}(s,t),$$

where $\mathcal{E}_{\boldsymbol{w}_i} \colon \mathbb{R}^2 \to \mathbb{R}^2$ is coordinate-wise bilinear spline interpolation of $\boldsymbol{w}_i \in \mathbb{R}^{m_w^1 \times m_w^2 \times 2}$ on a lattice spanned by $\boldsymbol{s}_w \in \mathbb{R}^{m_w^1}, \boldsymbol{t}_w \in \mathbb{R}^{m_w^2}$. In other words, $\boldsymbol{w}_i$ models discrete spatial displacements at the lattice anchor points. Figure 4.2 shows an example of disparity vectors on a grid of anchor points and the corresponding warping function.

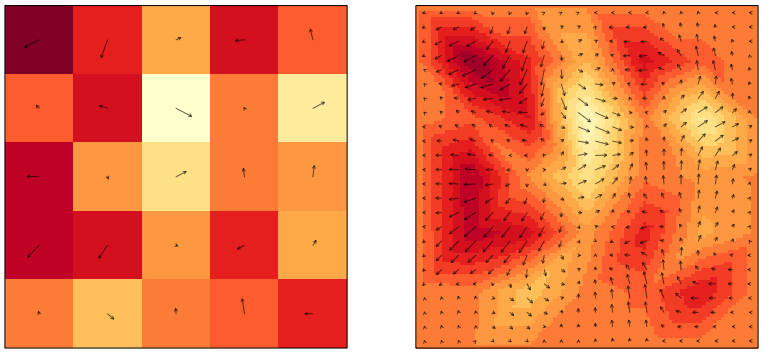


Figure 4.2: An example of disparity vectors at a 5×5 grid of anchor points and the corresponding warping function.

The displacements are modeled as random effects, $\boldsymbol{w}_i \sim \mathcal{N}(0, \sigma^2 C)$ where C is a $2m_w^1 m_w^2 \times 2m_w^1 m_w^2$ covariance matrix, and, as a result, the warping functions can be considered nonlinear functional random effects. As $\boldsymbol{w}_i$ is assumed to be normally distributed with mean zero, small displacements are favorited and hence the warp effect will be less prone to fold. The model is a spatial extension of the phase and amplitude varying population pattern (pavpop) model for curves [108, 106].

4.3.1 Estimation

First, we will consider estimation of the template θ from the functional observations, and we will estimate the contributions of the different sources

of variation. In the proposed model, this is equivalent to estimating the covariance structure C for the warping parameters, the covariance structure S for the spatially correlated intensity variation, and the noise variance σ^2. The estimate of the template is found by considering model (4.2) in the back-warped template space

$$y_i(v_i^{-1}(s_j, t_k)) = \theta(s_j, t_k) + x_i(v_i^{-1}(s_j, t_k)) + \tilde{\varepsilon}_{ijk}. \tag{4.3}$$

Because every back-warped image represents θ on the observation lattice, a computationally attractive parametrization is to model θ using one parameter per observation point, and evaluate non-observation points using bilinear interpolation. This parametrization is attractive, because Henderson's mixed-model equations [46, 109] suggests that the conditional estimate for $\theta(s_j, t_k)$ given $\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n$ is the pointwise average

$$\hat{\theta}(s_j, t_k) = \frac{1}{n} \sum_{i=1}^{n} y_i(v_i^{-1}(s_j, t_k)), \tag{4.4}$$

if we ignore the slight change in covariance resulting from the back-warping of the random intensity effects. As this estimator depends on the warping parameters, the estimation of θ and the variance parameters has to be performed simultaneously with the prediction of the warping parameters. We note that, as in any linear model, the estimate of the template is generally quite robust against slight misspecifications of the covariance structure. And the idea of estimating the template conditional on the posterior warp is similar to the idea of using a hard EM algorithm for computing the maximum likelihood estimator for θ [90].

We use maximum-likelihood estimation to estimate variance parameters, that is, we need to minimize the negative log-likelihood function of model (4.2). Note that (4.2) contains nonlinear random effects due to the term $\theta(v_i(s, t, \boldsymbol{w}_i))$ where $\theta \circ v_i$ is a nonlinear transformation of $\boldsymbol{w}_i$. We handle the nonlinearity and approximate the likelihood by linearizing the model (4.2) around the current predictions $\boldsymbol{w}_i^0$ of the warping parameters $\boldsymbol{w}_i$:

$$\begin{aligned} y_i(s_j, t_k) &\approx \theta(v(s_j, t_k, \boldsymbol{w}_i^0)) \\ &\quad + (\nabla\theta(v(s_j, t_k, \boldsymbol{w}_i^0)))^\top J_{\boldsymbol{w}_i} v(s_j, t_k, \boldsymbol{w}_i)\Big|_{\boldsymbol{w}_i = \boldsymbol{w}_i^0} (\boldsymbol{w}_i - \boldsymbol{w}_i^0) \\ &\quad + x_i(s_j, t_k) + \varepsilon_{ijk} \\ &= \theta(v(s_j, t_k, \boldsymbol{w}_i^0)) + Z_{ijk}(\boldsymbol{w}_i - \boldsymbol{w}_i^0) + x_i(s_j, t_k) + \varepsilon_{ijk}, \end{aligned} \tag{4.5}$$

where $J_{\boldsymbol{w}_i} v(s_j, t_k, \boldsymbol{w}_i)$ denotes the Jacobian matrix of v with respect to $\boldsymbol{w}_i$ and

$$Z_{ijk} = (\nabla\theta(v(s_j, t_k, \boldsymbol{w}_i^0)))^\top J_{\boldsymbol{w}_i} v(s_j, t_k, \boldsymbol{w}_i)\Big|_{\boldsymbol{w}_i = \boldsymbol{w}_i^0}. \tag{4.6}$$

Letting $Z_i = (Z_{ijk})_{jk} \in \mathbb{R}^{m \times 2m_w^1 m_w^2}$, the linearized model can be rewritten

$$\boldsymbol{y}_i \approx \boldsymbol{\theta}^{\boldsymbol{w}_i^0} + Z_i(\boldsymbol{w}_i - \boldsymbol{w}_i^0) + \boldsymbol{x}_i + \boldsymbol{\varepsilon_i}. \tag{4.7}$$

We notice that in this manner, $\boldsymbol{y}_i$ can be approximated as a linear combination of normally distributed variables, hence the negative log-likelihood for the linearized model is given by

$$\begin{aligned}\ell_{\boldsymbol{y}}(\theta, C, \sigma^2) =& \frac{nm_1m_2}{2}\log\sigma^2 + \frac{1}{2}\sum_{i=1}^{n}\log\det V_i \\ &+ \frac{1}{2\sigma^2}\sum_{i=1}^{n}(\boldsymbol{y}_i - \boldsymbol{\theta}^{\boldsymbol{w}_i^0} + Z_i\boldsymbol{w}_i^0)^\top V_i^{-1}(\boldsymbol{y}_i - \boldsymbol{\theta}^{\boldsymbol{w}_i^0} + Z_i\boldsymbol{w}_i^0),\end{aligned} \tag{4.8}$$

where $V_i = Z_iCZ_i^\top + S + \mathbb{I}_m$. The idea of linearizing nonlinear mixed-effects models in the nonlinear random effects is a solution that has been shown to be effective and which is implemented in standard software packages [79, 103, 104]. The proposed model is, however, both more general and computationally demanding than what can be handled by conventional software packages. Furthermore, we note that the linearization in a random effect as done in model (4.7) is fundamentally different than the conventional linearization of a nonlinear dissimilarity measure such as in the variational problem (4.1). As we see from the linearized model (4.7), the density of $\theta(v(s_j, t_k, \boldsymbol{w}_i)$ is approximated by the density of a linear combination, $\theta(v(s_j, t_k, \boldsymbol{w}_i^0)) + Z_{ijk}(\boldsymbol{w}_i - \boldsymbol{w}_i^0)$, of multivariate Gaussian variables. The likelihood function for the first-order Taylor expansion in $\boldsymbol{w}_i$ of the model (4.2) is thus a Laplace approximation of the true likelihood, and the quality of this approximation is approximately second order [133].

Computing the likelihood function

As mentioned above the proposed model is computationally demanding. Even the approximated likelihood function given in equation (4.8) is not directly computable because of the large data sizes. In particular, the computations related to determinants and inverses of the covariance matrix V_i are infeasible unless we impose certain structures on these. In the following, we will assume that the covariance matrix for the spatially correlated intensity variation S has full rank and sparse inverse. We stress that this assumption is merely made for computational convenience and that the proposed methodology is also valid for non-sparse precision matrices. The zeros in the precision matrix S^{-1} are equivalent to assuming conditional independences between the intensity variation in corresponding pixels given all other pixels [73]. A variety of classical models have this structure, in particular (higher-order) Gaussian Markov random fields models have sparse precision matrices because of their Markov property.

To efficiently do computations with the covariances $V_i = Z_i C Z_i^\top + S + \mathbb{I}_m$, we exploit the structure of the matrix. The first term $Z_i C Z_i^\top$ is an update to the intensity covariance $S + \mathbb{I}_m$ with a maximal rank of $2m_w^1 m_w^2$. Furthermore, the first term of the intensity covariance S has a sparse inverse and the second term $\mathbb{I}_m$ is of course sparse with a sparse inverse. Using the Woodbury matrix identity, we obtain

$$\begin{aligned} V_i^{-1} &= (Z_i C Z_i^\top + S + \mathbb{I}_m)^{-1} \\ &= (S + \mathbb{I}_m)^{-1} - (S + \mathbb{I}_m)^{-1} Z_i (C^{-1} + Z_i^\top (S + \mathbb{I}_m)^{-1} Z_i)^{-1} Z_i^\top (S + \mathbb{I}_m)^{-1} \end{aligned}$$

which can be computed if we can efficiently compute the inverse of the potentially huge $m \times m$ intensity covariance matrix $(S + \mathbb{I}_m)^{-1}$. We can rewrite the inverse intensity covariance as

$$(S + \mathbb{I}_m)^{-1} = \mathbb{I}_m - (\mathbb{I}_m + S^{-1})^{-1}.$$

Thus we can write V_i^{-1} in a way that only involves operations on sparse matrices. To compute the inner product $\boldsymbol{y}^\top V_i^{-1} \boldsymbol{y}$, we first form the matrix $\mathbb{I}_m + S^{-1}$ and compute its Cholesky decomposition using the Ng-Peyton method [92] implemented in the `spam` R-package [41]. By solving a low-rank linear system using the Cholesky decomposition, we can thus compute $L = (C^{-1} + Z_i^\top (S + \mathbb{I}_m)^{-1} Z_i)^{-1}$. The inner product is then efficiently computed as

$$\boldsymbol{y}^\top V_i^{-1} \boldsymbol{y} = \boldsymbol{y}^\top \boldsymbol{x} - (Z_i \boldsymbol{x})^\top L Z_i \boldsymbol{x}$$

where

$$\boldsymbol{x} = (S + \mathbb{I}_m)^{-1} \boldsymbol{y}.$$

To compute the log determinant in the likelihood, one can use the matrix determinant lemma similarly to what was done above to split the computations into low-rank computations and computing the determinant of $S + \mathbb{I}_m$,

$$\begin{aligned} \det(V_i) &= \det(Z_i C Z_i^\top + S + \mathbb{I}_m) \\ &= \det(C^{-1} + Z_i^\top (S + \mathbb{I}_m)^{-1} Z_i) \det(C) \det(S + \mathbb{I}_m). \end{aligned}$$

For the models that we will consider, the latter computation is done by using the operator approximation proposed in [107] which, for image data with sufficiently high resolution (e.g. $m > 30$), gives a high-quality approximation of the determinant of the intensity covariance that can be computed in constant time.

By taking the described strategy, we never need to form a dense $m \times m$ matrix, and we can take advantage of the sparse and low-rank structures to reduce the computation time drastically. Furthermore, the fact that we assume equal-size images allows us to only do a single Cholesky factorization per likelihood computation, which is further accelerated by using the updating scheme described in [92].

4.3.2 Prediction

After the maximum-likelihood estimation of the template θ and the variance parameters, we have an estimate for the distribution of the warping parameters. We are therefore able to predict the warping functions that are most likely to have occurred given the observed data. This prediction parallels the conventional estimation of deformation functions in image registration. Let $p_{w_i|y_i}$ be the density for the distribution of the warping functions given the data and define p_{w_i}, $p_{y_i|w_i}$ in a similar manner. Then, by applying $p_{w_i|y_i} \propto p_{y_i|w_i} p_{w_i}$, we see that the warping functions that are most likely to occur are the minimizers of the posterior

$$-\log(p_{w_i|y_i}) \propto \frac{1}{2\sigma^2}(\boldsymbol{y}_i - \boldsymbol{\theta}^{\boldsymbol{w}_i})^\top (S + I_m)^{-1}(\boldsymbol{y}_i - \boldsymbol{\theta}^{\boldsymbol{w}_i}) + \frac{1}{2\sigma^2}\boldsymbol{w}_i^\top C^{-1}\boldsymbol{w}_i. \quad (4.9)$$

Given the updated predictions $\hat{\boldsymbol{w}}_i$ of the warping parameters, we update the estimate of the template and then minimize the likelihood (4.8) to obtain updated estimates of the variances. This procedure is then repeated until convergence is obtained. The estimation algorithm is given in Algorithm 1. The run time for the algorithm will be very different depending on the data in question. As an example we ran the model for 10 MRI midsaggital slices (for more details see Section 4.5.2) of size 210×210, with $i_{\max} = 5, j_{\max} = 3$. We ran the algorithm on an Intel Xeon E5-2680 2.5GHz processor. The run time needed for full maximum likelihood estimation in this setup was 1 hour and 15 minutes using a single core. This run time is without parallization, but it is possible to apply parallization to make the algorithm go faster.

The spatially correlated intensity variation can also be predicted. Either as the best linear unbiased prediction $\mathrm{E}[\boldsymbol{x}_i \mid \boldsymbol{y}]$ from the linearized model (4.7) (see e.g. equation 5 in [83]). Alternatively, to avoid a linear correction step when predicting $\boldsymbol{w}_i$, one can compute the best linear unbiased prediction given the maximum-a-posteori warp variables

$$\mathrm{E}[x_i(s,t) \mid \boldsymbol{y}_i, \boldsymbol{w}_i = \hat{\boldsymbol{w}}_i] = S(S + I_m)^{-1}(\boldsymbol{y}_i - \hat{\boldsymbol{\theta}}^{\hat{\boldsymbol{w}}_i}). \quad (4.10)$$

The prediction of the spatially correlated intensity variation can, for example, be used for bias field correction of the images.

4.4 Models for the spatially correlated variations

The main challenge of the presented methods is the computability of the likelihood function, in particular computations related to the $m \times m$ covariance matrix of the spatially correlated intensity variation S. The same issues are not associated with the covariance matrix C, for the warping parameters, as the dimensions of this matrix are considerably smaller than the dimensions of S. In the end of this section, we will give a short description of how the

Algorithm 1: Inference in the model (4.2).

Data: y
Result: Estimates of the fixed effect and variance parameters of the model, and the resulting predictions of the warping parameters w

```
// Initialize parameters
```
Initialize w^0
Compute $\hat{\theta}^{w^0}$ following (4.4)
for $i = 1$ *to* $i_{\max}$ **do**
```
    // Outer loop:  parameters
```
    Estimate variance parameters by minimizing (4.8)
    for $j = 1$ *to* $j_{\max}$ **do**
```
        // Inner loop:  fixed effect, warping
           parameters
```
        Predict warping parameters by minimizing (4.9)
        Update linearization points w^0 to current prediction
        Recompute $\hat{\theta}^{w^0}$ from (4.4)
    end
end

displacement vectors can be modeled, but first we consider the covariance matrix S.

As mentioned in the previous section, the path we will pursue to make likelihood computations efficient is to assume that the systematic random effect x_i has a covariance matrix S with sparse inverse. In particular, modeling x_i as a Gaussian Markov random field will give sparse precision matrices S^{-1}. The Markov random field structure gives a versatile class of models that has been demonstrated to be able to approximate the properties of general Gaussian fields surprisingly well [111]. Estimation of a sparse precision matrix is a fundamental problem and a vast literature exists on the subject. We mention in passing the fundamental works, [21, 39], which could be adapted to the present setup to estimate unstructured sparse precision matrices. We will however not pursue that extension in the present paper.

We here model x_i as a tied-down Brownian sheet, which is the generalization of the Brownian bridge (which is Markov) to the unit square $[0,1]^2$. The covariance function, $\mathcal{S}\colon [0,1]^2 \times [0,1]^2 \to \mathbb{R}$, for the tied-down Brownian sheet is

$$\mathcal{S}((s,t),(s',t')) = \tau^2(s \wedge s' - ss')(t \wedge t' - tt'), \qquad \tau > 0.$$

The covariance is 0 along the boundary of the unit square and reaches its maximal variance at the center of the image. These properties seem reason-

able for many image analysis tasks, where one would expect the subject matter to be centered in the image with little or no variation along the image boundary.

Let S be the covariance matrix for a Brownian sheet observed at the lattice spanned by

$$(s_1, \ldots, s_{m_1}) \text{ and } (t_1, \ldots, t_{m_2}),\ s_i = i/(m_1+1),\ t_i = i/(m_2+1)$$

with row-major ordering. The precision matrix S^{-1} is sparse with the following structure for points corresponding to non-boundary elements:

$$\frac{1}{\tau^2(m_1+1)(m_2+1)} S^{-1}[i,j] = \begin{cases} 4 & \text{if } j = i \\ -2 & \text{if } j \in \{i-1, i+1, i+m_2, i-m_2\} \\ 1 & \text{if } j \in \{i-1-m_2, i+1-m_2, \\ & \qquad i-1+m_2, i+1+m_2\} \end{cases}.$$

For boundary elements, the j elements outside the observation boundary vanish.

As explained in Section 4.3.1, the computational difficulties related to the computation of the log determinant in the negative log likelihood function (4.8) comes down to computing the log determinant of the intensity covariance $S + \mathbb{I}_m$. For the tied-down Brownian sheet, the log determinant can be approximated by means of the operator approximation given in [107, Example 3.4]. The approximation is given by

$$\log\det(S + \mathbb{I}_m) = \sum_{\ell=1}^{\infty} \log\left(\frac{\pi\ell}{\sqrt{\tau^2(m_1+1)(m_2+1)}} \sinh\left(\frac{\sqrt{\tau^2(m_1+1)(m_2+1)}}{\pi\ell} \right) \right).$$

To compute the approximation we cut the sum off after 10,000 terms.

As a final remark, we note that the covariance function $\tau^{-2}\mathcal{S}$ is the Green's function for the differential operator $\partial_s^2\partial_t^2$ on $[0,1]^2$ under homogeneous Dirichlet boundary conditions. Thus the conditional linear prediction of $\boldsymbol{x}_i$ given by (4.10) is equivalent to estimating the systematic part of the residual as a generalized smoothing spline with roughness penalty

$$\frac{1}{2\tau^2} \int_0^1 \int_0^1 x_i(s,t) \partial_s^2 \partial_t^2 x_i(s,t)\, \mathrm{d}s\, \mathrm{d}t = \frac{1}{2\tau^2} \int_0^1 \int_0^1 \|\partial_s \partial_t x_i(s,t)\|^2\, \mathrm{d}s\, \mathrm{d}t.$$

The tied-down Brownian sheet can also be used to model the covariance between the displacement vectors. Here the displacement vectors given by the warping variables $\boldsymbol{w}_i$ are modeled as discretely observed tied-down Brownian sheets in each displacement coordinate. As was the case for the intensity covariance, this model is a good match to image data since it allows the largest deformations around the middle of the image. Furthermore,

the fact that the model is tied down along the boundary means that we will predict the warping functions to be the identity along the boundary of the domain $[0,1]^2$, and for the found variance parameters, the predicted warping functions will be homeomorphic maps of $[0,1]^2$ onto $[0,1]^2$ with high probability.

In the applications in the next section, we will use the tied-down Brownian sheet to model the spatially correlated variations.

4.5 Applications

In this section, we will apply the developed methodology on two different real-life datasets. In the first example, we apply the model to a collection of face images that are difficult to compare due to varying expressions and lighting sources. We compare the results of the proposed model to conventional registration methods and demonstrate the effects of the simultaneous modeling of intensity and warp effects. In the second example, we apply the methodology to the problem of estimating a template from affinely aligned 2D MR images of brains.

4.5.1 Face registration

Figure 4.3: Ten images of the same face with varying expressions and illumination. The images are from the AT&T Laboratories Cambridge Face Database [112].

Consider the ten 92×112 face images from the AT&T Laboratories Cambridge Face Database [112] in Figure 4.3. The images are all of the same person, but vary in head position, expression and lighting. The dataset contains two challenges from a registration perspective, namely the differences in expression that cause dis-occlusions or occlusions (e.g. showing teeth, closing

eyes) resulting in large local deviations; and the difference in placement of the lighting source that causes strong systematic deviations throughout the face.

To estimate a template face from these images, the characteristic features of the face should be aligned, and the systematic local and global deviations should be accounted for. In the proposed model (4.2), these deviations are explicitly modeled through the random effect x_i.

Using the maximum-likelihood estimation procedure, we fitted the model to the data using displacement vectors $\boldsymbol{w}_i$ on an equidistant 4×4 interior grid in $[0,1]^2$. We used 5 outer and 3 inner iterations in Algorithm 1. The image value range was scaled to $[0,1]$. The estimated variance scale for the random effect x_i was $\hat{\sigma}^2\hat{\tau}^2 = 0.658$; for the warp variables, the variance scale was estimated to $\hat{\sigma}^2\hat{\gamma}^2 = 0.0680$; and for the residual variance, the estimated scale was $\hat{\sigma}^2 = 0.00134$.

To illustrate the effect of the simultaneous modeling of random intensity and warp effects, we estimated a face template using three more conventional variants of the proposed framework: a pointwise estimation that corresponds to model (4.2) with no warping effect; a *Procrustes* model that corresponds to model (4.2) with no intensity component and where the warp variables $\boldsymbol{w}_i$ were modeled as unknown parameters and estimated using maximum-likelihood estimation; and a *warp-regularized Procrustes* method where the warp variables $\boldsymbol{w}_i$ were penalized using a term $\lambda \boldsymbol{w}_i^\top C^{-1} \boldsymbol{w}_i$ where C^{-1} is the precision matrix for the 2D tied-down Brownian sheet with smoothing parameter $\lambda = 3.125$ (chosen to give good visual results).

The estimated templates for the proposed model and the alternative models described above can be found in Figure 4.4. Going from left to right, it is clear that the sharpness and representativeness of the estimates increase.

To validate the models, we can consider how well they predict the observed faces under the maximum-likelihood estimates and posterior warp predictions. These predictions are displayed in Figure 4.5. The rightmost column displays the five most deviating observed faces. From the left, the first three columns show the corresponding predictions from the Procrustes model, the warp-regularized Procrustes model and, for comparison, the predicted warped templates from the proposed model. It is clear that both the sharpness and the representativeness increase from left to right. The predictions in the third column show the warped template of model (4.2) which does not include the predicted intensity effect x_i. The fourth column displays the full prediction from the proposed model given as the best linear unbiased prediction conditional on the maximum-a-posteori warp variables $\hat{\theta}(v(s,t,\hat{\boldsymbol{w}}_i)) + \mathrm{E}[x_i(s,t) \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i = \hat{\boldsymbol{w}}_i]$. The full predictions are very faithful to the observations, with only minor visible deviations around the eyes in the second and fifth row. This suggests that the chosen model for the spatially correlated intensity variation, the tied-down Brownian sheet, is sufficiently versatile to model the systematic part of the residuals.

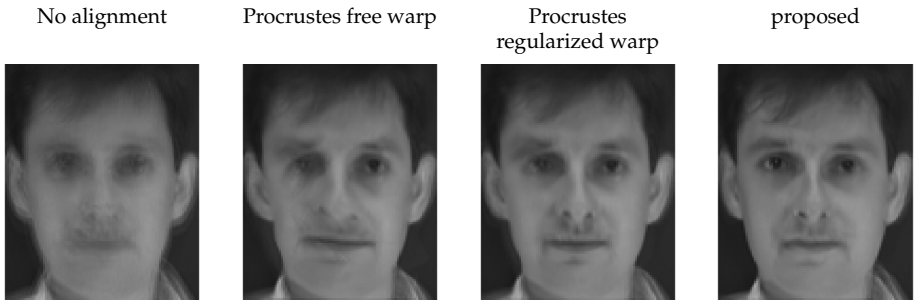


Figure 4.4: Estimates for the fixed effect θ using different models. The models used to calculate the estimates are from left to right: model assuming no warping effect and Gaussian white noise for the intensity model, the same model but with a free warping function based on 16 displacement vectors, the same model but with a penalized estimation of warping functions (2D tied-down Brownian sheet with scale fixed $\tau = 0.4$), the full model (4.2).

4.5.2 MRI slices

The data considered in this section are based on 3D MR images from the ADNI database [96]. We have based the example on 50 images with 18 normal controls (NC), 13 with Alzheimer's disease (AD) and 19 who are mild cognitively impaired (MCI). The 3D images were initially affinely aligned with 12 degrees of freedom and normalized mutual information (NMI) as a similarity measure. After the registration, the mid-sagittal slices were chosen as observations. Moreover the images were intensity normalized to $[0, 1]$ and afterwards the mid-sagittal plane was chosen as the final observations. The 50 mid-sagittal planes are given as 210×210 observations on an equidistant grid on $[0, 1]^2$. Six samples are displayed in Figure 4.6 where differences in both contrast, placement and shape of the brains are apparent.

For the given data, we used 25 displacement vectors $\boldsymbol{w}_i$ on an equidistant 5×5 interior grid in $[0, 1]^2$. The number of inner iterations in the algorithm was set to 3, while the number of outer iterations was set to 5 as the variance parameters and likelihood value already stabilized after a couple of iterations. The estimated variance scales are given by $\hat{\sigma}^2\hat{\tau}^2 = 2.23$ for the spatially correlated intensity variation, $\hat{\sigma}^2\hat{\gamma}^2 = 0.202$ for the warp variation and $\hat{\sigma}^2 = 7.79 \cdot 10^{-4}$ for the residual variance. The estimated template can be found in the rightmost column in Figure 4.7.

For comparison, we have estimated a template without any additional warping (i.e. only using the rigidly aligned slices), and a template estimated using a Procrustes model with fixed warping effects and no systematic intensity variation, but otherwise comparable to the proposed model. These templates can be found in the leftmost and middle columns of Figure 4.7. Comparing the three, we see a clear increase in details and sharpness from

Figure 4.5: Model predictions of five face images (rightmost column). The two first columns display the maximum-likelihood predictions from the Procrustes and regularized Procrustes models. The third column displays the warped template $\hat{\theta}(v(s,t,\hat{\boldsymbol{w}}_i))$ where $\hat{\boldsymbol{w}}_i$ is the most likely warp given data $\boldsymbol{y}$. The fourth column displays the full conditional prediction given the posterior warp variables $\hat{\theta}(v(s,t,\hat{\boldsymbol{w}}_i)) + \mathrm{E}[x_i(s,t) \mid \boldsymbol{y}_i, \boldsymbol{w}_i = \hat{\boldsymbol{w}}_i]$.

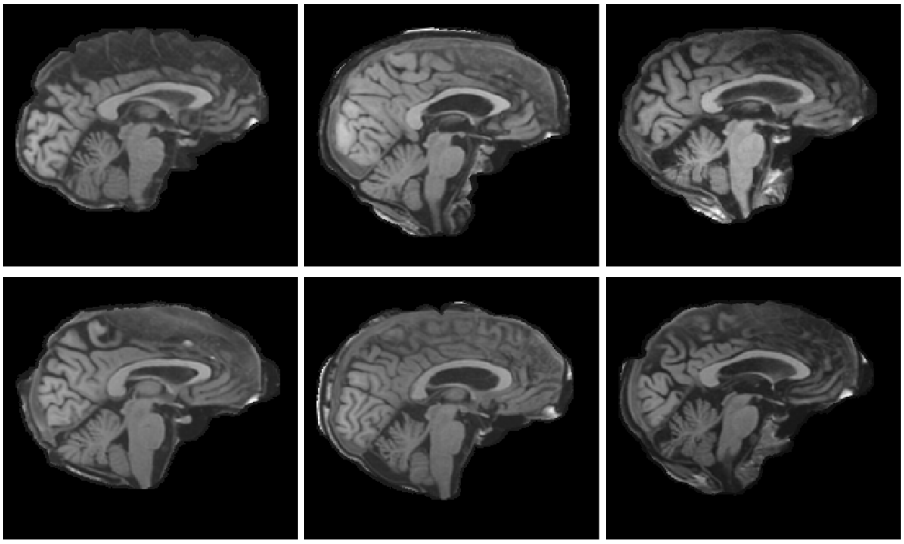


Figure 4.6: A sample of six MRI slices from the data set of 50 mid-sagittal MRI slices.

left to right. The reason for the superiority of the proposed method is both that the regularization of warps is based on maximum-likelihood estimation of variance parameters, but also that the prediction of warps takes the systematic deviations into account. Indeed, we can rewrite the data term in the posterior (4.9) as

$$(\boldsymbol{y}_i - \boldsymbol{\theta}^{\boldsymbol{w}_i} - \mathrm{E}[\boldsymbol{x}_i \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i])^\top (\boldsymbol{y}_i - \boldsymbol{\theta}^{\boldsymbol{w}_i} - \mathrm{E}[\boldsymbol{x}_i \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i]) \\ + \mathrm{E}[\boldsymbol{x}_i \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i]^\top S^{-1} \mathrm{E}[\boldsymbol{x}_i \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i].$$

Thus, in the prediction of warps, there is a trade-off between the regularity of the displacement vectors (the term $\boldsymbol{w}_i^\top C^{-1} \boldsymbol{w}_i$ in eq. 4.9) and the regularity of the predicted spatially correlated intensity variation given the displacement vectors (the term $\mathrm{E}[\boldsymbol{x}_i \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i]^\top S^{-1} \mathrm{E}[\boldsymbol{x}_i \,|\, \boldsymbol{y}_i, \boldsymbol{w}_i]$).

The difference in regularization of the warps is shown in Figure 4.8, where the estimated warps using the Procrustes model are compared to the predicted warps from the proposed model. We see that the proposed model predicts much smaller warps than the Procrustes model.

One of the advantages of the mixed-effects model is that we are able to predict the systematic part of the intensity variation of each image, which in turn also gives a prediction of the residual intensity variation—the variation that cannot be explained by systematic effects. In Figure 4.9, we have predicted the individual observed slices using the Procrustes model and the proposed model. As we also saw in Figure 4.8, the proposed model predicts less deformation of the template compared to the Procrustes model, and we see that the Brownian sheet model is able to account for the majority of the

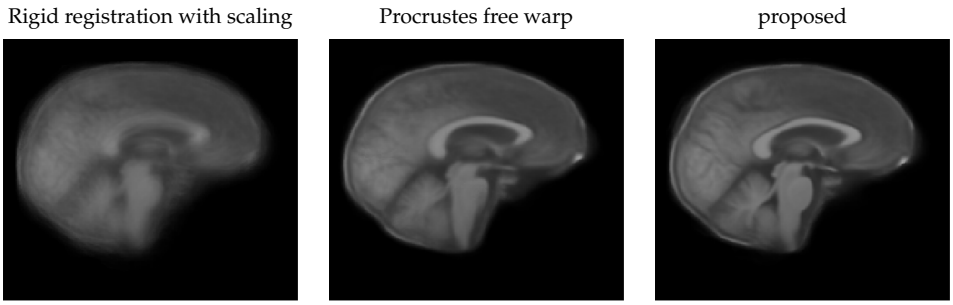


Figure 4.7: Estimates for the fixed effect θ in three different models. From left to right: pointwise mean after rigid registration and scaling; non-regularized Procrustes; and the proposed model (4.2).

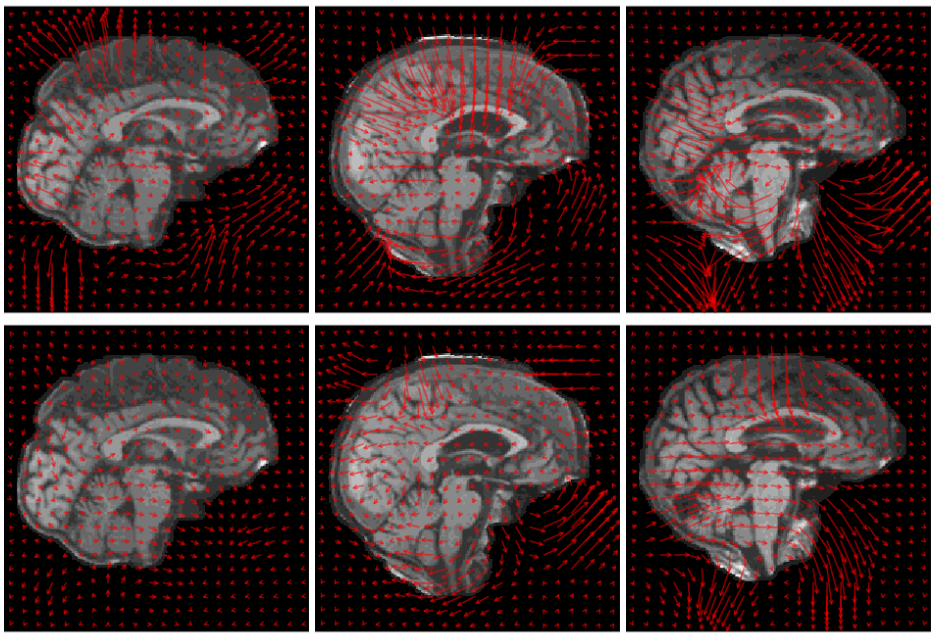


Figure 4.8: Three MRI slices and their estimated/predicted warping functions for the Procrustes model and the proposed model. The top row shows the Procrustes displacement fields, while the displacement fields for the proposed model are given in the bottom row. The arrows corresponds to the deformation of the observation to the template.

personal structure in the sulci of the brain. Moreover, the predicted intensity variation seems to model intensity differences introduced by the different MRI scanners well.

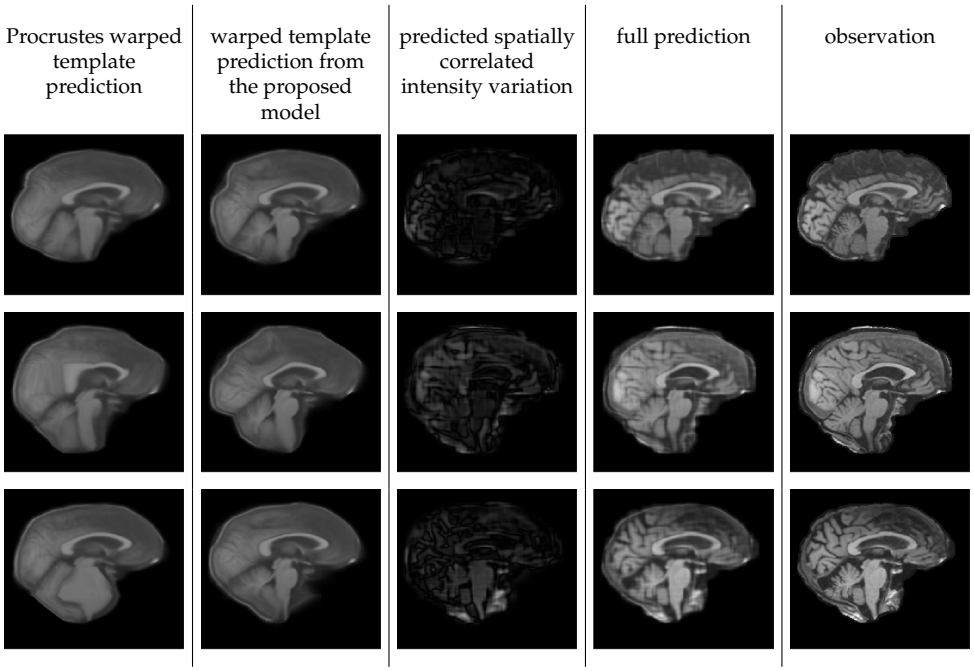


Figure 4.9: Model predictions of three mid-saggital slices (rightmost column). The first two rows display the warped templates from the Procrustes model and the proposed model. The third row displays the absolute value of the predicted spatially correlated intensity variation from the proposed model. The fourth row displays the full conditional prediction given the posterior warp variables $\hat{\theta}(v(s,t,\hat{\boldsymbol{w}}_i)) + \mathrm{E}[x_i(s,t) \mid \boldsymbol{y}_i, \boldsymbol{w}_i = \hat{\boldsymbol{w}}_i]$.

4.6 Simulation study

In this section, we present a simulation study for investigating the precision of the proposed model. The results are compared to the previously introduced models: Procrustes free warp and a regularized Procrustes. Data are generated from model (4.2) in which θ is taken as one of the MRI slices considered in Section 4.5.2. The warp, intensity and the random noise effects are all drawn from the previously described multivariate normal distributions with variance parameters respectively

$$\sigma^2\gamma^2 = 0.01, \quad \sigma^2\tau^2 = 0.1, \quad \sigma^2 = 0.001$$

and applied to the chosen template image θ. To consider more realistic brain simulations, the systematic part of the intensity effect was only added to the brain area of θ and not the background. As this choice makes the proposed model slightly misspecified, it will be hard to obtain precise estimates of the variance parameters. In practice, one would expect any model with a limited number of parameters to be somewhat misspecified in the presented setting.

The simulations thus present a realistic setup and our main interest will be in estimating the template and predicting warp and intensity effects. Figure 4.10 displays 5 examples of the simulated observations as well as the chosen θ.

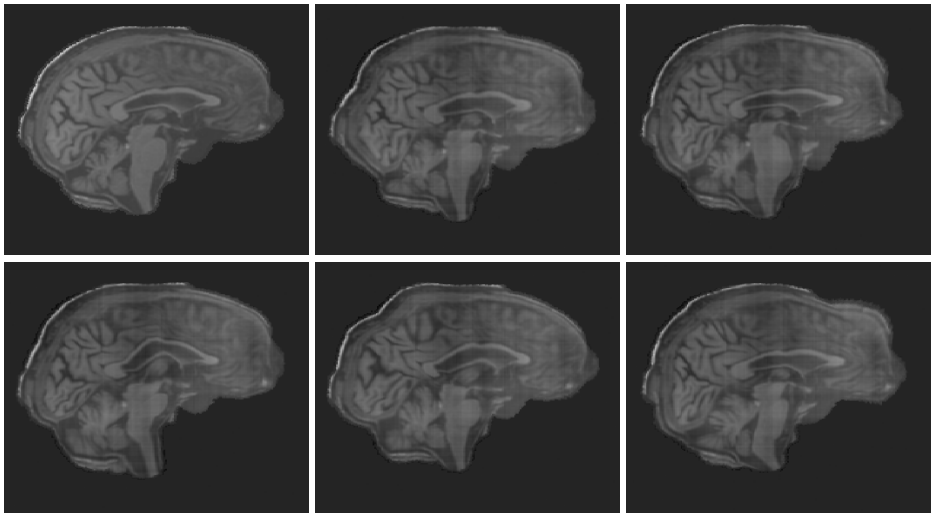


Figure 4.10: 5 examples of simulated brains. The template brain θ is shown in the upper left corner.

The study is based on 100 data sets of 100 simulated brains. For each simulated dataset we applied the proposed, Procrustes free warp and Procrustes regularized model. The regularization parameter, λ, in the regularized Procrustes model, was set to the true parameter used for generating the data $\lambda = \gamma^{-2}/2$.

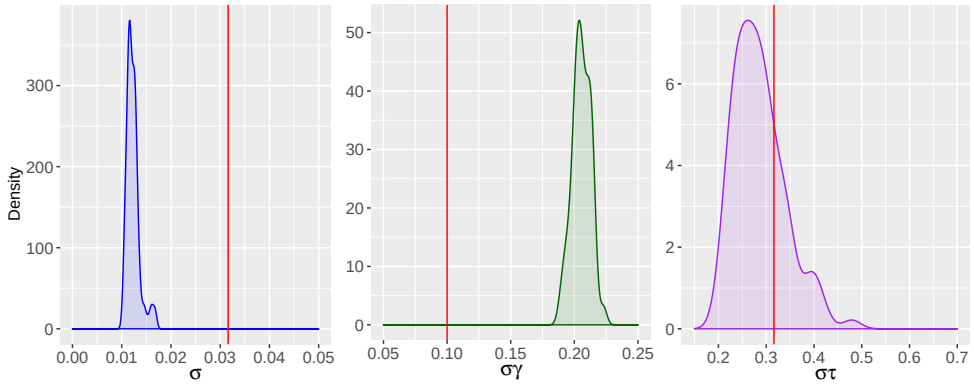


Figure 4.11: Density plots for the estimated variance parameters in the proposed model. The red lines correspond to the true parameters.

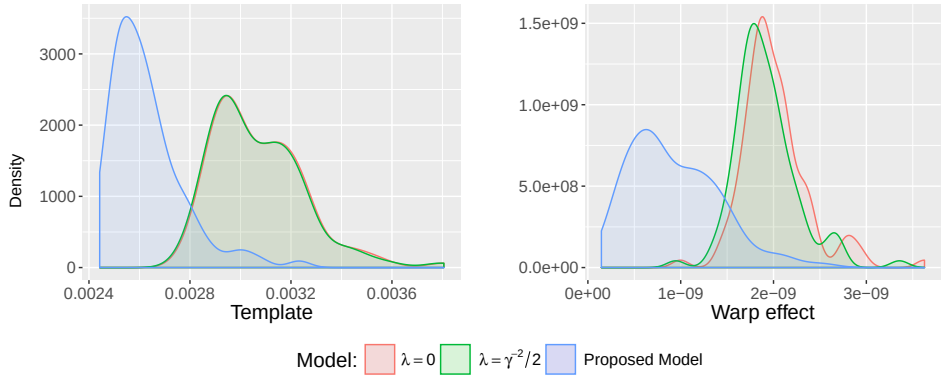


Figure 4.12: Density plots for the mean squared differences of template and warp estimates for the three models. The plot to the left shows the density for the mean squared difference for the template effect and the plot to the right shows the mean squared difference for the warp effect. $\lambda = 0$ denotes the procrustes free warp model, $\lambda = \gamma^{-2}/2$ is the Procrustes regularized model and the blue density corresponds to the Proposed model

The variance estimates based on the simulations are shown in Figure 4.11. The true variance parameters are plotted for comparison. We see some bias in the variance parameters. While bias is to be expected, the observed bias for the noise variance σ^2 and the warp variance scale $\sigma^2\gamma^2$ are bigger than what one would expect. The reason for the underestimation of the noise variance seems to be the misspecification of the model. Since the model assumes spatially correlated noise outside of the brain area, where there is none, the likelihood assigns the majority of the variation in this area to the systematic intensity effect. The positive bias of the warp variance scale seems to be a compensating effect for the underestimated noise variance.

The left panel of Figure 4.12 shows the mean squared difference for the estimated templates θ with the three types of models. We see that the proposed model produces conisderably more accurate estimates than the alternative frameworks.

To give an example of the difference between template estimates for the three different models, one set of template estimates for each of the models is shown in Figure 4.13. From this example we see that the template for the proposed model is slightly more sharp than the Procrustes models and are more similar to the true θ which was also the conclusion obtained from the density of the mean squared difference for the template estimates (Figure 4.12).

The right panel of Figure 4.12 shows the mean squared prediction/estimation error of the warp effects. The error is calculated using only the warp effects in the brain area since the background is completely untextured, and any warp effect in this area will be completely determined by the predic-

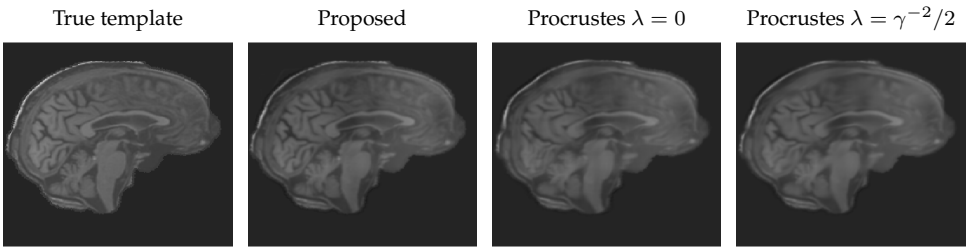


Figure 4.13: Example of a template estimate for each of the three models. For comparison, the true θ are plotted as well.

tion/estimation in the brain area. We find that the proposed model estimates warp effects that are closest to the true warps. It is worth noticing that the proposed model is considerably better at predicting the warp effects than the regularized Procrustes model. This happens despite the fact that the value for the warp regularization parameter in the model was chosen to be equal to the true parameter ($\lambda = \gamma^{-2}/2$). Examples of the true warping functions in the simulated data and the predicted/estimated effects in the different models are shown in Figure 4.14. None of the considered models are able to make sensible predictions on the background of the brain, which is to be expected. In the brain region, the predicted warps for the proposed model seem to be very similar to the true warp effect, which we also saw in Figure 4.12 was a general tendency.

4.7 Conclusion and outlook

We generalized the likelihood based mixed-effects model for template estimation and separation of phase and intensity variation to 2D images. This type of model was originally proposed for curve data [108]. As the model is computationally demanding for high dimensional data, we presented an approach for efficient likelihood calculations. We proposed an algorithm for doing maximum-likelihood based inference in the model and applied it to two real-life datasets.

Based on the data examples, we showed how the estimated template had desirable properties and how the model was able to simultaneously separate sources of variation in a meaningful way. This feature eliminates the bias from conventional sequential methods that process data in several independent steps, and we demonstrated how this separation resulted in well-balanced trade-offs between the regularization of warping functions and intensity variation.

We made a simulation study to investigate the precision of the template and warp effects of the proposed model and for comparison with two other

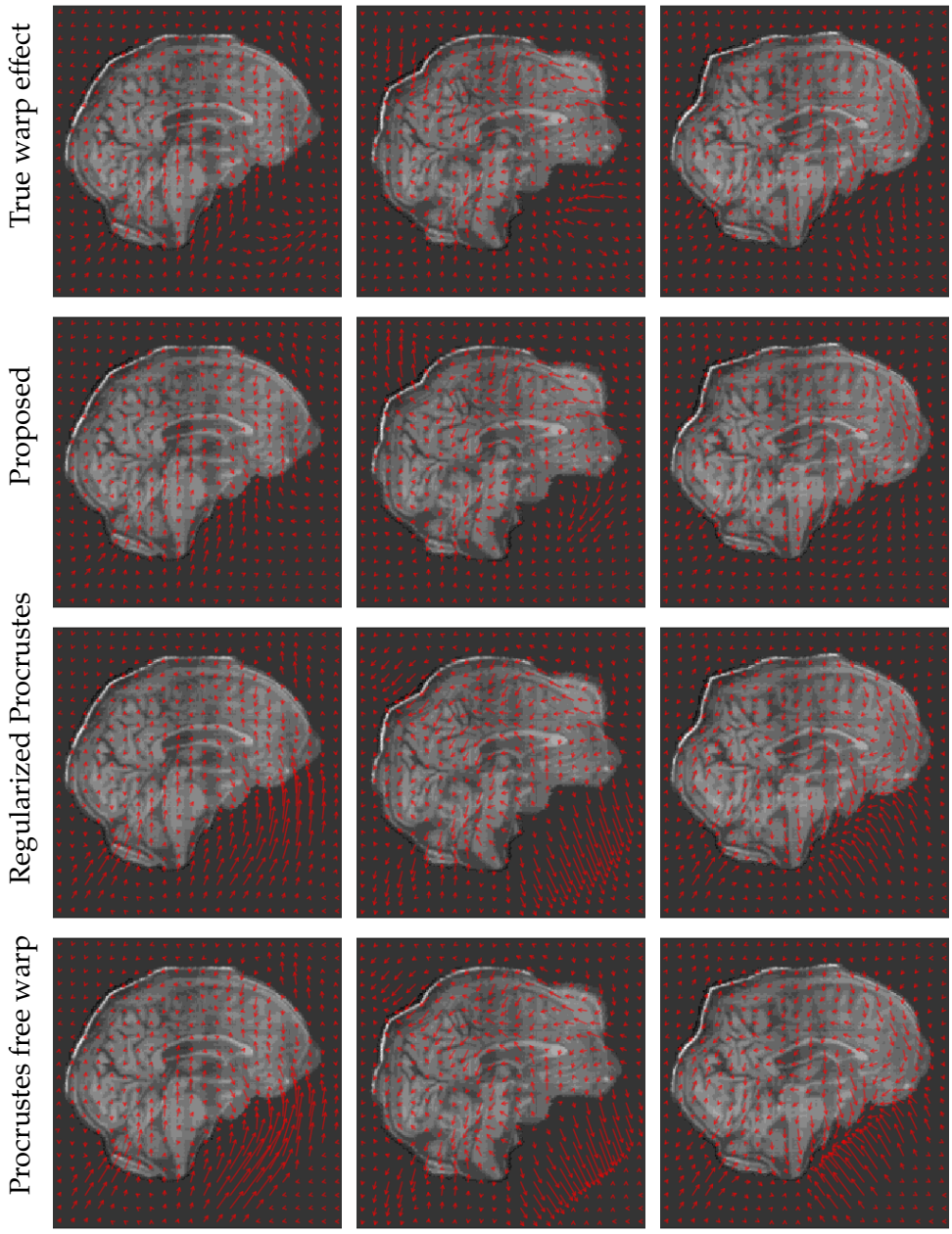


Figure 4.14: Examples of predicted warp effect for each model. The top row shows the true warp effect, the second row the estimated warp effect of the proposed model, the third row regularized Procrustes and the final row, the Procrustes model with free warps.

models. The proposed model was compared with a Procrustes free warp model, as well as a Procrustes regularized model. Since the noise model was misspecified, the proposed methodology could not recover precise maximum likelihood estimates of the variance parameters. However, the maximum likelihood estimate for the template was seen to be a lot sharper and closer to the true template compared to alternative Procrustes models. Furthermore, we demonstrated that the proposed model was better at predicting the warping effect than the alternative models.

The main restriction of the proposed model is the computability of the likelihood function. We resolved this by modeling intensity variation as a Gaussian Markov random field. An alternative approach would be to use the computationally efficient operator approximations of the likelihood function for image data suggested in [107]. This approach would, however, still require a specific choice of parametric family of covariance functions, or equivalently, a family of positive definite differential operators. An interesting and useful extension would be to allow a free low-rank spatial covariance structure and estimate it from the data. This could, for example, be done by extending the proposed model (4.2) to a factor analysis model where both the mean function and intensity variation is modeled in a common functional basis, and requiring a specific rank of the covariance of the intensity effect. Such a model could be fitted by means of an EM algorithm similar to the one for the reduced-rank model for computing functional principal component analysis proposed in [55], and it would allow simulation of realistic observations by sampling from the model.

For the computation of the likelihood function of the nonlinear model, we relied on local linearization which is a simple well-proven and effective approach. In recent years, alternative frameworks for doing maximum likelihood estimation in nonlinear mixed-effects models have emerged, see [23] and references therein. An interesting path for future work would be to formulate the proposed model in such a framework that promises better accuracy than the local linear approximation. This would allow one to investigate how much the linear approximation of the likelihood affects the estimated parameters. In this respect, it would also be interesting to compare the computing time across different methods to identify a suitable tradeoff between accuracy and computing time.

The proposed model introduced in this paper is a tool for analyzing 2D images. The model, as it is, could be used for higher dimensional images as well, but the analysis would be computationally infeasible with the current implementation. To extend the proposed model to 3D images there is a need to devise new computational methods for improving the calculation of the likelihood function.

CHAPTER 5

Stochastic Image Deformation in Frequency Domain and Parameter Estimation using Moment Evolutions

The following paper is currently under review, but has been published on ArXiv as submission,

- L. Kühnel, A. Arnaudon, T. Fletcher, and S. Sommer. Stochastic Image Deformation in Frequency Domain and Parameter Estimation using Moment Evolutions. *arXiv:1812.05537 [cs, math, stat]*, December 2018. arXiv: 1812.05537

It is joint work with Alexis Arnaudon, Tom Fletcher and Stefan Sommer. The paper presents a method for modelling uncertainty in image data by combining the stochastic LDDMM framework described in [5] with the fast LDDMM Fourier solver, FLASH, defined in [140].

Stochastic Image Deformation in Frequency Domain and Parameter Estimation using Moment Evolutions

Line Kühnel[1], Alexis Arnaudon[2], Tom Fletcher[3] and Stefan Sommer[1]

Department of Computer Science (DIKU), University of Copenhagen, Denmark[1]

Department of Mathematics Imperial College London, UK[2]

Department of Electrical and Computer Engineering / Department of Computer Science, University of Virginia, USA[3]

Abstract

Modelling deformation of anatomical objects observed in medical images can help describe disease progression patterns and variations in anatomy across populations. We apply a stochastic generalisation of the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework to model differences in the evolution of anatomical objects detected in populations of image data. The computational challenges that are prevalent even in the deterministic LDDMM setting are handled by extending the FLASH LDDMM representation to the stochastic setting keeping a finite discretisation of the infinite dimensional space of image deformations. In this computationally efficient setting, we perform estimation to infer parameters for noise correlations and local variability in datasets of images. Fundamental for the optimisation procedure is using the finite dimensional Fourier representation to derive approximations of the evolution of moments for the stochastic warps. Particularly, the first moment allows us to infer deformation mean trajectories. The second moment encodes variation around the mean, and thus provides information on the noise correlation. We show on simulated datasets of 2D MR brain images that the estimation algorithm can successfully recover parameters of the stochastic model.

Keywords: Uncertainty Estimation, Stochastic LDDMM Registration, FLASH, Method of Moments, Fokker-Planck equations.

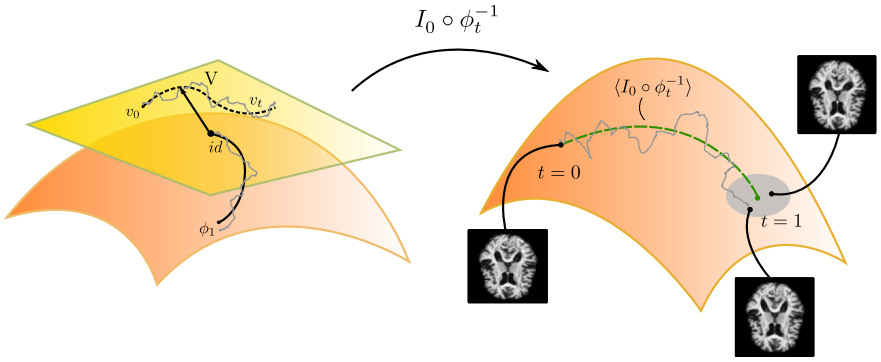


Figure 5.1: The presented stochastic deformation model includes both the average population trend over time starting at time $t = 0$, and the subject specific stochastic evolution. The deformation ϕ_t^{-1} is modelled as a stochastic perturbation of the LDDMM flow of diffeomorphisms resulting in a stochastic deformation $I_0 \circ \phi_t^{-1}$ of the image I_0. The stochastic deformation forms at each time point a distribution around the population trend describing uncertainty in the evolution. The mean flow $\langle I_0 \circ \phi_t^{-1} \rangle$ (dashed green line) models the general population evolution.

5.1 Introduction

Classical models describing the evolution of anatomical objects, occurring from child development, from natural ageing, or from disease processes, are generally smooth and deterministic. However, when analysing such deformations across populations of subjects, the individual deviations have to be taken into account as they otherwise affect the average trend in the evolution of the population. It is natural to assume that the subject-specific deformations are not purely deterministic and that stochastic variation may occur at any time of the evolution process. In this paper, we develop the technical framework to model such combinations of population average and subject-specific stochastic evolutions.

Shape changes of anatomical objects, observed in an image I_0, can be modelled using image registration. Here, the goal is to determine a deformation $\phi \colon \mathcal{D} \to \mathcal{D}$ of the image domain $\mathcal{D}$, minimising an energy $E(\phi) = R(\phi) + S(I_0 \circ \phi^{-1}, I_1)$, for a regularisation R and similarity measure S, see for example [135]. In this framework, the deformation ϕ is a deterministic, smooth, bijective function independent of time. The Large Deformation Diffeomorphic Metric Mapping framework (LDDMM, see e.g. [12]), defines the deformation ϕ as the endpoint of a flow of diffeomorphisms ϕ_t solution to

the differential equation

$$\frac{d}{dt}\phi_t(x,y) = v_t \circ \phi_t(x,y)\,, \tag{5.1}$$

with initial value $\phi_0 = \mathrm{id}$. The time-dependent velocity field v_t is for each t an element of the space of vector fields on $\mathcal{D}$, $V = \mathfrak{X}(D)$, and solves the EPDiff equation $\frac{d}{dt}v_t = -K\mathrm{ad}^*_{v_t}m_t$. Here $m_t = \frac{\delta l}{\delta v_t} = K^{-1}v_t = Lv_t$ is the conjugate momentum of the velocity v_t if the regularisation $R(\phi) = l(v) = \frac{1}{2}\|v\|^2_K$ corresponds to the kinetic energy of the flow with a reproducible kernel K, see for example [135] and references therein. Considering deformations via a flow in the diffeomorphism group introduces a natural time component which can be used to model the evolution of anatomical objects over time.

The flow of diffeomorphisms in the LDDMM model is deterministic, and it is hence only possible to introduce uncertainties in the initial velocity field v_0. Random variation of the initial velocity field has been discussed in, for example, the random orbit model of [86] and, for Bayesian principal geodesic analysis, [139].

Modelling uncertainty by a random initial velocity field implies that the entire variation over time is the result of uncertainty at the initial point of the evolution process. For longitudinal models, it is arguably more natural to model uncertainty time-continuously and thus having randomness occurring through the entire evolution process. To enable random evolution, we aim at modelling the variation as the endpoint of a *stochastic* flow of deformations, ϕ_t. This, in turn, gives a separation between the deterministic mean population evolution, and the per subject individual stochastic trajectories as illustrated in Fig. 5.1.

In this work, we use the stochastic LDDMM framework presented in [5], based on the stochastic fluid dynamics model introduced in [49]. Arnaudon et al. [5] used a stochastic flow of diffeomorphisms to model uncertainties in the evolution of any data type on which the diffeomorphism group acts, in particular for landmarks. The stochastic flow is defined by a stochastic differential equation (SDE) with a diffusion term parametrised by Eulerian noise fields $\sigma(x)$ instead of a standard Lagrangian noise associated with the flow. In the latter case, [128, 132] studied a stochastic model of landmarks dynamics with a different noise for each landmark and more recently [85] added dissipation to this model.

The stochastic LDDMM with Eulerian noise fields of [5] applies to any data structure on which the diffeomorphism group acts without modification of the noise structure. Thus, in addition to landmarks, which have a finite dimensional structure, the framework can be applied to complete images [6], provided that a good finite dimensional approximation can be applied to make numerical simulations possible. In this paper, we extend the fast LDDMM solver FLASH presented in [140] to include the stochastic deformations. The algorithm is based on spatial Fourier transformation of a

stochastic version of the EPDiff equation which results in a natural truncation of the high-frequencies in the stochastic process v_t. For Eulerian noise field, this is not an issue as we have total control on the spatial correlation of the noise (and white noise in time). Thus we avoid introducing high frequencies with the noise and thus obtain a good numerical approximation of the flow. This truncation gives rise to a dimensionality reduction resulting in a significant computational speed up, and it makes the use of noise in image matching possible and relevant for applications.

With the paper, we make the following contributions: 1) we incorporate stochasticity in the FLASH framework to model stochastic image evolutions in a finite-dimensional setting; 2) we derive the Fokker-Planck equations in the finite dimensional model and use this to approximate the evolution of the moments of the stochastic flow and deformed images; 3) we show how matching of the moment images can be used to estimate unknown parameters in the model; 4) we illustrate the use of the model on 2D brain images by recovering parameters from images simulated from the model.

5.2 Stochastic Image Deformation

The LDDMM framework models deformation over a time region as a flow, ϕ_t, in the space of diffeomorphisms $\mathrm{Diff}(\mathcal{D})$. As ϕ_t varies smoothly and deterministically, applying the flow to an image I_0 results in a time evolution of the image which does not describe any uncertainty in the deformation. In this section, we describe the stochastic LDDMM extension [5] which exactly models the uncertainty of deformations.

We consider a probability space $(\Omega, P, \mathcal{F})$ and let $\phi_t \colon \Omega \times [0,1] \times \mathcal{D} \to \mathcal{D}$ be a stochastic process on the time interval $[0,1]$, i.e. for each $t \in [0,1]$, $\omega \in \Omega$, $\phi_t(\omega) \colon \mathcal{D} \to \mathcal{D}$ is a deterministic deformation of the image domain $\mathcal{D}$. The stochastic flow ϕ_t is defined as a Stratonovich SDE with diffusion term based on noise fields, $\sigma_k \in \mathfrak{X}(\mathcal{D})$, $k = 1, \ldots, p$, on the image domain $\mathcal{D}$. See Fig. 5.2 for an example of noise fields describing the stochastic flow of deformations. The Stratonovich SDE defining the flow ϕ_t is,

$$d\phi_t = v_t(\phi_t)dt + \sum_{k=1}^{p} \sigma_k(\phi_t) \circ_S dW_t^k, \quad \phi_0 = \mathrm{id}\,, \tag{5.2}$$

where W_t^k denotes p one-dimensional Wiener processes (or Brownian motions) on $\mathbb{R}$ and $\circ_S$ Stratonovich integration. The time-varying velocity field v_t is the solution of the stochastic Euler-Poincaré equation,

$$dv_t = -K\mathrm{ad}^*_{v_t} m_t - \sum_{k=1}^{p} K\mathrm{ad}^*_{\sigma_k} m_t \circ_S dW_t^k, \quad v_0 \in \mathfrak{X}(\mathcal{D})\,, \tag{5.3}$$

for the same momentum $m_t = Lv_t$ as in the deterministic equations. In [5], it was shown that under the stochastic deformation (5.2), the momentum is preserved and (5.3) is hence the stochastic version of the EPDiff equation.

Zhang et al. [140] observed that the last operation of the EPDiff equation is applying a low-pass filter K. As K is an operator suppressing all high-frequencies in Fourier domain, performing dimensionality reduction on the number of frequencies results in a large computational gain, and, importantly, only a restricted amount of information being removed. Similarly to the drift, the stochastic term of (5.3) has the smoothing operator K, applied exclusively to the spatially dependent noise amplitude, and not dW_t^k. Hence, it is also possible to benefit from the computational speedup of FLASH in the stochastic setting. We make a spatial Fourier transformation of dv_t and truncate the high-frequencies of this stochastic field. Details on the FLASH Fourier space calculation in the deterministic setting can be found in [140]. As an example, stochastic shooting of an 128×128 image truncated to 16 Fourier frequencies with 100 times steps and 1 noise field on a standard laptop (i7 processor) takes approximately 1.5 seconds.

The stochastic deformation ϕ_t models both the population trend and the subject-specific variations. The population trend is represented by the deterministic part of (5.2) and (5.3). It is a function of the initial velocity field v_0 and describes the global trend of the population, e.g. ageing or a disease progression. The noise fields, on the other hand, describe the subject-specific variation present in the data. The noise fields are modelled as local objects of the full dimensional data space with correlations decreasing with the distance to the centre of the object, e.g. Gaussian kernels on the domain.

5.3 Moment Matching

Consider n individuals observed at $t = 0$, $I_0^1, \ldots, I_0^n$ and again at $t = 1$, $I_1^1, \ldots, I_1^n$. The goal is to model the population trend and infer the noise structure describing the variation in the observed data at $t = 1$. For this, we aim at estimating the parameters of the stochastic deformation model. These parameters are the noise fields $\sigma_1, \ldots, \sigma_p$, the initial velocity field v_0, and parameters of the LDDMM RKHS kernel K. In this paper, we focus on estimating the parameters of the noise fields $\sigma_1, \ldots, \sigma_p$.

We base parameter estimation on method of moments, i.e. we seek to match moments of the observed data distribution with moments of the distribution of images at $t = 1$ generated by the stochastic deformation model. For simplicity, we remove the subject effect at $t = 0$ by considering a single initial image I_0, the average $\hat{I}_0 = \frac{1}{n}\sum_{i=1}^n I_0^i$. This implies that population variation at time $t = 0$ is removed and that the model, therefore, needs to account for the entire population variation at $t = 1$. See Fig. 5.1 for a visualisation of the model. Moments of the random variable $\hat{I}_0 \circ \phi_1^{-1}$ are matched

to the data moments, to retrieve the parameters for the noise fields σ_k. The model can handle subject specific initial images with minor changes to the moment equations presented in Section 5.3.2.

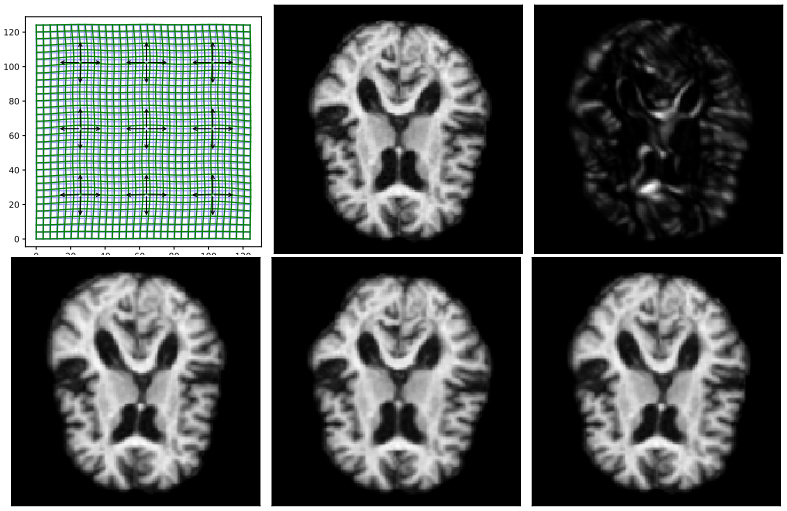
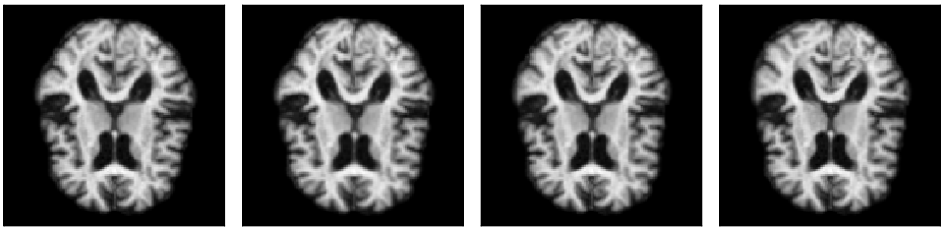


Figure 5.2: *First row:* (left) The noise fields location on the domain. (middle) Initial brain, I_0. (right) Variation in the data sample. *Second row:* 3 examples of simulated data by the stochastic deformation ϕ_t^{-1}.



(a) $t = 0$ (b) $t = 0.25$ (c) $t = 0.5$ (d) $t = 1$

Figure 5.3: 4 time steps of a sample of the stochastic deformation process $I_0 \circ \phi_t^{-1}$ at $t = 0,\ 0.25,\ 0.5,\ 1$.

5.3.1 Inverse of the flow

To calculate the moments of the stochastic deformation, we need to consider the nonlinear coupling between the process ϕ_t and the image. As the action of ϕ on I_0 is by composition with ϕ_t^{-1}, we determine the SDE of the inverse ϕ_t^{-1}, i.e. for each $t \in [0, 1],\ \omega \in \Omega$, $\phi_t(\omega, \phi_t^{-1}(\omega, x, y)) = (x, y)$ for $(x, y) \in \mathcal{D}$.

By the Itô-Wentzel theorem [62], the SDE of ϕ_t^{-1} is,

$$d\phi_t^{-1} = -D\phi_t^{-1} v_t dt - \sum_{k=1}^{p} D\phi_t^{-1} \sigma_k \circ_S dW_t^k \,. \tag{5.4}$$

By drawing sample paths of the stochastic flow ϕ_t^{-1}, we obtain samples of deformed images under the model. Sample images at $t = 1$ are shown in Fig. 5.2, together with a plot of the generated image variation, and the noise fields used to simulate the sample data. Fig. 5.3 shows 4 time points from a sample path of the stochastic process $I_0 \circ \phi_t^{-1}$ for $t = 0,\ 0.25,\ 0.5,\ 1$.

5.3.2 Moments of Stochastic Image Deformation

To approximate the first order moment of $\hat{I}_0 \circ \phi_1^{-1}$, we consider a first order Taylor approximation of $\hat{I}_0 \circ \phi_1^{-1}$ around the mean $\langle \phi_1^{-1} \rangle$, given as

$$\hat{I}_0 \circ \phi_1^{-1} \approx \hat{I}_0 \circ \langle \phi_1 \rangle + \nabla(\hat{I}_0 \circ \langle \phi_1^{-1} \rangle)^T (\phi_1^{-1} - \langle \phi_1^{-1} \rangle) \,. \tag{5.5}$$

We consider two different means: $\langle \cdot \rangle$ which denote the mean of the stochastic processes ϕ_t and v_t, and $\mathbb{E}[\hat{I}_0 \circ \phi_1^{-1}]$ for the expectation in image space. Applying $\mathbb{E}$ to the Taylor approximation (5.5) and using that $\nabla(\hat{I}_0 \circ \langle \phi_1^{-1} \rangle)$ is deterministic and can be moved outside the mean function, we obtain the approximation

$$\mathbb{E}[\hat{I}_0 \circ \phi_1^{-1}] \approx \hat{I}_0 \circ \langle \phi_1^{-1} \rangle \,. \tag{5.6}$$

In a similar manner, an approximation of the variance $\mathrm{Var}[\hat{I}_0 \circ \phi_1^{-1}]$ which singularly depend on the transition variance of the stochastic process ϕ_t^{-1}, can be described by applying the first order Taylor approximation of $\hat{I}_0 \circ \phi_1^{-1}$:

$$\begin{aligned}
\mathrm{Var}\left[\hat{I}_0 \circ \phi_1^{-1}\right] &= \mathbb{E}\left[(\hat{I}_0 \circ \phi_1^{-1})^2\right] - \left(\mathbb{E}[\hat{I}_0 \circ \phi_1^{-1}]\right)^2 \\
&\approx \mathbb{E}\left[\left(\hat{I}_0 \circ \langle \phi_1^{-1} \rangle + \left(\nabla(\hat{I}_0 \circ \langle \phi_1^{-1} \rangle)\right)^T (\phi_1^{-1} - \langle \phi_1^{-1} \rangle)\right)^2\right] - \left(\mathbb{E}[\hat{I}_0 \circ \phi_1^{-1}]\right)^2 \\
&\approx \left(\nabla(\hat{I}_0 \circ \langle \phi_1^{-1} \rangle)^2\right)^T \left\langle \left(\phi_1^{-1} - \langle \phi_1^{-1} \rangle\right)^2 \right\rangle \,.
\end{aligned} \tag{5.7}$$

In the last approximation of (5.7), we used the approximation of the mean value presented in (5.6). To determine the moments of $\hat{I}_0 \circ \phi_1^{-1}$, we therefore need the transition moments of the stochastic flow ϕ_t^{-1}. These moments are studied in the next section by considering the Fokker-Planck equation.

5.3.3 Moment Equations for ϕ_t^{-1} and $\tilde{v}_t$

The transition distributions of the stochastic process ϕ_t^{-1} and the Fourier transformation of v_t denoted $\tilde{v}_t$, can be determined via the Kolmogorov forward equation also called the Fokker-Planck equation. Based on this and

the Kolmogorov operator $\mathcal{L}$, the moment evolution of a transformation of a stochastic process X_t under a real-valued function h is described by

$$\frac{d}{dt}\langle h(X_t)\rangle = \langle \mathcal{L}h(X_t)\rangle\,. \tag{5.8}$$

More information on the procedure for calculating the moment evolution via the Fokker-Planck equation can be found in [5].

For sake of notation, let $\psi_t = \phi_t^{-1}$ and

$$\psi_{ij} = \psi_t(x_i, y_j) = (\psi_t^x(x_i, y_j), \psi_t^y(x_i, y_j))$$

for a discretisation of the image domain $\mathcal{D}$ to a grid $(x_i, y_j)_{ij}$. When considering a stochastic process, e.g. ψ_t, described by an Itô SDE, the Kolmogorov operator $\mathcal{L}$ is found by applying Itô's formula. The Kolmogorov operator is defined as the drift of the resulting stochastic process. As an example, the Kolmogorov operator for the 2-dimensional stochastic process $\psi_{ij,t}$ is given by

$$\begin{aligned}\mathcal{L}f = \frac{\partial f}{\partial t} + \sum_{\alpha\in\{x,y\}} \Bigg[&\frac{\partial f}{\partial \psi_{ij}^\alpha}(-D\psi_{ij}v_t(x_i,y_j) \\ &+ \frac{1}{2}\sum_k D[D\psi_{ij}\sigma_k(x_i,y_j)]D\psi_{ij}\sigma_k(x_i,y_j))_\alpha\Bigg] + \frac{1}{2}\sum_{\alpha,\beta\in\{x,y\}} C_{\alpha\beta}\frac{\partial^2 f}{\partial\psi_{ij}^\alpha\partial\psi_{ij}^\beta}\,,\end{aligned}$$

for $C = bb^T$, $b = (D\psi_{ij}\sigma_1(x_i,y_j)\cdots D\psi_{ij}\sigma_p(x_i,y_j))$, and a twice differentiable real-valued function f.

For the Kolmogorov operator for ψ_{ij}, the evolution of the moments are determined by (5.8). Let $\gamma \in \{x, y\}$ be given and (x_i, y_j) be a fixed pixel in the grid. Consider the time evolution of the first moment of ψ_{ij}^γ

$$\begin{aligned}\frac{\partial}{\partial t}\langle\psi_{ij}^\gamma\rangle &= \langle\mathcal{L}\psi_{ij}^\gamma\rangle \\ &= \left\langle(-D\psi_{ij}v_t(x_i,y_j) + \frac{1}{2}\sum_k D[D\psi_{ij}\sigma_k(x_i,y_j)]D\psi_{ij}\sigma_k(x_i,y_j))_\gamma\right\rangle \\ &\approx (-D\langle\psi_{ij}\rangle\langle v_t(x_i,y_j)\rangle + \frac{1}{2}\sum_k D[D\langle\psi_{ij}\rangle\sigma_k(x_i,y_j)]D\langle\psi_{ij}\rangle\sigma_k(x_i,y_j))_\gamma\,.\end{aligned} \tag{5.9}$$

Notice the coarse approximation used in (5.9) to describe the time evolution uniquely by moments of ψ_t and v_t. This approximation is used in the rest of the paper and assumes independence of the random variables. That is, any mean of a product of random variables ψ_{ij}^γ, v_{ij}^γ is approximated by the product of the first order moments of each random variable. Extending the equations to include higher order correlation will be the topic of future works.

The derivation of the variance of ψ_{ij}^{γ} is calculated as above and results in the moment evolution

$$\begin{aligned}
&\frac{\partial}{\partial t}\left\langle (\psi_{ij}^{\gamma} - \langle \psi_{ij}^{\gamma} \right\rangle)^2\rangle = \langle \mathcal{L}(\psi_{ij}^{\gamma} - \langle \psi_{ij}^{\gamma}\rangle)^2\rangle \\
&= \Bigg\langle 2(\psi_{ij}^{\gamma} - \langle \psi_{ij}^{\gamma}\rangle)(-D\psi_{ij} v_t(x_i, y_j) \\
&\qquad + \frac{1}{2}\sum_k D(D\psi_{ij}\sigma_k(x_i, y_j))D\psi_{ij}\sigma_k(x_i, y_j))^{\gamma} \Bigg\rangle + \langle C_{\gamma\gamma}\rangle \approx \langle C_{\gamma\gamma}\rangle \,,
\end{aligned}$$

where C is given as above and where the assumption of independence between variables is used to split the first term into the product of $\langle \psi_{ij}^{\gamma} - \langle \psi_{ij}^{\gamma}\rangle\rangle$ and $\left\langle -D\psi_{ij} v_t(x_i, y_j) + \frac{1}{2}\sum_k D(D\psi_{ij}\sigma_k(x_i, y_j))D\psi_{ij}\sigma_k(x_i, y_j))^{\gamma}\right\rangle$.

The moment evolution of ψ_{ij}^{γ} in (5.9) depends on the first order moment of the time-varying velocity field v_t. As described in Section 5.2, applying a spatial discrete Fourier transform to a discretization, $\boldsymbol{v}_t$, of v_t to a grid $(x_i, y_j)_{ij}$, results in a computationally feasible optimisation procedure. Using the property $\mathcal{F}[\langle \boldsymbol{v}_t^{\alpha}\rangle] = \langle \mathcal{F}[\boldsymbol{v}_t^{\alpha}]\rangle = \langle \tilde{\boldsymbol{v}}_t\rangle$, the moment evolution of $\tilde{\boldsymbol{v}}_t$ is calculated by the same procedure as above applied to the Fourier transform of the Itô SDE presented in Section 5.2. The moment evolution is given as

$$\begin{aligned}
&\frac{d}{dt}\langle \tilde{v}_{t,ij}^{\alpha}\rangle = \langle \mathcal{L}\tilde{v}_{t,ij}^{\alpha}\rangle \\
&\qquad \approx -\left(K\mathrm{ad}_{\langle \tilde{v}_t\rangle}^{*}\langle \tilde{m}_t\rangle\right)_{ij}^{\alpha} + \frac{1}{2}\sum_{k=1}^{p}\left((\tilde{D}[K\mathrm{ad}_{\tilde{\sigma}_k}^{*}\langle \tilde{m}_t\rangle) * (K\mathrm{ad}_{\tilde{\sigma}_k}^{*}\langle \tilde{m}_t\rangle)\right)_{ij}^{\alpha}.
\end{aligned}$$

The above moment equation is described in Fourier domain. Here, $*$ denotes convolution, and $\tilde{D}$ a central difference Jacobian matrix (for more information see [140]). Using the Fourier representaition, it takes approximately 8 seconds to solve the moment equations in a situation of 1 noise field, 100 time steps, truncated to 16 Fourier frequencies and on a standard laptop.

5.3.4 Similarity of Moment Images

With the method of moments, we seek to maximise the similarity between the moments of the observed data and moments of the stochastic deformation of the initial image $\hat{I}_0$. For an example of the variance of the deformed images and the data sample variance, see Fig. 5.4. Note that the approximation of the mean $\hat{I}_0 \circ \langle \phi_1^{-1}\rangle$ and the variance $\mathrm{Var}[\hat{I}_0 \circ \phi_1^{-1}]$ are images themselves. Therefore, moment matching turns into matching of images, however not a match of observed images as regularly performed in image registration, but instead a match of $\hat{I}_0 \circ \langle \phi_1^1\rangle$ and $\mathrm{Var}[\hat{I}_0 \circ \phi_1^{-1}]$ towards their sample equivalents. Interestingly, we see that classical image similarity measures can be used to compare these images effectively.

Due to the approximations described in Section 5.3.2, we cannot expect a perfect match between moment images of model and data samples. Because of this, we find that the L^2-distance often does not result in good matching. Instead, using normalised mutual information, we were able to retrieve the correct values of the variance parameters. However, as normalised mutual information corrects for intensity differences, it is generally invariant to changes in the noise amplitude parameters λ_k. Therefore, we use normalised mutual information to get a good match for the variance parameters before estimating the amplitude parameters using a combination of L^2 distance and unnormalised mutual information.

We let the noise kernels be Gaussian $\sigma_k(\boldsymbol{x}) = \lambda_k \exp\left(\frac{\|\boldsymbol{x}-\mu_k\|^2}{2\tau_k^2}\right) \mathrm{Id}_2$ for simplicity. Hence, the parameters to be estimated consist of the amplitudes $\lambda_k \in \mathbb{R}$ and the variances τ_k^2. Varying the variances τ_k^2 changes the spatial effect of the noise fields while varying the amplitude λ_k affects the intensity of the noise. Let μ_1 denote the sample average, i.e. $\mu_1 = \frac{1}{n}\sum_{i=1}^n I_1^i$. The optimisation procedure first optimise the function,

$$\begin{aligned} f(\{\tau_k\}_{k=1,\dots,p}) = &\mathrm{MI}_{\mathrm{norm}}\left(\hat{I}_0 \circ \langle \phi_1^1 \rangle_{(\tau_k,\lambda_k)},\ \mu_1\right) \\ &+ \mathrm{MI}_{\mathrm{norm}}\left(\mathrm{Var}_{(\tau_k,\lambda_k)}[\hat{I}_0 \circ \phi_1^{-1}],\ \frac{1}{n}\sum_{i=1}^n (I_1^i - \mu_1)^2\right), \end{aligned} \tag{5.10}$$

for the variance parameters τ_k^2 and then optimise the objective function

$$\begin{aligned} g(\{\lambda_k\}_{k=1,\dots,p}) = &\left\| \hat{I}_0 \circ \langle \phi_1^1 \rangle_{(\tau_k,\lambda_k)} - \mu_1 \right\| + \mathrm{MI}\left(\hat{I}_0 \circ \langle \phi_1^1 \rangle_{(\tau_k,\lambda_k)},\ \mu_1\right) \\ &+ \left\| \mathrm{Var}_{(\tau_k,\lambda_k)}[\hat{I}_0 \circ \phi_1^{-1}] - \frac{1}{n}\sum_{i=1}^n (I_1^i - \mu_1)^2 \right\| \\ &+ \mathrm{MI}\left(\mathrm{Var}_{(\tau_k,\lambda_k)}[\hat{I}_0 \circ \phi_1^{-1}],\ \frac{1}{n}\sum_{i=1}^n (I_1^i - \mu_1)^2\right), \end{aligned} \tag{5.11}$$

for the amplitude λ_k.

5.4 Simulation Study

In this section, we present a simulation study aiming at illustrating the ability of the framework to infer parameters given MR brain images. Given an initial 2D MRI image from the OASIS database, two datasets of 200 observations were simulated based on 9 noise fields located in a grid on the image domain.

The first data sample was simulated based on 9 Gaussian noise fields. The location of the noise fields is shown in Fig. 5.2, which also shows the initial brain image, I_0. The variation of the simulated data sample is visualized in Fig. 5.4.

In the simulation study, estimates of the variance parameters τ_k^2 and amplitude λ_k were found for the 9 noise fields. The true value of the standard deviation τ_k was set to 0.06 and the amplitudes $\lambda_k = 2.0$ for all $k = 1, \ldots, 9$.

The initial values of the optimisation procedure were found by investigating the parameter space randomly 40 times picking the values with the smallest objective value. A gradient descent estimation was used with a line search for determining the step size at each iteration.

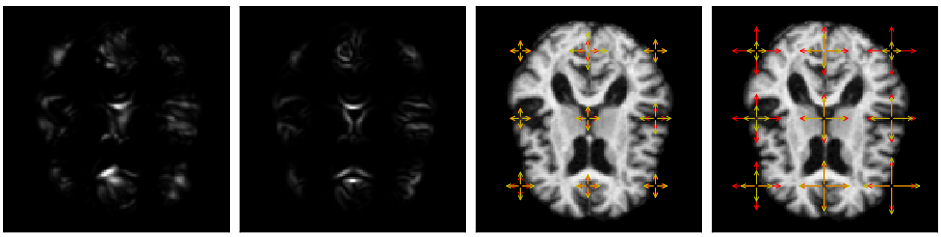


Figure 5.4: *From left:* (1.) Pixel-wise variation in the data. (2.) Variation estimated by the model. (3.) Estimates of the variance parameter τ_k^2 compaired to the true values (4.) Estimates of the variance and amplitude parameter compaired to the true values. The length of the arrows correspond to $\tau_k \lambda_k$. The arrows show the location and width of the noise fields. The red arrows correspond to the true values, while the yellow defines the resulting estimated parameters.

Fig. 5.4 shows the initial brain, I_0, with a comparison of the true values of τ_k and λ_k, and the values found by the optimisation procedure. The red arrows are the true values, and the yellow defines the estimated parameter values. The model is able to retrieve the parameters of τ_k for all $k = 1, \ldots, 9$. It also returns a good estimate of the amplitude parameters, in particular for the noise fields located inside the brain. For noise fields on the boundary of the brain or in the background, the model does not have access to enough information in the intensity differences to return precise estimates of the amplitude parameters.

To give an intuition of convergence of the optimisation, Fig. 5.5 (left) shows the objective function with gradient steps for the optimisation of τ_k in the case of $k = 2$. The amplitude is held fixed in this situation. In the same figure is shown the objective function of the amplitude λ_k when the variance is held fixed.

The second dataset was simulated based on 9 noise fields with larger deviation at $\tau_k = 0.1$ and where the area of variation of each field intersect. The 9 noise fields are shown in Fig. 5.2. The amplitude of the noise fields is again set to $\lambda_k = 2.0$. Retrieving the true values of the parameter vector is harder in this case as the optimisation procedure is more prone to reach a

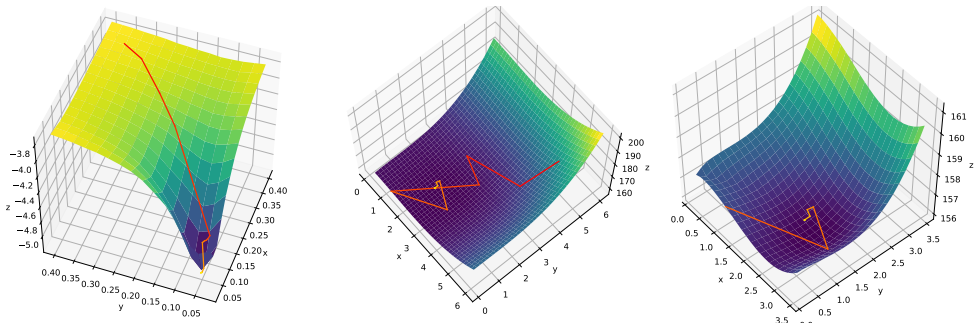


Figure 5.5: *From left:* (1.) Objective function for the optimisation of variance parameters τ_k^2 for an example of 2 noise fields with fixed amplitude parameters. (2.) The objective function for the amplitude λ_k for an example of 2 noise fields with fixed variance parameters. (3.) A zoom of the previous image (2.) around the estimated value.

local minimum. Therefore, we focus this example on estimating the variance parameters τ_k^2. As shown in Fig. 5.6, good estimates for the variance parameters of most of the noise fields are obtained. Estimation of both variance and amplitude with large intersection between the fields is challenging because little information in the data is availble to precisely determine to which of the intersecting noise fields observed variation belong. This could be handled either by imposing a prior on the noise to enforce spatial regularity of the noise amplitudes or by considering noise that is naturally uniform over the domain, e.g. by representing the noise itself in Fourier domain.

Figure 5.6: (left) Data variation. (middle) estimated variation from the model. (right) Estimated variance parameters. The red arrows correspond to the true values, the yellow defines the resulting estimated parameters.

5.5 Conclusion

We presented a model for estimating the variation in medical images occurring from time-continuous deformation variation. The model was based on the stochastic generalisation of the LDDMM framework, using the FLASH procedure to make a natural dimensionality reduction resulting in computationally fast image deformations. Determining the moments of the stochastic flow of the deformations ϕ_t and velocity fields v_t, the method of moments was applied to estimate the parameter vector for noise fields defining the variation in the data sample. The moments of ϕ_t and v_t have been calculated using the Fokker-Planck equation of the evolution of the truncated Fourier expansion of the image deformation. These moments were compared to the data distribution allowing for parameter estimation.

For future work, a natural extension is to define the noise fields in the Fourier domain instead of the spatial fields discussed here.

To calculate the image moments, a coarse Taylor approximation was applied. In future work, we wish to perform a broader investigation of the consequence of making this approximation and whether alternative methods can be used to get a more precise estimation of the image moments.

We disregarded the subject-specific variation and initialised the stochastic deformation by a single image. However, in real data, the subject-specific variation can be large and the model will generally not return a good estimate of data variation when this effect is not taken into account. The model can be extended to include a subject-specific initial image, such that only the variation over time for each individual is modelled and not the total population variation.

Finally, the model has been applied to 2D slices of 3D images. The model is fully general, and since the FLASH framework can handle 3D image data, we wish to include analyses of 3D images in the stochastic framework.

CHAPTER 6

Differential Geometry and Stochastic Dynamics with Deep Learning Numerics

The following chapter includes a manuscript currently under review. The manuscript has been made in collaboration with Alexis Arnaudon and Stefan Sommer and posted on Arxiv as submission,

- L. Kühnel, A. Arnaudon, and S. Sommer. Differential geometry and stochastic dynamics with deep learning numerics. *arXiv 1712.08364*, 2017

The work presents an introduction to the basic theory of Differential Geometry and non-linear statistics. The main objective of the manuscript is to show the application of numerical frameworks, mainly developed for deep learning problems, for concise implementation of the mathematical concepts. The symbolic calculations and automatic differentiation, reduce implementation tasks to a direct translation of mathematical equations. Moreover, even for complex models including for example a numerical integration scheme, Theano can calculate the gradient of the full model in a single function call, making it easy to test new ideas and methods.

Differential Geometry and Stochastic Dynamics with Deep Learning Numerics

Line Kühnel[1], Stefan Sommer[1], and Alexis Arnaudon[2]

Department of Computer Science, University of Copenhagen[1],
Department of Mathematics, Imperial College, London[2]

Abstract

With the emergence of deep learning methods, new computational frameworks have been developed that mix symbolic expressions with efficient numerical computations. In this work, we will demonstrate how deterministic and stochastic dynamics on manifolds, as well as differential geometric constructions can be implemented in these modern frameworks. In particular, we use the symbolic expression and automatic differentiation features of the python library Theano, originally developed for high-performance computations in deep learning. We show how various aspects of differential geometry and Lie group theory, connections, metrics, curvature, left/right invariance, geodesics and parallel transport can be formulated with Theano using the automatic computation of derivatives of any orders. We will also show how symbolic stochastic integrators and concepts from non-linear statistics can be formulated and optimized with only a few lines of code. We will then give explicit examples on low-dimensional classical manifolds for visualization and demonstrate how this approach allows both a concise implementation and efficient scaling to high dimensional problems.

Keywords: Deep Learning Numerics, Theano, Automatic Differentiation, Differential Geometry, Non-linear Statistics

6.1 Introduction

Differential geometry extends standard calculus on Euclidean spaces to non-linear spaces described by a manifold structure, i.e. spaces locally isomorphic to the Euclidean space [75]. This generalisation of calculus turned out to be extremely rich in the study of manifolds and dynamical systems on manifolds. In the first case, being able to compute distances, curvature, and even torsion provides local information on the structure of the space. In the second

case, the question is rather on how to write a dynamical system intrinsically on a nonlinear space, without relying on external constraints from a larger Euclidean space. Although these constructions are general and can be rather abstract, many specific examples of both cases are used for practical applications. We will touch upon such examples later.

Numerical evaluation of quantities such as curvatures and obtaining solutions of nonlinear dynamical systems constitute important problems in applied mathematics. Indeed, high dimensional manifolds or just complicated nonlinear structures make explicit closed-form computations infeasible, even if they remain crucial for applications. The challenge one usually faces is not even in solving the nonlinear equations but in writing them explicitly. Nonlinear structures often consist of several coupled nonlinear equations obtained after multiple differentiations of elementary objects such as nontrivial metrics. In these cases, there is no hope of finding explicit solutions. Instead, the standard solution is to implement the complicated equations in a mathematical software packages such as Matlab or Python using numerical libraries.

In this work, we propose to tackle both issues - being able to solve the equations and being able to implement the equations numerically - by using automatic differentiation software which takes symbolic formulae as input and outputs their numerical solutions. Such libraries in Python includes Theano [127], TensorFlow [1] and PyTorch (`http://pytorch.org`). It is important to stress that these libraries are not symbolic computer algebra packages such as Mathematica or Sympy, as they do not have any symbolic output, but rather numerical evaluations of symbolic inputs. Here, we chose to use Theano but similar codes can be written with other packages with automatic differentiation features. The main interest for us in using Theano is that it is a fully developed package which can handle derivatives of any orders, it has internal compilation and computational graph optimization features that can optimize code for multiple computer architectures (CPU, GPU).

It is the recent surge of interest of deep learning methods that has lead to the development of python libraries such as Theano that mix automatic differentiation with the ability to generate efficient numerical code. The work presented in this paper takes advantage of the significant software engineering efforts to produce robust and efficient libraries for deep learning to benefit a separate domain, computational differential geometry and dynamical systems. We aim to present the use of Theano for these applications in a similar manner as the Julia framework recently presented in [15].

We now wish to give a simple example of Theano code to illustrate this process of symbolic input and numerical output via compiled code. We consider the symbolic implementation of the scalar product, that is the vector function $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y}$, and want to evaluate its derivative with respect to the first argument. In Theano, the function f is first defined as a symbolic

function, `f = lambda x,y: T.dot(x,y)`, where `T` calls functions of the library `theano.tensor`. Then, the gradient of f with respect to x is defined by calling the gradient function `T.grad`, as `df = lambda x,y: T.grad(f(x,y),x)`. Both functions `f` and `df` are still symbolic but can be evaluated on any numerical arrays after the compilation process, or construction of an evaluation function. For our function `f`, the compilation is requested by `ff = theano.function([x,y], f(x,y))`, where we have previously declared the variables x and y as `x = T.vector()` and `y = T.vector()`. The function `ff` is now a compiled version of the function `f` that can be evaluated on any pair of vectors. As we will see later in the text, such code can be written for many different functions and combination of derivatives, in particular for derivatives with respect to initial conditions in a `for`-loop.

One aim is to illustrate this transparent use of Theano in various numerical computations based on objects from differential geometry. We will only cover a few topics and many other such applications will remain for future works. Apart from our running example, the sphere, or the rotation group, we will use higher dimensional examples, in particular the manifold of landmarks as often used in computational anatomy. In both cases, we will show how to compute various geometrical quantities arising from Riemannian metrics on the spaces. In most cases, the metric is the only information on the manifold that is needed, and it allows for computing geodesics, Brownian motion, parallel transport etc. In some cases, it will be convenient to extend to computations in a fiber bundle of the manifold to have more freedom and allow for example anisotropic diffusion processes. Also, when the manifold has a group structure, we can perform for example reduction by symmetry for dynamical systems invariant under the group action. All of these mechanical constructions can be used to real-world applications such as in control or robotics. We refer to the books [19, 26, 27] for more theories and applications in these directions. We will not directly consider these applications here, but rather focus on applications of computational anatomy. We refer the interested reader to the book [135] and references therein for a good overview of this topic. We also refer to the conference paper [65] for a short introduction of the use of Theano in computational anatomy. Computational anatomy is a vast topic, and we will only focus here on a few aspects when shapes or images are represented as sets of points, or landmarks, that are used as tracers of the original shape. With these landmarks, we show how many algorithms related to matching of shapes, statistics of shapes or random deformations, can be implemented concisely and efficiently using Theano.

As an example, we display in Figure 6.1 two examples of solving the inverse problem of estimating the initial momenta for a geodesic matching landmark configurations on high-dimensional manifolds of landmarks on the plane. On the left panel of Figure 6.1, we solved the problem of matching a letter 'T' to a letter 'O', or more precisely an ellipse, with 2,500 land-

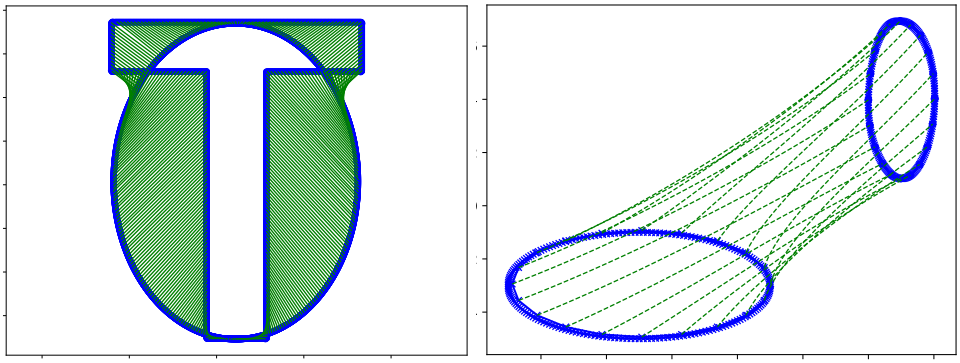


Figure 6.1: (left) Matching of 2,500 landmarks on the outline of a letter 'T' to a letter 'O'. The matching is performed by computing the logarithm map Log considering the 5,000 dimensional landmark space a Riemannian manifold. (right) Similar matching of landmark configurations using Log while now using the transparent GPU features of Theano to scale to configurations with 20,000 landmarks on a 40,000 dimensional manifold. Theano generates highly efficient numerical code and allows GPU acceleration transparently to the programmer. For both matches, only a subset of the geodesic landmark trajectories are display.

marks. On the right panel, we solved the problem of matching two simple shapes, ellipses, however with 20,000 landmarks. The shapes represented by landmarks are considered elements of the LDDMM landmark manifold of dimension 5,000 and 40,000, see [135]. The geodesics equation and inverse problem are implemented using the few lines of code presented in this paper and the computation is transparently performed on GPUs.

Parts of the code will be shown throughout the paper with corresponding examples. The full code is available online in the Theano Geometry repository `http://bitbucket.org/stefansommer/theanogeometry`. The interested reader can find a more extensive description of the mathematical notions used in this paper in the books *Riemannian Manifolds: an introduction to curvature* by J. Lee [76], *Stochastic Analysis on Manifolds* by E. P. Hsu [52] and *Introduction to Mechanics and Symmetry* by Marsden, Ratiu [84].

Content of the paper

The paper will be structured as follows. Section 6.2 gives an account of how central concepts in Riemannian geometry can be described symbolically in Theano, including the exponential and logarithm maps, geodesics in Hamiltonian form, parallel transport and curvature. Concepts from Lie group theory are covered in section 6.3, and section 6.4 continues with sub-Riemannian frame bundle geometry. In addition to the running example of surfaces em-

bedded in $\mathbb{R}^3$, we will show in section 6.5 applications on landmark manifolds defined in the LDDMM framework. At the end, concepts from nonlinear statistics are covered in section 6.6.

6.2 Riemannian Geometry

In this section, we will show how to implement some of the theoretical concepts from Riemannian geometry. This includes geodesics equation, parallel transport and curvature. The focus is to present simple and efficient implementation of these concepts using Theano [127].

Though the code applies to any smooth manifolds $\mathcal{M}$ of dimension d, we will only visualize the results of numerical computations on manifolds embedded in $\mathbb{R}^3$. We represent these manifolds by a smooth injective map $F : \mathbb{R}^2 \to \mathbb{R}^3$ and the associated metric on $\mathcal{M}$ inherited from $\mathbb{R}^3$, that is

$$g = (dF)^T dF \,, \tag{6.1}$$

where dF denotes the Jacobian of F. One example of such representation is the sphere S^2 in stereographic coordinates. In this case, $F \colon \mathbb{R}^2 \to S^2 \subset \mathbb{R}^3$ is

$$F(x,y) = \begin{pmatrix} \frac{2x}{1+x^2+y^2} & \frac{2y}{1+x^2+y^2} & \frac{-1+x^2+y^2}{1+x^2+y^2} \end{pmatrix} . \tag{6.2}$$

6.2.1 Geodesic Equation

We begin by computing solutions to the Riemannian geodesic equations on a smooth d-dimensional manifold $\mathcal{M}$ equipped with an affine connection ∇ and a metric g. A connection on a manifold defines the relation between tangent spaces at different points on $\mathcal{M}$. Let (U, φ) denote a local chart on $\mathcal{M}$ with coordinate basis $\partial_i = \frac{\partial}{\partial x^i}$, $i = 1, \ldots, d$. The connection ∇ is related to the Christoffel symbols Γ_{ij}^k by the relation

$$\nabla_{\partial_i} \partial_j = \Gamma_{ij}^k \partial_k \,. \tag{6.3}$$

An example of a frequently used connection on a Riemannian manifold $(\mathcal{M}, g)$ is the Levi-Civita connection. The Christoffel symbols for the Levi-Civita connection is uniquely defined by the metric g. Let g_{ij} denote the coefficients of the metric g, i.e. $g = g_{ij} dx^i dx^j$, and g^{ij} be the inverse of g_{ij}. The Christoffel symbols for the Levi-Civita connection are then

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl} (\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}) \,. \tag{6.4}$$

The implementation of the Christoffel symbols in Theano are shown in the code snippet below.

```
"""
Christoffel symbols for the Levi-Civita connection

Args:
    x: Point on the manifold
    g(x): metric g evaluated at position x on the manifold.

Returns:
    Gamma_g: 3-tensor with dimensions k,i,j in the respective
        order
"""
# Derivative of metric:
Dg = lambda x: T.jacobian(g(x).flatten(),x).reshape((d,d,d))
# Inverse metric (cometric):
gsharp = lambda x: T.nlinalg.matrix_inverse(g(x))
# Christoffel symbols:
Gamma_g = lambda x: 0.5*(T.tensordot(gsharp(x),Dg(x),axes =
    [1,0])\
      +T.tensordot(gsharp(x),Dg(x),axes = [1,0]).dimshuffle
          (0,2,1)\
      -T.tensordot(gsharp(x),Dg(x),axes = [1,2]))
```

Straight lines in $\mathbb{R}^n$ are lines with no acceleration and path minimizers between two points. Geodesics on a manifold are defined in a similar manner. The acceleration of a geodesic γ is zero, i.e. $D_t\dot{\gamma} = 0$, in which D_t denotes the covariant derivative. Moreover, geodesics determines the shortest distances between points on $\mathcal{M}$. Let $x_0 \in \mathcal{M}$, (U, φ) be a chart around x_0 and consider $v_0 \in T_{x_0}\mathcal{M}$, a tangent vector at x_0. A geodesic $\gamma\colon I \to \mathcal{M}$, $I = [0, 1]$, $\gamma_t = (x_t^i)_{i=1,\dots,d}$, satisfying $\gamma_0 = x_0$, $\dot{\gamma}_0 = v_0$ can be obtained by solving the geodesic equations

$$\ddot{x}_t^k + \dot{x}_t^i \dot{x}_t^j \Gamma_{ij}^k(\gamma_t) = 0\,. \tag{6.5}$$

The goal is to solve this second order ordinary differential equation (ODE) with respect to x_t^k. We first rewrite the ODE in term of $w_t^k = \dot{x}_t^k$ and x_t^k to instead have a system of first order ODE of the form

$$\dot{w}_t^k = -w_t^i w_t^j \Gamma_{ij}^k(\gamma_t)\,,\ \dot{x}_t^k = w_t^k\,,$$

which can be solved by numerical integration. For this, we can use the Euler method

$$y_{n+1} = y_n + f(t_n, y_n)\Delta t, \quad \Delta t = t_{n+1} - t_n\,, \tag{6.6}$$

or by higher-order integrators such as a fourth-order Runge-Kutta method. Both integrators are available in symbolic form in the code repository. In Theano, we use the symbolic `for`-loop `theano.scan` for the loop over time-steps. As a consequence, symbolic derivatives of the numerical integrator can be evaluated. For example, we will later use derivatives with respect to

the initial values when solving the geodesic matching problem in the definition of the Logarithm map. In addition, it is possible to solve stochastic differential equations in a similar way, see Appendix 6.A. The symbolic implementation of the integrator method is shown below.

```
"""
Numerical Integration Method

Args:
   ode: Symbolic ode function to be solved
   integrator: Integration scheme (Euler, RK4, ...)
   x: Initial values of variables to be updated by
      integration method
   *y: Additional variables for ode.

Returns:
   Tensor (t,xt)
           t: Time evolution
           xt: Evolution of x
"""
def integrate(ode,integrator,x,*y):
   (cout, updates) = theano.scan(fn=integrator(ode),
         outputs_info=[T.constant(0.),x],
         sequences=[*y],
         n_steps=n_steps)
   return cout
```

Based on the symbolic implementation of the integrators, solutions to the geodesic equations are obtained by the following code.

```
"""
Geodesic Equation

Args:
   xq: Tensor with x and xdot components.

Returns:
   ode_geodesic: Tensor (dx,dxdot).
   geodesic: Tensor (t,xt)
              t: Time evolution
              xt: Geodesic path
"""
def ode_geodesic(t,xq):
   dxdott = - T.tensordot(T.tensordot(xq[1], Gamma_g(xq[0]),
      axes=[0,1]),
                       xq[1],axes=[1,0])
   dxt = xq[1]
   return T.stack((dxt,dxdott))
# Geodesic:
geodesic = lambda x,xdot: integrate(ode_geodesic, T.stack((x,
   xdot)))
```

Figure 6.2 shows examples of geodesics on three different manifolds obtained as the solution to the geodesic equations in (6.5) using the above code.

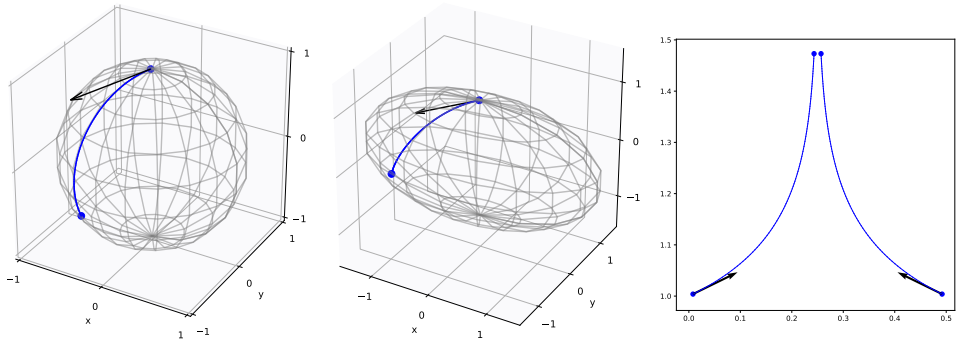


Figure 6.2: The solution of the geodesic equations for three different manifolds; the sphere S^2, an ellipsoid, and landmark manifold defined in the LDDMM framework. The arrows symbolizes the initial tangent vector v_0.

6.2.2 The Exponential and Logarithm Maps

For a geodesic γ_t^v, $t \in [0,1]$ with initial velocity $\dot{\gamma}_0^v = v$, the exponential map, $\text{Exp}_x : T_x\mathcal{M} \to \mathcal{M}$, $x \in \mathcal{M}$ is defined by

$$\text{Exp}_x(v) = \gamma_1^v \,, \tag{6.7}$$

and can be numericaly computed from the earlier presented geodesic equation.

```
"""
Exponential map

Args:
    x: Initial point of geodesic
    v: Velocity vector

Returns:
    y: Endpoint of geodesic
"""
Exp = lambda x,v: geodesic(x,v)[1][-1,0]
```

Where defined, the inverse of the exponential map is denoted the logarithm map. For computational purposes, we can define the logarithm map as finding a minimizing geodesic between $x_1, x_2 \in \mathcal{M}$, that is

$$\text{Log}(x_1, x_2) = \arg\min_v \|\text{Exp}_{x_1}(v) - x_2\|^2_{\mathcal{M}} \,, \tag{6.8}$$

for a norm coming for example from the embedding of $\mathcal{M}$ in $\mathbb{R}^3$. From the logarithm, we also get the geodesic distance by

$$d(x, y) = \|\mathrm{Log}(x, y)\| . \tag{6.9}$$

The logarithm map can be implemented in Theano by using the symbolic calculations of derivatives by computing the gradient of the loss function (6.8) with Theano function `T.grad`, and then use it in a standard minimisation algorithm such as BFGS. An example implementation is given below, where we used the function `minimize` from the Scipy package.

```
"""
Logarithm map

Args:
   v0: Initial tangent vector
   x1: Initial point for geodesic
   x2: Target point on the manifold

Return:
   Log: Tangent vector
"""
# Loss function:
loss = lambda v,x1,x2: 1./d*T.sum(T.sqr(Exp(x1,v)-x2))
dloss = lambda v,x1,x2: T.grad(loss(v,x1,x2),v)
# Logarithm map: (v0 initial guess)
Log = minimize(loss, v0, jac=dloss, args=(x1,x2))
```

6.2.3 Geodesics in Hamiltonian Form

In section 6.2.1, geodesics were computed as solutions to the standard second order geodesic equations. We now compute geodesics from a Hamiltonian viewpoint. Let the manifold $\mathcal{M}$ be equipped with a cometric g^* and consider a connection ∇ on $\mathcal{M}$. Given a point $x \in \mathcal{M}$ and a covector $p \in T_x^*\mathcal{M}$, geodesics can be obtained as the solution to Hamilton's equations, given by the derivative of the Hamiltonian, which in our case is

$$H(x, p) = \frac{1}{2}\langle p, g_x^*(p)\rangle_{T_x^*\mathcal{M}\times T_x^*\mathcal{M}} . \tag{6.10}$$

Hamilton's equations are then

$$\begin{aligned} \frac{d}{dt}x &= \nabla_p H(x, p) \\ \frac{d}{dt}p &= -\nabla_x H(x, p) , \end{aligned} \tag{6.11}$$

and describe the movement of a particle at position $x \in \mathcal{M}$ with momentum $p \in T_x^*\mathcal{M}$.

Depending on the form of the Hamiltonian and in particular of the metric, the implementation of Hamilton's equations (6.11) can be difficult. In the present case, the metric on $\mathcal{M}$ is inherited from an embedding F, hence g^* is defined only via derivatives of F, which makes the computation possible with Theano.

```
"""
Calculate the Exponential map defined by Hamilton's equations

Args:
   x: Point on manifold
   p: Momentum vector at x
   gsharp(x): Matrix representation of the cometric at x

Returns:
   Exp: Tensor (t,xt)
            t: Time evolution
            xt: Geodesic path
"""
# Hamiltonian:
H = lambda x,p: 0.5*T.dot(p,T.dot(gsharp(x),p))

# Hamilton's equation
dx = lambda x,p: T.grad(H(x,p),p)
dp = lambda x,p: -T.grad(H(x,p),x)
def ode_Hamiltonian(t,x):
   dxt = dx(x[0],x[1])
   dpt = dp(x[0],x[1])
   return T.stack((dxt,dpt))

# Geodesic:
Exp = lambda x,v: integrate(ode_Ham,T.stack((x,g(v))))
```

Calculating geodesics on a Riemannian manifold $\mathcal{M}$ by solving Hamilton's equations can be generalized to manifolds for which only a sub-Riemannian structure is available. An example of such geodesics is given in section 6.4 on a different construction, the frame bundle.

Example 6.2.1 (Geodesic on the sphere). *Consider the sphere $S^2 \subset \mathbb{R}^3$ in stereographic coordinates such that for $(x, y) \in \mathbb{R}^2$, a point on the sphere is given by $F(x, y)$ with F defined in (6.2). Equip S^2 with the metric g defined in (6.1) and let $x_0 = F(0, 0) \in S^2$ and $v_0 = dF(1, -1) \in T_{x_0}S^2$. The initial momentum vector is chosen as the corresponding covector of v_0 defined by the flat map $\flat \colon T\mathcal{M} \to T^*\mathcal{M}$, i.e. $p_0 = v_0^\flat$. The geodesic, or the solution to Hamilton's equations can be seen in the left plot of Figure 6.3.*

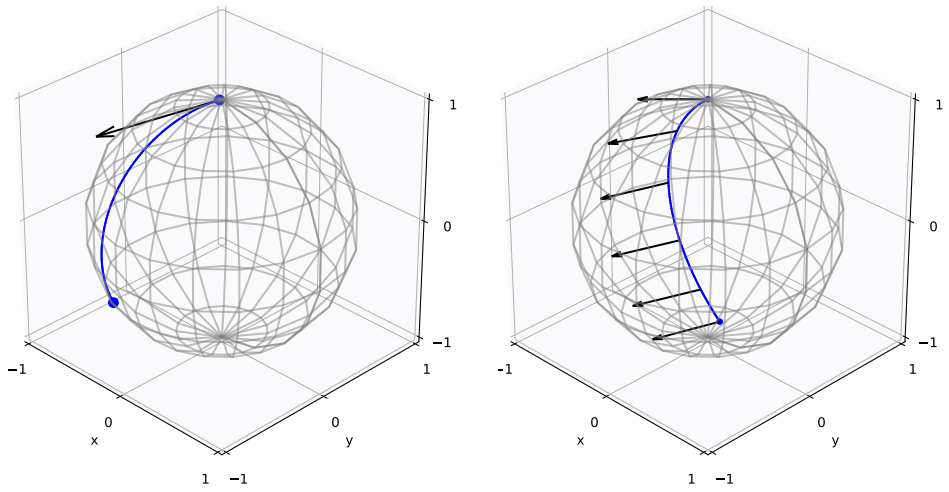


Figure 6.3: (left) Geodesic defined by the solution to Hamilton's equations (6.11) with initial point $x_0 = F(0,0) \in S^2$ and velocity $v_0 = dF(1,-1) \in T_{x_0}S^2$. See Example 6.2.1. (right) Parallel transport of vector $v = dF\left(-\frac{1}{2}, -\frac{1}{2}\right)$ along the curve $\gamma_t = F(t^2, -\sin(t))$. See Example 6.2.2.

6.2.4 Parallel Transport

Let again $\mathcal{M}$ be a d-dimensional manifold with an affine connection ∇ and let (U, φ) denote a local chart on $\mathcal{M}$ with coordinate basis $\partial_i = \frac{\partial}{\partial x^i}$ for $i = 1, \dots, d$. A vector field V along a curve γ_t, is said to be parallel if the covariant derivative of V along γ_t is zero, i.e. $\nabla_{\dot{\gamma}_t} V = 0$. It can be shown that given a curve $\gamma \colon I \to \mathcal{M}$ and a tangent vector $v \in T_{\gamma_{t_0}}\mathcal{M}$ there exists a unique parallel vector field V along γ such that $V_{t_0} = v$. We further assume that $\gamma_t = (\gamma_t^i)_{i=1,\dots,d}$ in local coordinates and we let $V_t = v^i(t)\partial_i$ be a vector field. V is then parallel to the curve γ_t if the coefficients $v^i(t)$ solve the following differential equation,

$$\dot{v}^k(t) + \Gamma_{ij}^k(\gamma_t)\dot{\gamma}_t^i v^j(t) = 0\,. \tag{6.12}$$

The parallel transport can be implemented in an almost similar manner as the geodesic equations introduced in section 6.2.1.

```
"""
Parallel Transport

Args:
    gamma: Discretized curve
    dgamma: Tangent vector of gamma to each time point
    v: Tangent vector that will be parallel transported.
```

```
Returns:
    pt: Tensor (t,vt)
            t: Time Evolution
            vt: Parallel transported tangent vector at each
                time point
"""
def ode_partrans(gamma,dgamma,t,v):
    dpt = - T.tensordot(T.tensordot(dgamma, Gamma_g(gamma),
        axes = [0,1]),
                        v, axes = [1,0])
    return dpt

# Parallel transport
pt = lambda v,gamma,dgamma: integrate(ode_partrans,v,gamma,
    dgamma)
```

Example 6.2.2. *In this example, we consider a tangent vector* $v = dF\left(-\frac{1}{2}, -\frac{1}{2}\right) \in T_x S^2$ *for* $x = F(0,0) \in S^2$ *that we want to parallel transport along the curve* $\gamma\colon [0,1] \to S^2$ *given by* $\gamma_t = F\left(t^2, -\sin(t)\right)$. *The solution of the problem is illustrated in the right panel of Figure 6.3.*

6.2.5 Curvature

The curvature of a Riemannian manifold $\mathcal{M}$ is described by the Riemannian curvature tensor, a $(3,1)$-tensor $R\colon \mathcal{T}(\mathcal{M}) \times \mathcal{T}(\mathcal{M}) \times \mathcal{T}(\mathcal{M}) \to \mathcal{T}(\mathcal{M})$ defined as

$$R(X,Y)Z = \nabla_X \nabla_Y Z - \nabla_Y \nabla_X Z - \nabla_{[X,Y]} Z\,. \tag{6.13}$$

Let (U,φ) be a local chart on $\mathcal{M}$ and let ∂_i for $i = 1,\ldots,d$ denote the local coordinate basis with dx^i being the dual basis. Given this local basis, the curvature tensor is, in coordinates, given as

$$R = R_{ijk}{}^m dx^i \otimes dx^j \otimes dx^k \otimes \partial_m\,, \tag{6.14}$$

where the components $R_{ijk}{}^m$ depend on the Christoffel symbols as follow

$$R(\partial_i,\partial_j)\partial_k = R_{ijk}{}^m \partial_m = (\Gamma^l{}_{jk}\Gamma^m{}_{il} - \Gamma^l{}_{ik}\Gamma^m{}_{jl} + \partial_i \Gamma^m{}_{jk} - \partial_j \Gamma^m{}_{ik})\partial_m\,. \tag{6.15}$$

In Theano, the Riemannian curvature tensor can be computed in coordinates as follow.

```
"""
Riemannian curvature tensor in coordinates

Args:
    x: point on manifold

Returns:
```

```
    4-tensor R_ijk^m in with order i,j,k,m
"""
def R(x):
 return (T.tensordot(Gamma_g(x),Gamma_g(x),(0,2)).dimshuffle
     (3,0,1,2)
      - T.tensordot(Gamma_g(x),Gamma_g(x),(0,2)).dimshuffle
          (0,3,1,2)
      + T.jacobian(Gamma_g(x).flatten(),x).reshape((d,d,d,d)).
          dimshuffle(3,1,2,0)
      - T.jacobian(Gamma_g(x).flatten(),x).reshape((d,d,d,d)).
          dimshuffle(1,3,2,0))
```

In addition to the curvature tensor $R_{ijk}{}^{m}$, the Ricci and scalar curvature can be computed by contracting the indices as

$$R_{ij} = R_{kij}{}^{k} \ , \ S = g^{ij} R_{ij} \, . \tag{6.16}$$

The sectional curvature can also be computed and describes the curvature of a Riemannian manifold by the curvature of a two-dimensional submanifold. Let Π be a two-dimensional sub-plane of the tangent space at a point $x \in \mathcal{M}$. Let e_1, e_2 be two linearly independent tangent vectors spanning Π. The sectional curvature is the Gaussian curvature of the sub-space formed by geodesics passing x and tangent to Π, that is

$$\kappa(e_1, e_2) = \frac{\langle R(e_1, e_2)e_2, e_1 \rangle}{\|e_1\|^2 \|e_2\|^2 - \langle e_1, e_2 \rangle^2} \, . \tag{6.17}$$

Example 6.2.3 (Curvature of S^2)**.** *We consider $x = F(0,0) \in S^2$ and the orthonormal basis vectors $e_1 = dF(0.5, 0)$, $e_2 = dF(0, 0.5)$ in the tangent space $T_x\mathcal{M}$ with respect to the metric g. As expected, we found that the Gaussian curvature of S^2 is 1 and its scalar curvature is 2 [76].*

The Ricci, scalar and sectional curvature have also been implemented in Theano as follow.

```
"""
Different curvature measures

Args:
    x: point on manifold
    e1, e2: linearly independent tangent vectors
"""
# Ricci curvature:
Ricci_curv = lambda x: T.tensordot(R(x),T.eye(d),((0,3),(0,1)
    ))
# Scalar curvature:
S_curv = lambda x: T.tensordot(Ricci_curv(x),gsharp(x),((0,1)
    ,(0,1)))
# Sectional curvature:
```

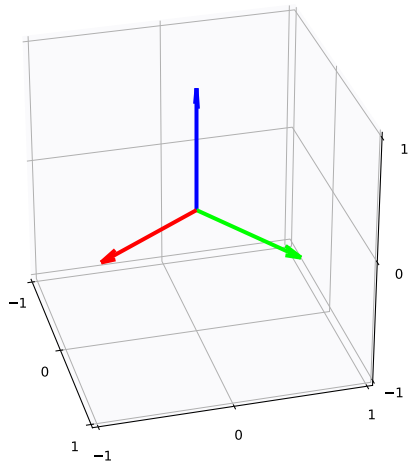


Figure 6.4: We show an example of an element of $\mathrm{SO}(3)$ represented as a matrix $g \in \mathbb{R}^{3\times 3}$. The vectors represent each column of g.

```
def sec_curv(x,e1,e2):
      Rflat = T.tensordot(R(x),g(x),[3,0])
      sec = T.tensordot(
            T.tensordot(
               T.tensordot(
                  T.tensordot(Rflat, e1, [0,0]),
                  e2, [0,0]),
               e2, [0,0]),
            e1, [0,0])
      return sec
```

6.3 Dynamics on Lie Groups

In this section, we consider a manifold equipped with a smooth group structure, that is $\mathcal{M} = G$ is a Lie group. As the most interesting finite dimensional Lie groups are isomorphic to matrix groups, we can without loss of generalities represent elements of Lie group G by matrices. We will give examples of how various fundamental Lie group constructions can be written with Theano and how to compute geodesics in the Hamiltonian and Lagrangian setting. We will mostly follow [84] for the notation and definitions. We will use $G = \mathrm{SO}(3)$, the three dimensional rotation group acting on $\mathbb{R}^3$ as an illustration, where an element of G is represented by a coordinate basis as for example in Figure 6.4.

The group operation on G defines the left and right translation maps $L_a(g) = ag$ and $R_a(g) = ga$ for $a, g \in G$. As elements of G are represented

by matrices, these maps are in Theano computed by matrix multiplications. Their corresponding tangent maps dL and dR can be directly obtained by taking symbolic derivatives. The left and right translation maps relate elements of the Lie algebra $\mathfrak{g}$ of the group with the left (and right) invariant vector fields $X_\eta(g) := dL_g(\eta)$ on TG, where $\eta \in \mathfrak{g}$. The algebra structure on $\mathfrak{g}$ is then defined from the Jacobi-Lie bracket of vector fields $[\xi, \eta] = [X_\xi, X_\eta]$, $\xi, \eta \in \mathfrak{g}$.

Using invariance under the group action, either left or right, an inner product on $\mathfrak{g} = T_eG$ can be extended to a Riemannian metric on G by setting $\langle v, w\rangle_g = \langle dL_a v, dL_a w\rangle_{L_a(g)}$ for $v, w \in T_gG$. Invariant metrics can thus be identified with a symmetric positive definite inner product $\langle \cdot, \cdot \rangle_A$ on $\mathfrak{g}$, where after fixing a basis for $\mathfrak{g}$, we can consider that $A \in \mathrm{Sym}^+(\mathfrak{g})$ and $\langle \cdot, \cdot \rangle_A = \langle \cdot, A \cdot \rangle$. Hence, A^{-1} is the corresponding co-metric.

In Theano, these constructions can be formulated as shown below. A basis e_i for $\mathfrak{g}$ is fixed, and `LAtoV` is the inverse of the mapping $v \to e_i v^i$ between $V = \mathbb{R}^d$ and the Lie algebra $\mathfrak{g}$.

```
"""
General functions for Lie groups

Args:
   g,h: Elements of G
   v: Tangent vector
   xi,eta: Elements of the Lie Algebra
   d: Dimension of G
   vg,wg: Elements of tangent space at g
"""

L = lambda g,h: T.tensordot(g,h,(1,0)) # left translation L_g
    (h)=gh
R = lambda g,h: T.tensordot(h,g,(1,0)) # right translation
    R_g(h)=hg

# Derivative of L
def dL(g,h,v):
   dL = T.jacobian(L(theano.gradient.disconnected_grad(g),h).
       flatten(),
                   h).reshape((N,N,N,N))
   return T.tensordot(dL,v,((2,3),(0,1)))

# Lie bracket
def bracket(xi,eta):
   return T.tensordot(xi,eta,(1,0))-T.tensordot(eta,xi,(1,0))

# Left-invariant metric
def g(g,v,w):
   xiv = dL(inv(g),g,v)
   xiw = dL(inv(g),g,w)
```

```
    v = LAtoV(xiv)
    w = LAtoV(xiw)
    return T.dot(v,T.dot(A,w))
```

6.3.1 Euler-Poincaré Dynamics

In the context of Lie groups, we can also derive the geodesic equations for a left-invariant metric. Geodesics on the Lie group can, similar to geodesics on manifolds defined in section 6.2.3, be described as solutions to Hamilton's equations for a Hamiltonian generated from the left-invariant metric. In this section, we will, however, present another method for calculating geodesics based on the Euler-Poincaré equations.

The conjugation map $h \mapsto aha^{-1}$ for fixed $a \in G$ has as a derivative the adjoint map $\mathrm{Ad}(a) : \mathfrak{g} \to \mathfrak{g}$, $\mathrm{Ad}(a)X = (L_a)_*(R_{a^{-1}})_*(X)$. The derivative of Ad with respect to a is the Lie bracket $\mathrm{ad}_\xi : \mathfrak{g} \to \mathfrak{g}$, $\mathrm{ad}_\xi(\eta) = [\xi, \eta]$. The coadjoint action is defined by $\left\langle \mathrm{ad}^*_\xi(\alpha), \eta \right\rangle = \langle \alpha, \mathrm{ad}_\xi(\eta) \rangle$, $\alpha \in \mathfrak{g}^*$ with $\langle \cdot, \cdot \rangle$ the standard pairing on the Lie algebra $\mathfrak{g}$. For the kinetic Lagrangian $l(\xi) = \xi^T A \xi$, $\xi \in \mathfrak{g}$, a geodesic is a solution of the Euler-Poincaré equation

$$\partial_t \frac{\delta l}{\delta \xi} = \mathrm{ad}^*_\xi \frac{\delta l}{\delta \xi}\,, \tag{6.18}$$

together with the reconstruction equation $\partial_t g_t = g_t \xi_t$. This relatively abstract set of equations can be expressed in Theano with the following code.

```
"""
Euler-Poincare Geodesic Equations

Args:
    a,g: Element of G
    xi,eta: Element of Lie Algebra
    p,pp,mu: Elements of the dual Lie Algebra

Returns:
    EPrec: Tensor (t,xt)
              t: Time evolution
              gt: Geodesic path in G
"""
# Adjoint functions:
Ad = lambda a,xi: dR(inv(a),a,dL(a,e,xi))
ad = lambda xi,eta: bracket(xi,eta)
coad = lambda p,pp: T.tensordot(T.tensordot(C,p,(0,0)),pp
    ,(1,0))

# Euler-Poincare equations:
def ode_EP(t,mu):
    xi = T.tensordot(inv(A),mu,(1,0))
    dmut = -coad(xi,mu)
```

```
    return dmut
EP = lambda mu: integrate(ode_EP,mu)

# reconstruction
def ode_EPrec(mu,t,g):
    xi = T.tensordot(inv(A),mu,(1,0))
    dgt = dL(g,e,VtoLA(xi))
    return dgt
EPrec = lambda g,mus: integrate(ode_EPrec,g,mus)
```

Example 6.3.1 (Geodesic on $SO(3)$)**.** *Let $g_0 \in G$ be the identity matrix. An example of a geodesic on* SO(3) *found as the solution to the Euler-Poincaré equation is shown in Figure 6.5.*

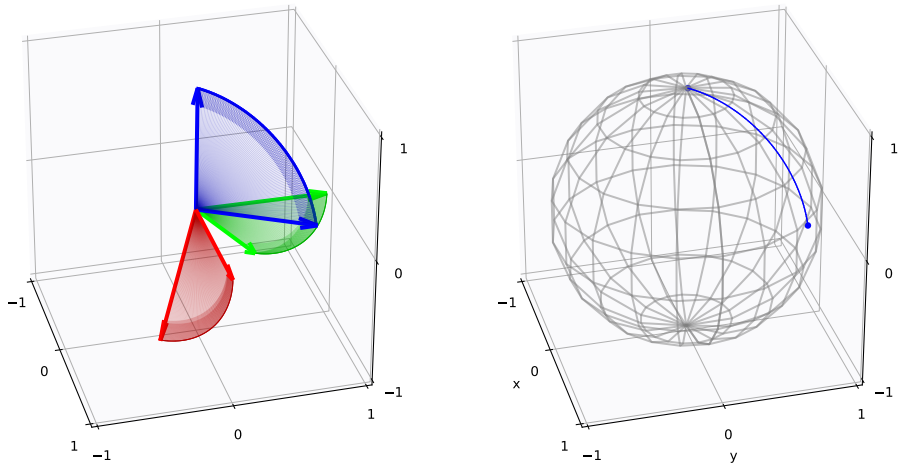


Figure 6.5: (left) Geodesic on SO(3) found by the Euler-Poincaré equations. (right) The geodesic on SO(3) projected to the sphere using the left action $g.x = gx$ for $x \in S^2 \subset \mathbb{R}^3$.

6.3.2 Brownian motion on G

In the following subsection, we will go through a construction of Brownian motions on a group G where the evolution is given as a Stratonovich SDE. With a group structure, we can simulate a Brownian motion which remains in the group G. Using the inner product A, let $e_1, \ldots, e_d$ be an orthonormal basis for $\mathfrak{g}$, and construct an orthonormal set of vector fields on the group as $X_i(g) = dL_g e_i$, for $g \in G$. Recall that the structure constant of the Lie algebra C^i_{jk} are the same as in the commutator of these vector fields, that is

$$[X_j, X_k] = C^i_{jk} X_i \,. \tag{6.19}$$

The corresponding Brownian motion on G is the following Stratonovich SDE

$$dg_t = -\frac{1}{2}\sum_{j,i} C^j{}_{ij} X_i(g_t)dt + X_i(g_t) \circ dW_t^i\,, \tag{6.20}$$

where W_t is an $\mathbb{R}^d$-valued Wiener processes. We refer to [77] for more information on Brownian motions on Lie groups.

In Theano, the stochastic process (6.20) can be integrated with the following code.

```
"""
SDE for Brownian Motions on a Lie group G

Args:
    g: Starting point for the process
    dW: Steps of a Euclidean Brownian motion

Returns:
    Tensor (t,gt)
        t: Time evolution
        gt: Evolution of g
"""
def sde_Brownian(dW,t,g):
    X = T.tensordot(dL(g,e,eiLA),sigma,(2,0))
    det = -.5*T.tensordot(T.diagonal(C,0,2).sum(1),X,(0,2))
    sto = T.tensordot(X,dW,(2,0))
    return (det,sto)
Brownian = lambda g,dWt: integrate_sde(sde_Brownian,
    integrator_stratonovich,g,dWt)
```

Here, we used `integrate_sde` which is a discrete time stochastic integrator as described in section 6.A.2.

Example 6.3.2 (Brownian motion on $SO(3)$)**.** *Figure 6.6 shows an example of a Brownian motion on* $\mathrm{SO}(3)$*. The initial point* $x_0 \in \mathrm{SO}(3)$ *for the Brownian motion was the 3-dimensional identity matrix.*

There are other ways of defining stochastic processes on a Lie group G. An example can be found in [3] for finite dimensional Lie groups and in [49] for infinite dimensions. See also [30] for the general derivation of these stochastic equations. In this theory, the noise is introduced in the reconstruction relation to form the motion on the dual of the Lie algebra to the Lie group and appears in the momentum formulation of the Euler-Poincaré equation given in (6.18). This framework has also been implemented in Theano and can be found in the repository. Another interesting approach, not yet implemented in Theano, is the one of [7], where noise is introduced on the Lie group, and an expected reduction by symmetries results in a dissipative deterministic Euler-Poincaré equation.

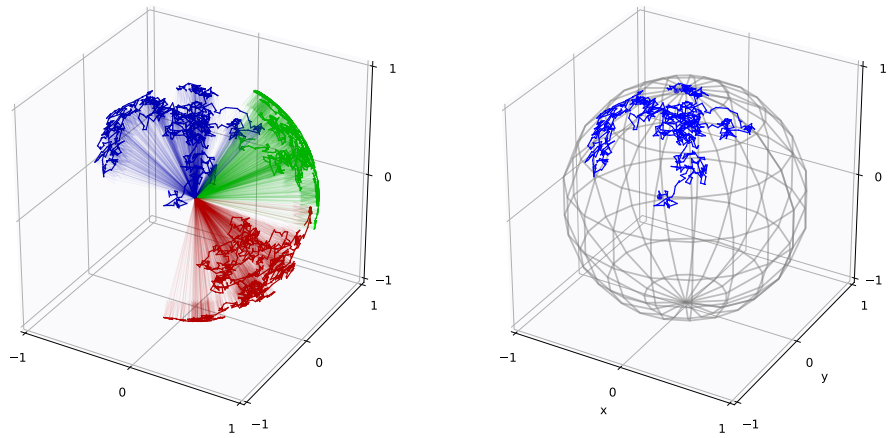


Figure 6.6: (left) Brownian motion on the group $\mathrm{SO}(3)$ defined by (6.20). The initial point, $x_0 \in \mathrm{SO}(3)$, was set to the identity matrix. (right) The projection by the left action of the Brownian motion on $\mathrm{SO}(3)$ to the sphere

6.4 Sub-Riemannian Frame Bundle Geometry

We now consider dynamical equations on a more complicated geometric construction, a frame bundle or more generally fibre bundles. A frame bundle $F\mathcal{M} = \{F_x\mathcal{M}\}_{x\in\mathcal{M}}$ is the union of the spaces $F_x\mathcal{M}$, the frames of the tangent space at $x \in \mathcal{M}$. A frame $\nu\colon \mathbb{R}^d \to T_x\mathcal{M}$ is thus an ordered basis for the tangent space $T_x\mathcal{M}$. The frame bundle $F\mathcal{M}$ is a fibre bundle $\pi : FM \to M$ with projection π and can be equipped with a natural sub-Riemannian structure induced by the metric g on $\mathcal{M}$ [89]. Given a connection on $\mathcal{M}$ the tangent space $TF\mathcal{M}$ can be split into a horizontal and vertical subspace, $HF\mathcal{M}$ and $VF\mathcal{M}$, i.e. $TF\mathcal{M} = HF\mathcal{M} \oplus VF\mathcal{M}$. Consider a local trivialization $u = (x, \nu)$ of $F\mathcal{M}$ so that $\pi(u) = x$. A path $u_t = (x_t, \nu_t)$ on $F\mathcal{M}$ is horizontal if $\dot{u}_t \in HFM$ for all t. A horizontal motion of u_t corresponds to a parallel transport of the frame along the curve $\pi(u_t)$ on $\mathcal{M}$. Consequently, the parallel transport ν_t of a frame ν_0 of $T_{x_0}\mathcal{M}$ along a curve x_t on $\mathcal{M}$ is called a horizontal lift of x_t.

Let $\partial_i = \frac{\partial}{\partial x^i}$, $i = 1, \dots, d$ be a coordinate frame and assume that the frame ν has basis vectors ν_α for $\alpha = 1, \dots, d$ such that (x, ν) has coordinates (x^i, ν^i_α) where $\nu_\alpha = \nu^i_\alpha \frac{\partial}{\partial x^i}$. In these coordinates, a matrix representation of a sub-Riemannian metric $g_{F\mathcal{M}}\colon TF\mathcal{M}^\star \to HF\mathcal{M}$ is given by

$$(g_{F\mathcal{M}})_{ij} = \begin{pmatrix} W^{-1} & -W^{-1}\Gamma^T \\ -\Gamma W^{-1} & \Gamma W^{-1}\Gamma^T \end{pmatrix}, \tag{6.21}$$

where $(W^{-1})^{ij} = \delta^{\alpha\beta}\nu^i_\alpha\nu^j_\beta$ and the matrix $\Gamma = (\Gamma^{k_\alpha}_i)$ has elements $\Gamma^{k_\alpha}_i =$

$\Gamma^k_{ij}\nu^j_\alpha$. We refer to [125, 89, 118] for more details on sub-Riemannian structures and the derivation of the sub-Riemannian metric on $F\mathcal{M}$. Using the sub-Riemannian metric $g_{F\mathcal{M}}$, normal geodesics on $F\mathcal{M}$ can be generated by solving Hamilton's equations described earlier in (6.11).

Example 6.4.1 (Normal sub-Riemannian geodesics on $F\mathcal{M}$)**.** *With the same setup as in Example 6.2.1, let $u_0 = (x_0, \nu_0) \in FS^2$ such that $x_0 = F(0,0)$ and ν_0 has orthonormal frame vectors $\nu_1 = dF(0.5, 0)$, $\nu_2 = dF(0, 0.5)$. Figure 6.7 shows two geodesics on FS^2 visualised on S^2 with different initial momenta.*

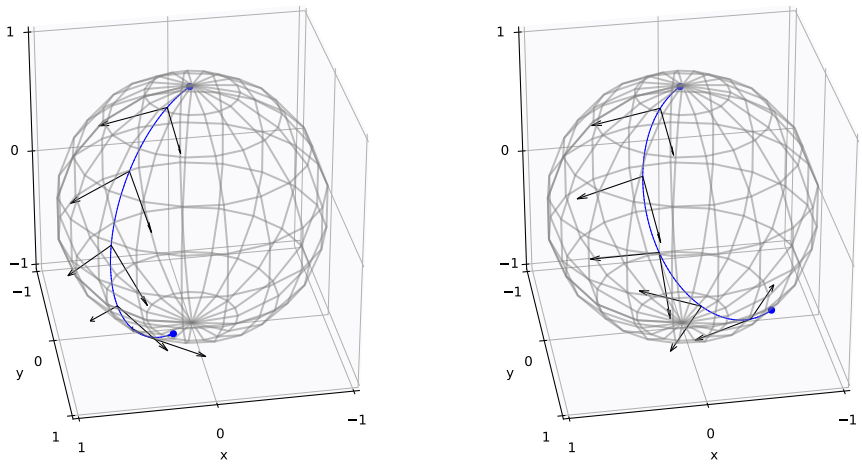


Figure 6.7: Geodesics on FS^2 solving Hamiltion's equations for the sub-Riemannian metric $g_{F\mathcal{M}}$ with different initial momenta. The curves on S^2 show the evolution of x_t while the evolution of the frame ν_t is shown by the tangent vectors in $T_{x_t}S^2$.

6.4.1 Curvature

The curvature form on the frame bundle is defined from the Riemannian curvature tensor $R \in \mathcal{T}^3_1(\mathcal{M})$ described in section 6.2.5 [61]. Let $u = (x, \nu)$ be a point in $F\mathcal{M}$, the curvature form $\Omega\colon T F\mathcal{M} \times T F\mathcal{M} \to \mathfrak{gl}(d)$ on the frame bundle is

$$\Omega(v_u, w_u) = u^{-1} R(\pi_*(v_u), \pi_*(w_u))u\,, \quad v_u, w_u \in T_u F\mathcal{M}\,, \tag{6.22}$$

where $\pi_*\colon T F\mathcal{M} \to T\mathcal{M}$ is the projection of a tangent vector of $F\mathcal{M}$ to a tangent vector of $\mathcal{M}$. By applying the relation,

$$\Omega(v_u, w_u) = \Omega(h_u(\pi_*(v_u)), h_u(\pi_*(w_u))), \tag{6.23}$$

where $h_u : T_{\pi(u)}\mathcal{M} \to H_u F\mathcal{M}$ denotes the horizontal lift, the curvature tensor R can be considered as a $\mathfrak{gl}(d)$ valued map

$$\begin{aligned} R_u \colon \mathcal{T}^2(T_{\pi(u)}\mathcal{M}) &\to \mathfrak{gl}(d) \\ (v, w) &\mapsto \Omega(h_u(\pi_*(v_u)), h_u(\pi_*(w_u))) \,, \end{aligned} \tag{6.24}$$

for $u \in F\mathcal{M}$. The implementation of the curvature form R_u is shown in the code below.

```
"""
Riemannian Curvature form
R_u (also denoted Omega) is the gl(n)-valued curvature form
  u^{-1}Ru for a frame u for T_xM

Args:
   x: point on the manifold

Returns:
   4-tensor (R_u)_ij^m_k with order i,j,m,k
"""
def R_u(x,u):
   return T.tensordot(T.nlinalg.matrix_inverse(u),
                      T.tensordot(R(x),u,(2,0)),
                      (1,2)).dimshuffle(1,2,0,3)
```

Example 6.4.2 (Curvature on S^2)**.** *Let $u = (x, \nu) \in F\mathcal{M}$ with $x = F(0, 0)$ and ν as shown in Figure 6.8 (solid arrows). We visualize the curvature at u by the curvature form $\Omega(\nu_1, \nu_2)$, calculated by applying R to the basis vectors of ν. The curvature is represented in Figure 6.8 by the dashed vectors showing the direction for which each basis vector change by parallel transporting the vectors around an infinitesimal parallelogram spanned by ν.*

6.4.2 Development and Stochastic Development

The short description of the development process in this section is based on the book [52]. The presented approach has also been described in [34, 118, 122], where the method is used for generalisation of Brownian motions to manifolds.

Using the frame bundle and its horizontal and vertical splitting, deterministic paths and stochastic processes on $F\mathcal{M}$ can be constructed from paths and stochastic processes on $\mathbb{R}^d$. In the deterministic case, this process is called development and when mapping Euclidean semi-martingales to $\mathcal{M}$-valued semi-martingales, the corresponding mapping is stochastic development. The development unrolls paths on $F\mathcal{M}$ by taking infinitesimal steps corresponding to a curve in $\mathbb{R}^d$ along a basis of $HF\mathcal{M}$. Let $e \in \mathbb{R}^d$ and

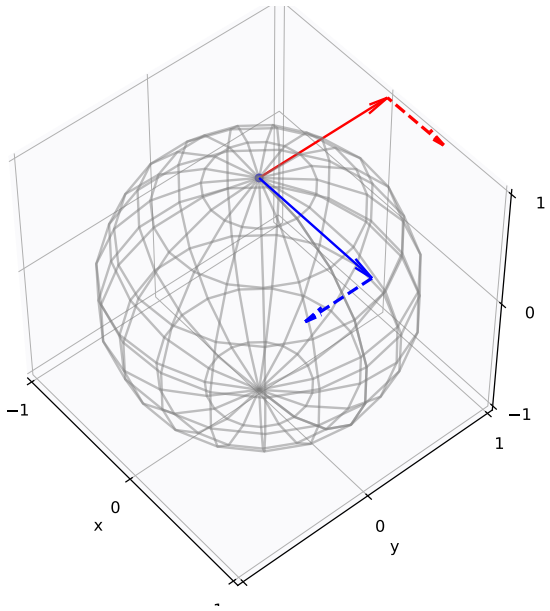


Figure 6.8: Curvature of each basis vector of ν. The solid arrows represents the basis vectors, while the dashed arrows are the curvature form $\Omega(\nu_1, \nu_2)$. The figure shows in which direction the basis vectors would change if they were parallel transported around an infinitesimal parallelogram spanned by the basis vectors of ν.

$u = (x, \nu) \in F\mathcal{M}$, then a horizontal vector field $H_e \in H_u F\mathcal{M}$ can be defined by the horizontal lift of the vector $\nu e \in T_x\mathcal{M}$, that is

$$H_e(x) = h_u(\nu e)\,.$$

If $e_1, \ldots, e_d$ is the canonical basis of $\mathbb{R}^d$, then for any $u \in F\mathcal{M}$, a basis for the horizontal subspace $H_u F\mathcal{M}$ is represented by the horizontal vector fields $H_i(x) = H_{e_i}(x)$, $i = 1, \ldots, d$. Consider a local chart (U, φ) on $\mathcal{M}$, the coordinate basis $\partial_i = \frac{\partial}{\partial x^i}$ on U, and the projection map $\pi\colon F\mathcal{M} \to \mathcal{M}$, then the coordinate basis ∂_i induces a local basis on the subset $\tilde{U} = \pi^{-1}(U) \subseteq F\mathcal{M}$. Notice that the basis vectors $\nu e_1, \ldots, \nu e_d$ of $T_x\mathcal{M}$ can be written as $\nu e_j = \nu_j^i \partial_i$ for each $j = 1, \ldots, d$. Hence (x^i, ν_j^i) is a chart for $\tilde{U}$ and $\left(\frac{\partial}{\partial x^i}, \frac{\partial}{\partial \nu_j^i}\right)$ spans the tangent space $T_u F\mathcal{M}$. The horizontal vector fields can be written in this local coordinate basis as

$$H_i(q) = \nu_i^j \frac{\partial}{\partial x^j} - \nu_i^j \nu_m^l \Gamma_{jl}^k \frac{\partial}{\partial \nu_m^k}\,. \tag{6.25}$$

The code below shows how these horizontal vector fields in the local basis can be implemented in Theano.

```
"""
Horizontal Vector Field Basis

Args:
    x: Point on the manifold
    nu: Frame for the tangent space at x
    Gamma_g(x): Christoffel symbols at x

Returns:
    Matrix of coordinates for each basis vector
"""
def Hori(x,nu):
    dnu = - T.tensordot(nu, T.tensordot(nu,Gamma_g(x),axes =
        [0,2]),
                        axes = [0,2])
    dnu = dnu.reshape((nu.shape[1],dnu.shape[1]*dnu.shape[2]))
    return T.concatenate([nu,dnu.T], axis = 0)
```

Example 6.4.3 (Horizontal vector fields)**.** *Figure 6.9 illustrates the horizontal vector fields H_i at a point $u \in FS^2$. Let $u = (x, \nu)$ with $x = F(0.1, 0.1) \in S^2$ and ν being the black frame shown in the figure. The horizontal basis for u is then found by (6.25) and is plotted in Figure 6.9 with the red frame being the horizontal basis vectors for x and the blue frames are the horizontal basis vectors for each frame vector in ν. The horizontal basis vectors describe how the point x and the frame ν change horizontally.*

Let now W_t be a $\mathbb{R}^d$-valued Euclidean semi-martingale, e.g. a Brownian motion. The stochastic version of the development maps W_t to $F\mathcal{M}$ by the solution to the Stratonovich stochastic differential equation

$$dU_t = \sum_{i=1}^{d} H_i(U_t) \circ dW_t^i \,. \tag{6.26}$$

The solution U_t to this stochastic differential equation is a path in $F\mathcal{M}$ for which a stochastic path on $\mathcal{M}$ can be obtained by the natural projection $\pi : U_t \to \mathcal{M}$. The stochastic development of W_t will be denoted $\varphi_{u_0}(W_t)$ where $u_0 \in F\mathcal{M}$ is the initial point on $F\mathcal{M}$. In Theano this Stratonovich stochastic differential equation can be implemented as follow.

```
"""
Stochastic Development

Args:
    dW: Steps of stochastic process
    u: Point in FM
    drift: Vector of constant drift of W

Returns:
```

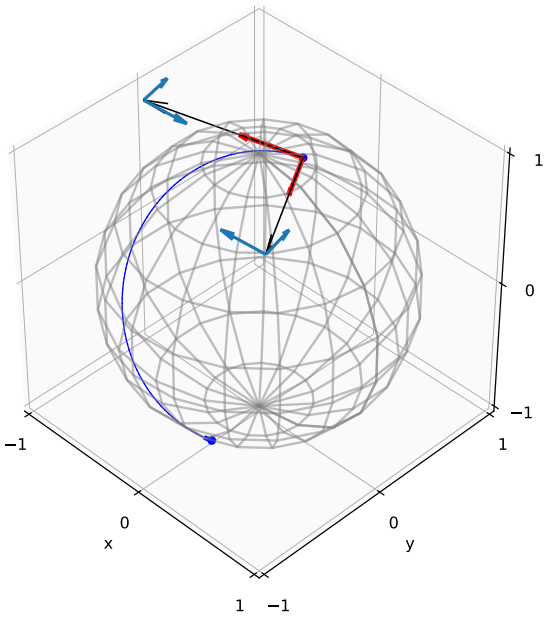


Figure 6.9: Horizontal vector fields for the point $u = (x, \nu) \in FM$ with $x = F(0.1, 0.1)$ and the frame ν visualized with black arrows. The horizontal tangent vectors at x is shown in red and the horizontal tangent vectors for each tangent vector at ν is shown in blue.

```
    det: Matrix of deterministic evolution of process on FM
    sto: Matrix of stochastic evolution of the process
"""
def stoc_dev(dW,u,drift):
    x = u[0:d]
    nu = u[d:(d+rank*d)].reshape((d,rank))
    det = T.tensordot(Hori(x,nu), drift, axes = [1,0])
    sto = T.tensordot(Hori(x,nu), dW, axes = [1,0])
    return det, sto
```

The variable `drift` can be used to find the stochastic development of a process with defined drift. The numerical solution to this SDE requires the use of stochastic numerical integration methods, described in the appendix 6.A, such as the Euler-Heun scheme, used in the example below.

Example 6.4.4 (Deterministic and stochastic Development). *Let γ_t be a curve in $\mathbb{R}^2$ defined by*

$$\gamma(t) = (20\sin(t), t^2 + 2t), \quad t \in [0, 10]\,,$$

and $x = F(0,0) \in S^2$. Consider the orthonormal frame for $T_x\mathcal{M}$ given by the Gram-Schmidt decomposition based on the metric g of the vectors $v_1 = dF(-1, 1)$,

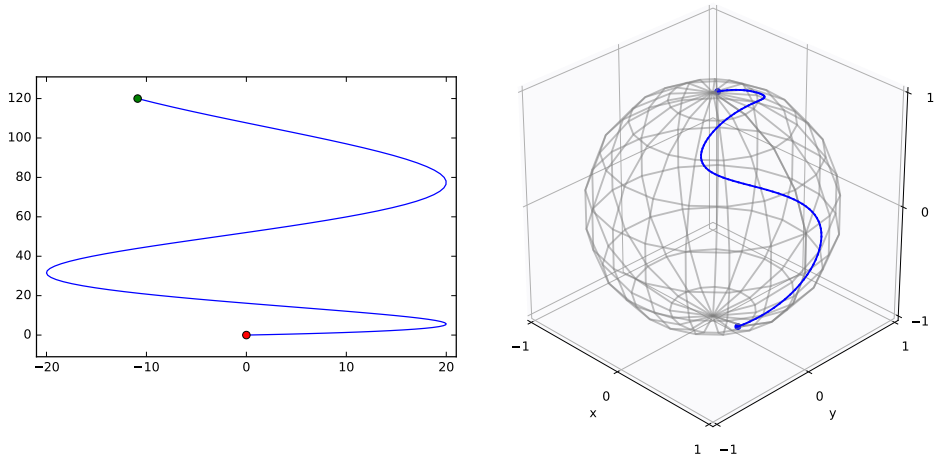


Figure 6.10: (left) The curve γ_t defined in Example 6.4.4. The red and green point denotes the start and endpoint respectively. (right) The development of γ_t on the sphere.

$v_2 = dF(1,1)$. *The curve* γ_t *is a deterministic process in* $\mathbb{R}^2$ *and hence* (6.26) *can be applied to obtain the development of* γ_t *to* S^2. *In Figure 6.10 is shown the curve* γ_t *and its development on the sphere.*

Let then X_t *be a stochastic process in* $\mathbb{R}^2$ *defined from a Brownian motion,* W_t, *with drift,* β. *Discretizing in time, the increments* dW_t *follow the normal distribution* $\mathcal{N}(0, dtI_2)$, *here with* $dt = 0.0001$. *Let* $\beta = (0.5, 0.5)$ *such that*

$$dX_t = dW_t + \beta dt\,.$$

A sample path of X_t *is shown in Figure 6.11. The stochastic development of* X_t *is obtained as the solution to the Stratonovich stochastic differential equation defined in* (6.26). *The resulting stochastic development on* S^2 *is shown in the right plot of Figure 6.11.*

6.4.3 Most Probable Path equations

The most common distance measure on Riemannian manifolds is the geodesic distance. However, in contexts where data exhibit non-trivial covariance, it is argued in [118, 122] that weighting the geodesic energy by the inverse of the covariance, the precision, gives a useful generalization of the geodesic distance. Extremal paths for the corresponding variational problem are precisely projections of $F\mathcal{M}$ geodesics with respect to the sub-Riemannian metric $g_{F\mathcal{M}}$ constructed earlier. These paths also have an interpretation as being most probable for a specific measure on the path space.

More formally, let X_t be a stochastic process with $X_0 = x_0$. Most probable paths in the sense of Onsager-Machlup [40] between $x_0, y \in \mathcal{M}$ are curves

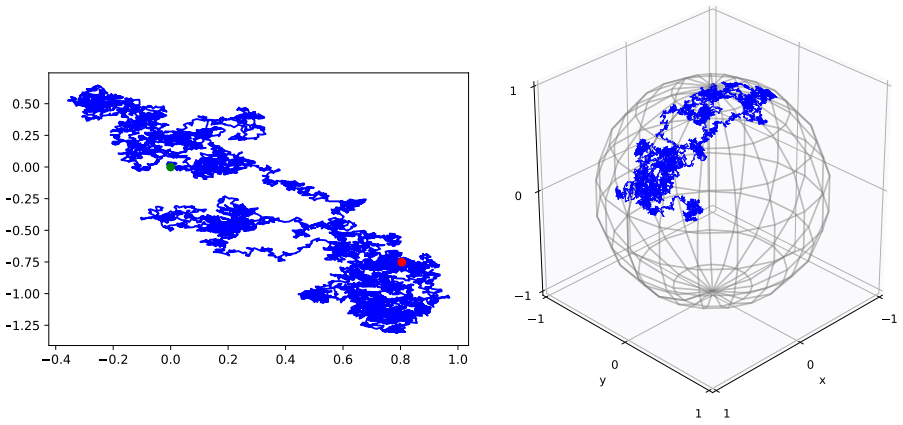


Figure 6.11: (left) The stochastic process X_t defined in Example 6.4.4. The red and green point denotes the start–and endpoint respectively of the process. (right) The stochastic development of X_t on S^2.

$\gamma_t \colon [0,1] \to \mathcal{M}$, $\gamma_0 = x_0$ maximizing

$$\mu_\varepsilon^M(\gamma_t) = P(d_g(X_t, \gamma_t) < \varepsilon,\ \forall t \in [0,1])\,, \tag{6.27}$$

for $\varepsilon \to 0$ and with the Riemannian distance d_g. Most probable paths are in general not geodesics but rather extremal paths for the Onsager-Machlup functional

$$\int_0^1 L_M(\gamma_t, \dot{\gamma}_t)\, dt = -E[\gamma_t] + \frac{1}{12}\int_0^1 S(\gamma_t)\, dt\,. \tag{6.28}$$

Here, S denotes the scalar curvature of $\mathcal{M}$ and the geodesic energy is given by $E[\gamma_t] = \frac{1}{2}\int_0^1 \|\dot{\gamma}_t\|_g^2\, dt$. In comparison, geodesics only minimize the energy $E[\gamma_t]$.

Instead of calculating the MPPs based on the Onsager-Machlup functional on the manifold, the MPPs for the driving process W_t can be found. It has been shown in [122] that under reasonable conditions, the MPPs of the driving process exist and coincide with projections of the sub-Riemannian geodesics on $F\mathcal{M}$ obtained from (6.11) with the sub-Riemannian metric $g_{F\mathcal{M}}$. The implementation of the MPPs shown below is based on this result and hence returns the tangent vector in $T_u F\mathcal{M}$ which leads to the sub-Riemannian geodesic on $F\mathcal{M}$ starting at u and hitting the fibre at y.

Let W_t be a standard Brownian motion and $X_t = \varphi_{u_0}(W_t)$, the stochastic development of W_t with initial point $u_0 \in F\mathcal{M}$. Then, the most probable path of the driving process W_t from $x_0 = \pi(u_0)$ to $y \in \mathcal{M}$ is defined as a smooth curve $\gamma_t \colon [0,1] \to \mathcal{M}$ with $\gamma_0 = x_0$, $\gamma_1 = y$ satisfying

$$\underset{\gamma_t, \gamma_0 = x_0, \gamma_1 = y}{\arg\min} \int_0^1 -L_{\mathbb{R}^n}\left(\varphi_{u_0}^{-1}(\gamma_t), \frac{d}{dt}\varphi_{u_0}^{-1}(\gamma_t)\right) dt\,, \tag{6.29}$$

that is, the anti-development $\varphi_{u_0}^{-1}(\gamma_t)$ is the most probable path of W_t in $\mathbb{R}^n$. The implementation of the MPPs is given below.

```
"""
Most probable paths for the driving process

Args:
   u: Starting point in FM
   y: Point on M

Returns:
   MPP: vector in T_uFM for sub-Riemannian geodesic hitting
       fiber above y
"""
loss = lambda v,u,y: 1./d*T.sum((Expfm(u,g(u,v))[0:d]-y)**2)
dloss = lambda v,u,y: T.grad(loss(v,u,y),v)
# Returns the optimal horizontal tangent vector defining the
    MPP:
MPP = minimize(loss, np.zeros(d.eval()), jac=dloss, args=(u,y
    ))
```

Example 6.4.5 (Most Probable Path on ellipsoid). *Let $u_0 = (x_0, \nu_0) \in F\mathcal{M}$ for which $x_0 = F(0,0)$ and ν_0 consists of the tangent vectors $dF(0.1, 0.3)$, $dF(0.3, 0.1)$ and $y = F(0.5, 0.5) \in S^2$. We then obtain a tangent vector*

$$v = (1.03, -5.8, 0, 0, 0, 0) \in H_{u_0} F\mathcal{M}$$

which leads to the MPP shown in Figure 6.12 as the blue curve. For comparison, the Riemannian geodesic between x_0 and y is shown in green.

6.5 Landmark Dynamics

In this section, we will apply the previous generic algorithms to the example of the manifold of landmarks, seen as a finite dimensional representation of shapes in the Large Deformation Diffeomorphic Metric Mapping (LDDMM). We will not review this theory in details here but only show how to adapt the previous code to this example. We refer to the book [12] for more details and LDDMM and landmark dynamics.

Let $\mathcal{M} \cong \mathbb{R}^{dn}$ be the manifold of n landmarks with positions $\boldsymbol{x}_i \in \mathbb{R}^d$ on a d-dimensional space. From now on, we will only consider landmarks in a plane, that is $d = 2$. In the LDDMM framework, deformations of shapes are modelled as flows on the group of diffeomorphisms acting on any data structure, which in this case are landmarks. To apply this theory, we need to have a special space, a reproducing kernel Hilbert space (RKHS), denoted by V. In general, an RKHS is a Hilbert space of functions for which evaluations of a function $v \in V$ at a point $x \in \mathcal{M}$ can be performed as an inner product

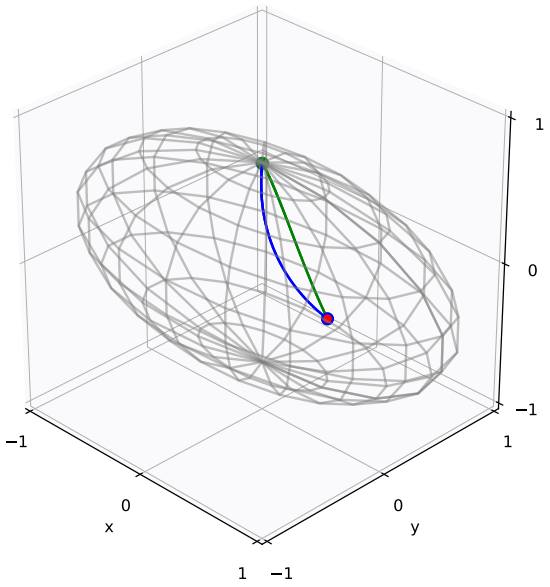


Figure 6.12: A most probable path between $x_0 = F(0,0)$ and $y = F(0.5,0.5)$ (red point) on an ellipsoid. The blue curve is the MPP and the green the Riemannian geodesic between x_0 and y.

of v with a kernel evaluated at x. In particular, for $v \in V$, $v(x) = \langle K_x, v\rangle_V$ for all $x \in M$, for which $K_x = K(.,x)$. In all the examples of this paper, we will use a Gaussian kernel given by

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \alpha \cdot \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right), \tag{6.30}$$

with standard deviation $\sigma = 0.1$ and a scaling parameter $\alpha \in \mathbb{R}^d$.

The diffeomorphisms modelling the deformation of shapes in $\mathcal{M}$ is defined by the flow

$$\partial_t \varphi(t) = v_t \circ \varphi(t), \text{ for } v_t \in V, \tag{6.31}$$

where $\varphi : \mathcal{M} \to \mathcal{M}$ and $\circ$ means evaluation $v_t(\varphi)$ for a time-dependent vector field v_t. Given a shape $x_1 \in \mathcal{M}$, a deformation of x_1 can be obtained by applying to x_1 a diffeomorphism φ obtained as a solution of (6.31) for times bteween 0 and 1. We write $x_2 = \varphi(1) \cdot x_1$, the resulting deformed shape.

Let a shape x in the landmark manifold $\mathcal{M}$ be the vector of positions $x = (x_1^1, x_1^2, \ldots, x_n^1, x_n^2)$, where the upper indices are the positions of each landmark on the image. Consider $\xi, \eta \in T_x^*\mathcal{M}$. The cometric on $\mathcal{M}$ is thus

$$g_x^*(\xi, \eta) = \sum_{i,j=1}^{n} \xi_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) \eta_j, \tag{6.32}$$

where the components of the cometric are $g^{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for $\boldsymbol{x}_i = (x_i^1, x_i^2)$. The coordinates of the metric are the inverse kernel matrix $g_{ij} = K^{-1}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and the cometric (6.32) corresponds to the standard landmark Hamiltonian when $\xi = \eta = p$, the momentum vector of the landmarks.

Recall that the Christoffel symbols depend only on the metric, hence they can be obtained by the general equation (6.4). Geodesics on $\mathcal{M}$ can then be obtained as solutions of Hamilton's equations described in section 6.2.3 with this landmark Hamiltonian. An example of geodesics for two landmarks is shown in Figure 6.13 along with an example of a geodesic on the frame bundle $F\mathcal{M}$, obtained as the solution to Hamilton's equations generated from the sub-Riemannian structure on $F\mathcal{M}$ described in section 6.4.

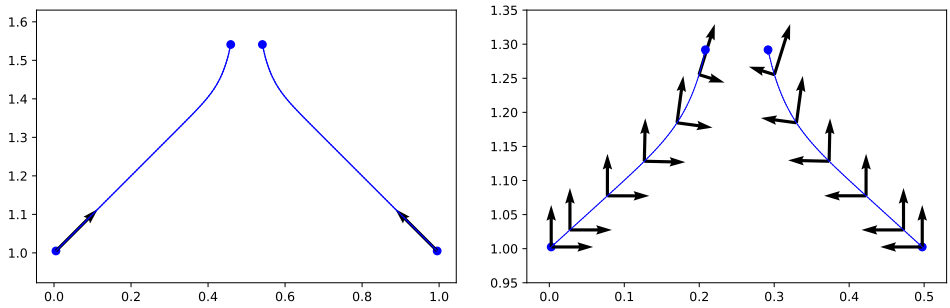


Figure 6.13: Geodesics on the landmark manifold. (left) Geodesic on $\mathcal{M}$ found with Hamilton's equations. (right) Geodesic on $F\mathcal{M}$ as the solution to Hamilton's equations generated from the sub-Riemannian structure on $F\mathcal{M}$.

Example 6.5.1 (Stochastic Development). *We use a two landmarks manifold $\mathcal{M}$, that is $\dim(\mathcal{M}) = 4$. Then, as in Example 6.4.4, we consider the curve $\gamma_t = (20\sin(t), t^2+2t)$, $t \in [0, 10]$ (Figure 6.14 top left panel) and point $x = (0, 1, 0.5, 1)$ in $\mathcal{M}$. The initial frame for each landmark is given as the canonical basis vectors $e_1 = (1, 0)$, $e_2 = (0, 1)$ shown in Figure 6.14 (top right panel) as well as the deterministic development of γ_t to $\mathcal{M}$. Figure 6.14 (bottom right panel) shows an example of a stochastic development for a 4-dimensional stochastic process W_t displayed on the bottom left panel. Notice that in the deterministic case, a single curve was used for both landmarks, thus their trajectories are similar and only affected by the correlation between landmarks. In the stochastic case, the landmarks follow different stochastic paths, also affected by the landmarks interaction.*

The examples shown in this section can, in addition, be applied to a higher dimensional landmark manifold as seen in figure 6.1. For more examples on Theano code used with more landmarks on for example the Corpus Callosum shapes, we refer to [65, 4]. For another stochastic deformation of shapes in the context of computational anatomy, with examples on land-

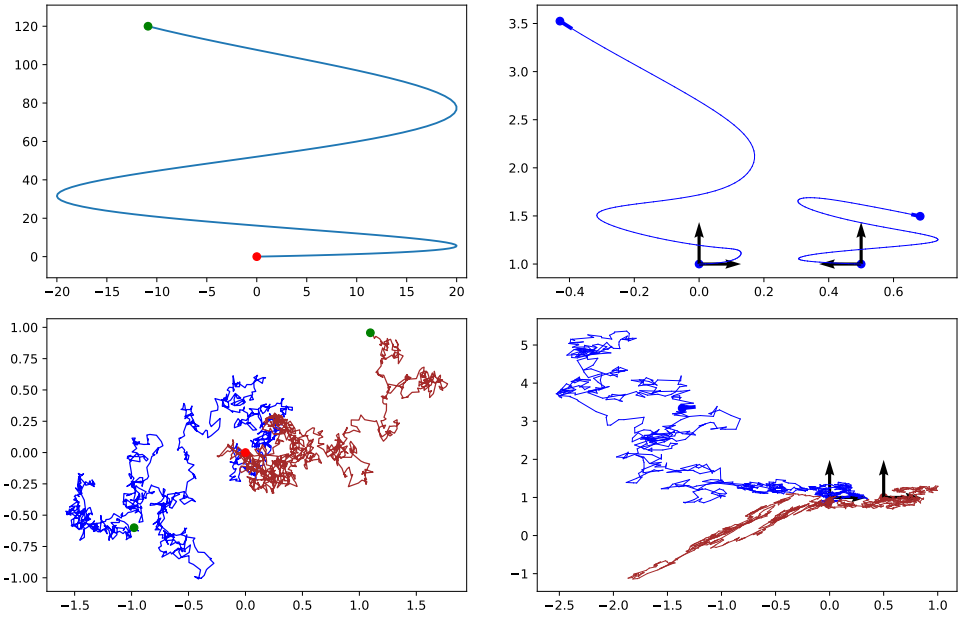


Figure 6.14: Deterministic development (top left) The curve γ_t defined in Example 6.5.1. The red and green point denotes the start–and endpoint of the process respectively using the displayed frame. (right) The development of γ_t on each landmark. Bottom row: Stochastic development (left) Brownian motion, W_t, in $\mathbb{R}^4$ plotted as two processes. (right) The stochastic development of W_t to the manifold.

marks, we refer to [5, 4], where the focus was on noise inference in these models. These works were inspired by [49], where stochastic models for fluid dynamics were introduced such that geometrical quantities remain preserved, and applied for finite dimensions in [3]. In the same theme of stochastic landmark dynamics, [85] introduced noise and dissipation to also tackle noise inference problems and [50] considered parametric stochastic deformation in the variational principle originated from the Euler-Poincaré equations.

6.6 Non-Linear Statistics

This section focuses on a selection of basic statistical concepts generalized to manifolds and how these can be implemented in Theano. We refer to [101] for an overview of manifold valued statistics.

6.6.1 Fréchet Mean

The Fréchet Mean is an intrinsic generalization of the mean-value in Euclidean space [38]. Consider a manifold $\mathcal{M}$ with a distance d and let P be a probability measure on $\mathcal{M}$. The Fréchet mean set is defined as the set of points minimizing the function

$$F(y) = \arg\min_{x \in \mathcal{M}} \mathbb{E}_P \left[d(x,y)^2 \right], \quad y \in \mathcal{M}. \tag{6.33}$$

Unlike the Euclidean mean, the solution to (6.33) is not necessarily unique. If the minimum exists and is unique, the minimum is called the Fréchet mean. The Fréchet mean for a sample of data points $y_1, \ldots, y_n$ is estimated as

$$F_{\bar{y}} = \arg\min_{x \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} d(x,y_i)^2 . \tag{6.34}$$

When considering a Riemannian manifold, a natural choice of distance measure is the geodesic distance described in section 6.2.2. With this choice of distance, the empirical Fréchet mean reduces to

$$F_{\bar{y}} = \arg\min_{x \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} \|\mathrm{Log}(x,y_i)\|^2 , \tag{6.35}$$

which can be implemented in Theano as follow.

```
"""
Frechet Mean

Args:
    x: Point on the manifold
    y: Data points
    x0: Initial point for optimization

Returns:
    The average loss from x to data y

"""
def Frechet_mean(x,y):
    (cout,updates) = theano.scan(fn=loss, non_sequences=[v0,x
        ],
                                 sequences=[y], n_steps=n_samples)
    return 1./n_samples*T.sum(cout)
dFrechet_mean = lambda x,y: T.grad(Frechet_mean(x,y),x)
FMean = minimize(Frechet_mean, x0, jac=dFrechet_mean, args=y)
```

Example 6.6.1 (Fréchet mean on S^2)**.** *Consider the Levi-Civita connection on S^2 and equip S^2 with the geodesic distance given in (6.9). A sample set of size 20 is*

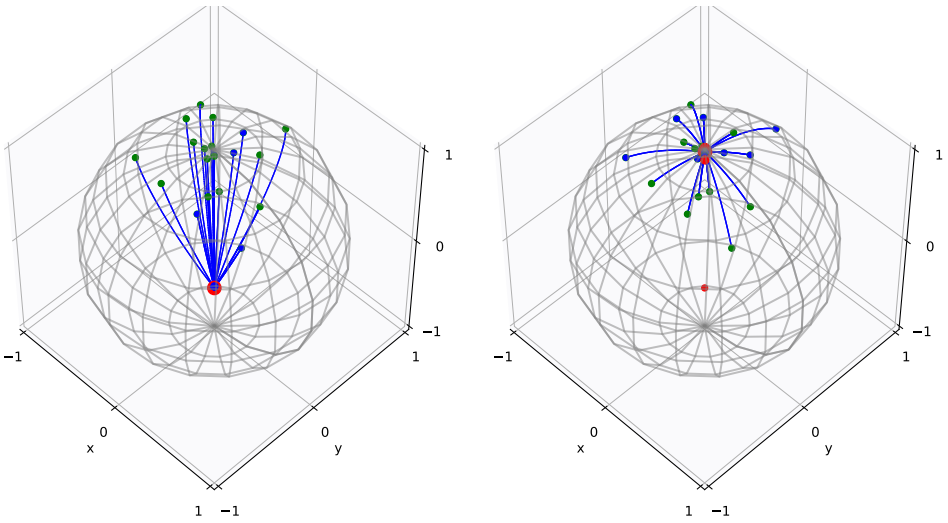


Figure 6.15: (left) Sampled data points, with the red point being the initial guess of the mean. (right) The resulting empirical Frechet mean. The iterated results are visualized as red dots. The final result is the largest red dot with the distance minimizing geodesics to each datapoint.

generated on the northern hemisphere. Each coordinate of a sample point has been drawn from a normal distribution with mean 0 *and standard deviation* 0.2*. The initial guess of the Fréchet mean is* $F(0.4, -0.4)$*. The sample set and initial mean are shown in the left plot of Figure 6.15. The resulting empirical Frechet mean found with the implementation above is visualized in Figure 6.15.*

The Fréchet mean can not just be used to calculate the mean on manifolds. In [122], the authors presented a method for estimating the mean and covariance of normal distributions on manifolds by calculating the Fréchet mean on the frame bundle. The next section will describe a way to generalize normal distributions to manifolds.

6.6.2 Normal Distributions

Normal distributions in Euclidean spaces can be considered as the transition distribution of Brownian motions. The generalization of normal distributions to manifolds can be defined in a similar manner. In [34], isotropic Brownian motions on $\mathcal{M}$ are constructed as the stochastic development of isotropic Brownian motions on $\mathbb{R}^n$ based on an orthonormal frame. However, [119, 122] suggested performing stochastic development with non-orthonormal frames, which leads to anisotropic Brownian motions on $\mathcal{M}$. Let W_t be a Brownian motion on $\mathbb{R}^2$ and consider the initial point $u = (x, \nu) \in FS^2$, for $x = F(0, 0)$ and ν the frame consisting of the canonical basis vectors e_1, e_2. An

example of a Brownian motion path on the sphere, derived as the stochastic development of W_t in $\mathbb{R}^2$, is shown in Figure 6.16.

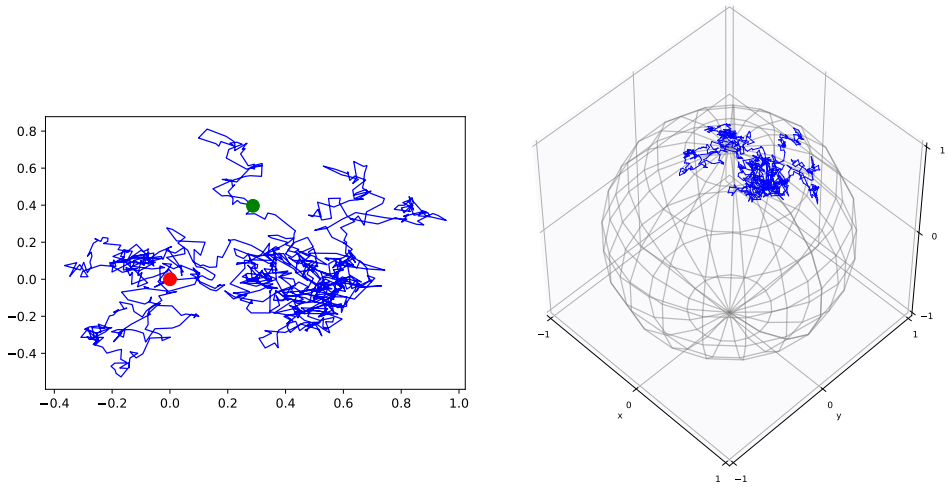


Figure 6.16: (left) Brownian motion, W_t, in $\mathbb{R}^2$. (right) The stochastic development of W_t to the sphere with initial point $u = (x, \nu)$, for $x = F(0, 0)$ and ν the frame consisting of the canonical basis vectors e_1, e_2.

Based on the definition of Brownian motions on a manifold, normal distributions can be generalized as the transition distribution of Brownian motions on $\mathcal{M}$. Consider the generalization of the normal distribution $\mathcal{N}(\mu, \Sigma)$. When defining the normal distribution on $\mathcal{M}$ as the stochastic development of Brownian motions, the initial point on $\mathcal{M}$ is the mean and the initial frame represents the covariance of the resulting normal distribution.

Example 6.6.2 (Normal distributions on S^2)**.** *Let W_t be a Brownian motion on $\mathbb{R}^2$ and consider $x = F(0, 0) \in S^2$ being the mean of the normal distributions in this example. Two normal distributions with different covariance matrices have been generated, one isotropic and one anisotropic distribution. The normal distributions are $\mathcal{N}(0, \Sigma_i)$ for $i = 1, 2$ with covariance matrices*

$$\Sigma_1 = \begin{pmatrix} 0.15 & 0 \\ 0 & 0.15 \end{pmatrix}, \quad \Sigma_2 = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}. \tag{6.36}$$

As explained above, the initial frame ν represents the covariance of the normal distribution on a manifold $\mathcal{M}$. Therefore, we chose ν_1 with basis vectors being the columns of Σ_1 and ν_2 with basis vectors represented by the columns of Σ_2. Density plots of the resulting normal distributions are shown in Figure 6.17.

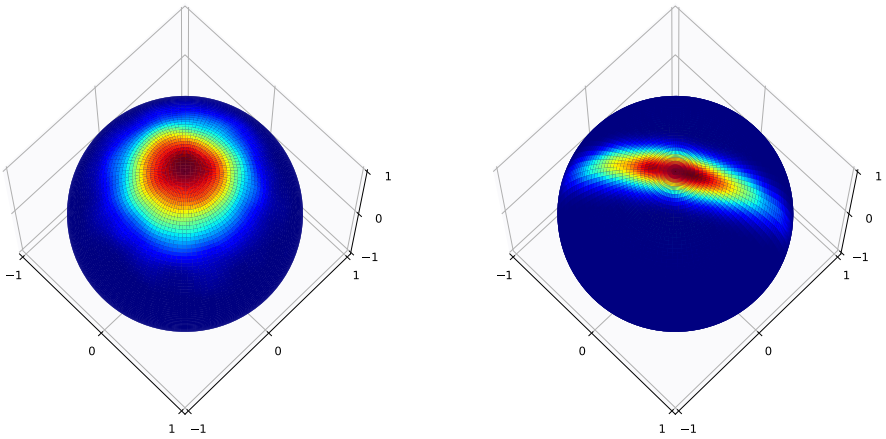


Figure 6.17: (left) Density estimate of the isotropic normal distribution on S^2 with covariance Σ_1 given in (6.36). (right) Density estimate of the anisotropic normal distribution on S^2 with covariance Σ_2.

6.7 Conclusion

In this paper, we have shown how the Theano framework and Python can be used for implementation of concepts from differential geometry and non-linear statistics. The opportunity to perform symbolic calculations makes implementations of even complex concepts such as stochastic integration and fibre bundle geometry easy and concise. The symbolic representation is often of great practical value for the implementation process, leading to shorter code, fewer bugs, and faster implementations, and formulas can almost directly be translated to Theano code. As seen in the examples, the symbolic representation of functions allows taking derivatives of any variables and of any orders. The task of calculating gradients for optimization procedures can be difficult and prone to errors while with symbolic calculations, only a few lines of code is needed to optimize over, for instance, the parameters of a stochastic integrator or the evolution of a sub-Riemannian geodesic. This makes numerical testing of new ideas fast and efficient and easily scalable to useful applications if optimized for parallel computers of GPUs.

We have just shown here a small fragment of mathematical problems which can be implemented with Theano and other similar software. Other problems that could be solved using these methods can be found in statistical analysis on manifold-valued data, such as geodesic regression, longitudinal analysis, and PCA, or in computational anatomy, by solving registration problem on continuous shapes and images and analysing or modelling

shape deformations. For example, we refer to [65, 4, 5] for further examples of Theano in the field of computational anatomy which were not treated here.

Packages such as Theano have their limitations, and one must sometimes be careful in the implementation and aware of the limitations of the algorithms. For example, if equations are simple enough that derivatives can be written explicitly, the code can in some situations be faster when computing from the explicit formula rather than relying on the automatic differentiation. For complicated constructions, the compilation step can be computationally intensive as well as memory demanding. Such limitations can be overcome by carefully writing the code in order to limit the compilation time and have the parameters of Theano properly adjusted to the machine at hand.

With this paper and its accompanying code[1], we hope to stimulate the use of modern symbolic and numerical computation frameworks for experimental applications in mathematics, for computations in applied mathematics, and for data analysis by showing how the resulting code allows for flexibility and simplicity in implementing many experimental mathematics endeavours.

Acknowledgements

LK and SS are supported by the CSGB Centre for Stochastic Geometry and Advanced Bioimaging funded by a grant from the Villum Foundation. AA is supported by the EPSRC through award EP/N014529/1 funding the EPSRC Centre for Mathematics of Precision Healthcare.

6.A Stochastic integration

In the following, we give a brief description of some basic theory on stochastic differential equations and stochastic integration methods. The symbolic specification in Theano allows us to take derivatives of parameters specifying the stochastic evolutions, and the presented methods can, therefore, be used for e.g. maximum likelihood estimation over stochastic processes. The theory in this appendix is based on [113].

6.A.1 Stochastic Differential Equations

We consider here stochastic processes, U_t in $\mathbb{R}^n$, solutions to SDEs of the form

$$dU_t = f(U_t, t)dt + g(U_t, t)dW_t, \quad t \in [0, T], \tag{6.37}$$

with drift $f(U_t, t)$ and diffusion field $g(U_t, t)$, functions from $\mathbb{R}^n \times \mathbb{R}$ to $\mathbb{R}^n$.

There are two types of stochastic differential equations; Itô and Stratonovich differential equations. The Stratonovich SDEs are usually denoted with

[1] http://bitbucket.org/stefansommer/theanogeometry

$\circ$, such that (6.37) reduces to

$$dU_t = f(U_t, t)dt + g(U_t, t) \circ dW_t \,. \tag{6.38}$$

For integration of deterministic ODEs, solutions to the integral equation can be defined as the limit of a sum of finite differences over the time interval. In this case, it does not matter in which point of the intervals the function is evaluated. For stochastic integrals, this is not the case. Itô integrals are defined by evaluating at the left point of the interval, while Stratonovich integrals use the average between the value at the two endpoints of the interval. The two integrals do not result in equal solutions, but they are related by

$$g(U_t, t)dW_t = \frac{1}{2}dg(U_t, t)g(U_t, dt)dt + g(U_t, t) \circ dW_t \,, \tag{6.39}$$

where dg denotes the Jacobian of g [14]. Whether to choose Itô or the Stratonovich framework depends on the problem to solve. One benefit from choosing the Stratonovich integral is that it obeys the chain rule making it easy to use in a geometric context.

6.A.2 Discrete Stochastic Integrators

We generally need numerical integration to find solutions to SDEs. There are several versions of numerical integrators of different order of convergence. Two simple integrators are the Euler method for Itô SDEs and the Euler-Heun for the Stratonovich SDEs.

Euler Method. Consider an Itô SDE as defined in (6.37). Let $0 = t_0 < t_1 < \ldots < t_n = T$ be a discretization of the interval $[0, T]$ for which the stochastic process is defined and assume $\Delta t = T/n$. Initialize the stochastic process, $U_0 = u_0$ for some initial value u_0. The process U_t is then recursively defined for each time point t_i by,

$$U_{t_{i+1}} = U_{t_i} + f(U_{t_i}, t_i)\Delta t + g(U_{t_i}, t_i)\Delta W_i \,, \tag{6.40}$$

in which $\Delta W_i = W_{t_{i+1}} - W_{t_i}$. Given an Itô stochastic differential equation, `sde_f`, the Euler method can be implemented in Theano by the following code example.

```
"""
Euler Numerical Integration Method

Args:
    sde: Stochastic differential equation to solve
    integrator: Choice of integrator_ito or
        integrator_stratonovich
    x: Initial values for process
    dWt: Steps of stochastic process
    *ys: Additional arguments to define the sde
```

```
Returns:
   integrate_sde: Tensor (t,xt)
                  t: Time evolution
                  xt: Evolution of x
"""
def integrator_ito(sde_f):
   def euler(dW,t,x,*ys):
      (detx, stox, X, *dys) = sde_f(dW,t,x,*ys)
      ys_new = ()
      for (y,dy) in zip(ys,dys):
         ys_new = ys_new + (y+dt*dy,)
      return (t+dt,x + dt*detx + stox, *ys_new)
   return euler


# Integration:
def integrate_sde(sde,integrator,x,dWt,*ys):
   (cout, updates) = theano.scan(fn=integrator(sde),
         outputs_info=[T.constant(0.),x, *ys],
         sequences=[dWt],
         n_steps=n_steps)
   return cout
```

Euler-Heun Method. An equivalent integration method as the Euler method for Itô SDEs, is the Euler-Heun method used to approximate the solution to Stratonovich SDEs. Consider a similar discretization as in the Euler method. The Euler-Heun numerical integration method is then defined as,

$$U_{t_{i+1}} = U_{t_i} + f(U_{t_i}, t_i)\Delta t + \frac{1}{2}\left(g(U_{t_i}, t_i) + g(\hat{U}_{t_i}, t_i)\right)\Delta W_i\,, \tag{6.41}$$

where $\hat{U}_{t_i} = U_{t_i} + g(U_{t_i}, t_i)\Delta W_i$. The implementation of the Euler-Heun method is similar to the Euler method, such that based on a Stratonovich SDE, `sde_f`, the implementation can be executed as follows,

```
"""
Euler-Heun Numerical Integration Method

Args:
   sde: Stochastic differential equation to solve
   integrator: Choice of integrator_ito or
      integrator_stratonovich
   x: Initial values for process
   dWt: Steps of stochastic process
   *ys: Additional arguments to define the sde

Returns:
   integrate_sde: Tensor (t,xt)
               t: Time evolution
```

```
            xt: Evolution of x
"""
def integrator_stratonovich(sde_f):
   def euler_heun(dW,t,x,*ys):
      (detx, stox, X, *dys) = sde_f(dW,t,x,*ys)
      tx = x + stox
      ys_new = ()
      for (y,dy) in zip(ys,dys):
         ys_new = ys_new + (y+dt*dy,)
      return (t+dt,
             x + dt*detx + 0.5*(stox + sde_f(dW,t+dt,tx,*ys)
                [1]),
             *ys_new)
   return euler_heun

# Integration:
def integrate_sde(sde,integrator,x,dWt,*ys):
   (cout, updates) = theano.scan(fn=integrator(sde),
         outputs_info=[T.constant(0.),x, *ys],
         sequences=[dWt],
         n_steps=n_steps)
   return cout
```

CHAPTER 7

Computational Anatomy in Theano

The paper presented in this chapter was accepted for the workshop MFCA at MICCAI 2017 and published in the conference proceedings

- L. Kühnel and S. Sommer. *Computational Anatomy in Theano*, pages 164–176. Springer International Publishing, 2017

The content overlap with the longer journal version presented in Chapter 6, but focus on applying the methods to analyse data in the field of computational anatomy.

Computational Anatomy in Theano

Line Kühnel, and Stefan Sommer

Department of Computer Science, University of Copenhagen

Abstract

To model deformation of anatomical shapes, non-linear statistics are required to take into account the non-linear structure of the data space. Computer implementations of non-linear statistics and differential geometry algorithms often lead to long and complex code sequences. The aim of the paper is to show how the Theano framework can be used for simple and concise implementation of complex differential geometry algorithms while being able to handle complex and high-dimensional data structures. We show how the Theano framework meets both of these requirements. The framework provides a symbolic language that allows mathematical equations to be directly translated into Theano code, and it is able to perform both fast CPU and GPU computations on high-dimensional data. We show how different concepts from non-linear statistics and differential geometry can be implemented in Theano, and give examples of the implemented theory visualized on landmark representations of Corpus Callosum shapes.

Keywords: Computational Anatomy, Differential Geometry, Non-Linear Statistics, Theano.

7.1 Introduction

Euclidean statistical methods can generally not be used to analyse anatomical shapes because of the non-linearity of shape data spaces. Taking into account non-linearity and curvature of the data space in statistical analysis often requires implementation of concepts from differential geometry.

Numerical implementation of even simple concepts in differential geometry is often a complex task requiring manual implementation of long and complicated expressions involving high-order derivatives. We propose to use the Theano framework in Python to make implementation of differential geometry and non-linear statistics algorithms a simpler task. One of the main advantages of Theano is that it can perform symbolic calculations and

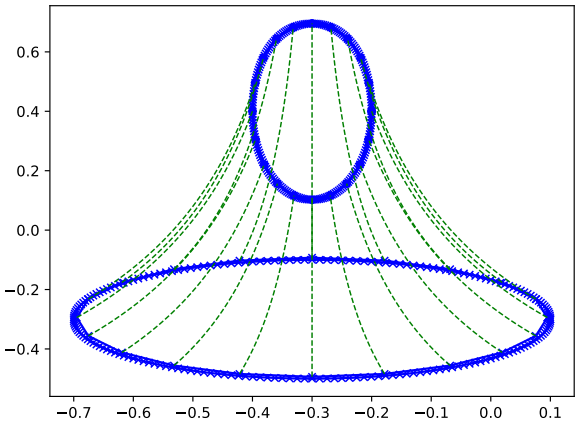


Figure 7.1: Matching of 20000 landmarks on two ellipsoids. Only the matching curve for 20 landmarks have been plotted to make the plot interpretable. The GPU computation is automatic in Theano and no explicit GPU code is used for the implementation.

take symbolic derivatives of even complex constructs such as symbolic integrators. As a consequence, mathematical equations can almost directly be translated into Theano code. For more information on the Theano framework, see [127].

Even though Theano make use of symbolic calculations, it is still able to perform fast computations on high-dimensional data. A main reason why Theano can handle complicated data is the opportunity to use both CPU and GPU for calculations. As an example, Fig. 7.1 shows matching of 20000 landmarks on two different ellipsoids performed on a 40000-dimensional landmark manifold. The matching code was implemented symbolically using no explicit GPU code.

The paper will discuss multiple concepts in differential geometry and non-linear statistics relevant to computational anatomy and provide corresponding examples of Theano implementations. We start by considering simple theoretical concepts and then move to more complex constructions from sub-Riemannian geometry on fiber bundles. Examples of the implemented theory will be shown for landmark representations of Corpus Callosum shapes using the Riemannian manifold structure on the landmark space defined in the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework.

The presented Theano code is available in the Theano Geometry repository,

http://bitbucket.org/stefansommer/theanogeometry,

that includes Theano implementations of additional differential geometry, Lie group, and non-linear statistics algorithms. The described implementations are not specific to the LDDMM landmark manifold used for examples here. The code is completely general and can be directly applied to analysis of data modelled in spaces with different non-linear structures. For more examples of Theano implementation of algorithms directly targeting landmark dynamics, see [4, 5].

The paper is structured as follows. Section 7.1.1 gives a short introduction to the LDDMM manifold. Section 7.2 concerns Theano implementation of geodesics as solution to Hamilton's equations. In Section 7.3, we use Christoffel symbols to define and implement parallel transport of tangent vectors. In Section 7.4, the Fréchet mean algorithm is considered, while stochastics, Brownian motions, and normal distributions are described in Section 7.5. Section 7.6 gives an example of calculating sample mean and covariance by estimating the Fréchet mean on the frame bundle. The paper ends with concluding remarks.

7.1.1 Background

The implemented theory is applied to data on a landmark manifold defined in the LDDMM framework [135]. More specifically, we will exemplify the theoretical concepts with landmark representations of Corpus Callosum (CC) shapes.

Consider a landmark manifold, $\mathcal{M}$, with elements $q = (x_1^1, x_1^2, \ldots, x_n^1, x_n^2)$ as illustrated in Fig. 7.2. In the LDDMM framework, deformation of shapes are modelled as a flow of diffeomorphisms. Let V denote a Reproducing Kernel Hilbert Space (RKHS) of vector fields and let $K \colon V \times V \to \mathbb{R}$ be the reproducing kernel, i.e. a vector field $v \in V$ satisfies $v(q) = \langle K_q, v \rangle_V$ for all $q \in \mathcal{M}$ with $K_q = K(., q)$. Deformation of shapes in $\mathcal{M}$ are then modelled by flows φ_t of diffeomorphisms acting on the landmarks. The flow solves the ordinary differential equation $\partial_t \varphi_t = v(t) \circ \varphi_t$, for $v \in V$. With suitable conditions on K, the norm on V defines a right-invariant metric on the diffeomorphism group that descends to a Riemannian structure on $\mathcal{M}$. The induced cometric $g_q^* \colon T_q^*\mathcal{M} \times T_q^*\mathcal{M} \to \mathbb{R}$ takes the form

$$g_q^*(\nu, \xi) = \sum_{i,j=1}^{n} \nu_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) \xi_j, \tag{7.1}$$

where $\boldsymbol{x}^i = (x_i^1, x_i^2)$ for $i \in \{1, \ldots, n\}$. The coordinate matrix of the cometric is $g^{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ which results in the metric g having coordinates $g_{ij} = K^{-1}(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

In the examples, we use 39 landmarks representing the CC shape outlines,

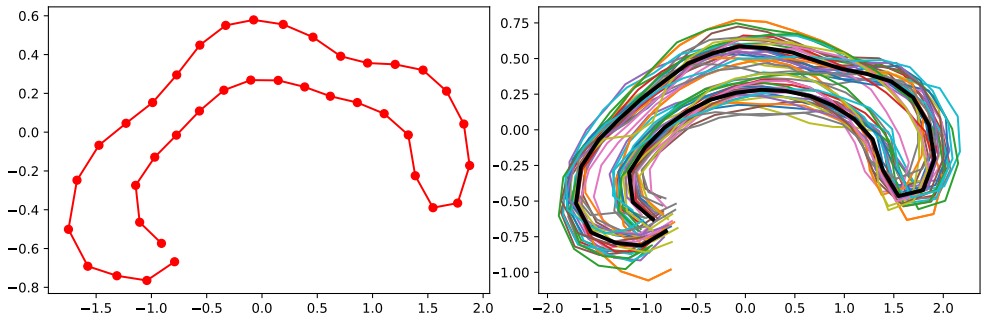


Figure 7.2: (left) An example of a point in $\mathcal{M}$. (right) A subset of the data considered in the examples of this paper. The black curve represents the mean CC of the data.

and the kernel used is a Gaussian kernel defined by

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}\right)$$

with variance parameter σ set to the average distance between landmarks in the CC data. Samples of CC outlines are shown in the right plot of Fig. 7.2.

7.2 Geodesics

Geodesics on $\mathcal{M}$ can be obtained as the solution to Hamilton's equations used in Hamiltonian mechanics to describe the change in position and momentum of a particle in a physical system. Let (U, φ) be a chart on $\mathcal{M}$ and assume $(\mathcal{M}, g)$ is a Riemannian manifold. The Hamiltonian H describes the total amount of energy in the physical system. From the cometric g^*, the Hamiltonian can be defined as $H(q,p) = \frac{1}{2}p^T g_q^* p$, where $g_q^* = (g^{ij})$ is the component matrix of g^* at q. Hamilton's equations are given as the system of ordinary differential equations

$$dq_t = \nabla_p H(q,p), \quad dp_t = -\nabla_q H(q,p).$$

Using the symbolic derivative feature of Theano, the system of ODE's can be represented and discretely integrated with the following code snippet:

```
"""
Hamiltonian function and equations
"""
# Hamiltonian function:
H = lambda q,p: 0.5*T.dot(p,T.dot(gMsharp(q),p))
# Hamiltonian equations:
dq = lambda q,p: T.grad(H(q,p),p)
```

```
dp = lambda q,p: -T.grad(H(q,p),q)

def ode_Ham(t,x):
    dqt = dq(x[0],x[1])
    dpt = dp(x[0],x[1])
    return T.stack((dqt,dpt))
# Geodesic:
Exp = lambda q,v: integrate(ode_Ham,T.stack((q,gMflat(v))))
```

where `gMflat` is the $\flat$ map turning tangent vectors in $T\mathcal{M}$ to elements in $T^\star\mathcal{M}$. `integrate` denotes a function that integrates the ODE by finite time discretization. For the examples considered here, we use a simple Euler integration method. Higher-order integrators are available in the implemented repository mentioned in Section 7.1. A great advantage of Theano is that such integrators can be implemented symbolically as done below using a symbolic `for`-loop specified with `theano.scan`. The actual numerical scheme is only available after asking Theano to compile the function.

```
"""
Numerical Integration Method
"""
def integrator(ode_f):
    def euler(*y):
        t = y[-2]
        x = y[-1]
        return (t+dt,x+dt*ode_f(*y))
    return euler

def integrate(ode,x):
    (cout, updates) = theano.scan(fn=integrator(ode),
          outputs_info=[x],sequences=[*y], n_steps=n_steps)
    return cout
%\end{minted}
```

In the above, `integrator` specifies the chosen integration method, in this example the Euler method. Since the `integrate` function is a symbolic Theano function, symbolic derivatives can be obtained for the integrator, allowing e.g. gradient based optimization for the initial conditions of the ODE. As the derivatives of the integration schemes are symbolic, the schemes remain compatible.

An example of a geodesic found as the solution to Hamilton's equations is visualized in the right plot of Fig. 7.3. The initial point $q_0 \in \mathcal{M}$ was set to the average CC for the data shown in Fig. 7.2 and the initial tangent vector $v_0 \in T_{q_0}\mathcal{M}$ was given as the tangent vector plotted in Fig. 7.3.

The exponential map, $\exp_x \colon T_x\mathcal{M} \to \mathcal{M}$, $x \in \mathcal{M}$ is defined as $\exp_x(v) = \gamma_1^v$, where γ_t^v, $t \in [0,1]$ is a geodesic with $\dot{\gamma}_0^v = v$. The inverse of the exponential map is called the logarithm map, denoted $\log$. Given two points $q_1, q_2 \in \mathcal{M}$, the logarithm map retuns the tangent vector $v \in T_{q_1}\mathcal{M}$ that re-

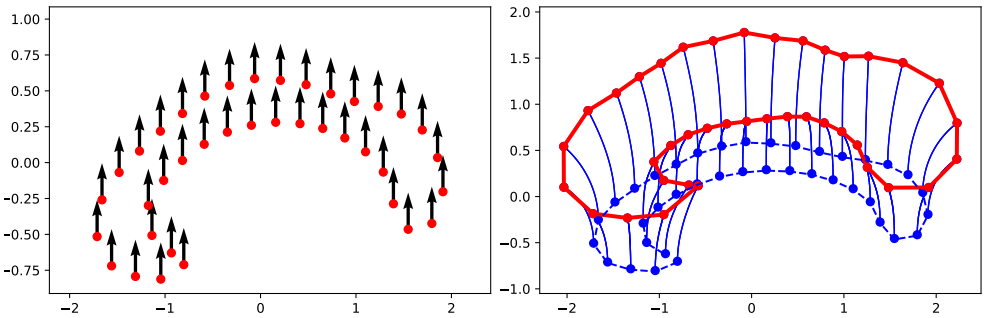


Figure 7.3: (left) The initial point and tangent vector for the geodesic. (right) A geodesic obtained as solution to Hamilton's equations.

sults in the minimal geodesic from q_1 to q_2, i.e. v satisfies $\exp_{q_1}(v) = q_2$. The logarithm map can be implemented using derivative based optimization by taking a symbolic derivative of the exponential map, `Exp`, implemented above:

```
"""
Logarithm map
"""
# Loss function for landmarks:
loss = lambda v,q1,q2: 1./d*T.sum(T.sqr(Exp(q1,v)-q2))
dloss = lambda v,q1,q2: T.grad(loss(v,q1,q2),v)
# Logarithm map: (v0 initial guess)
Log = minimize(loss, v0, jac=dloss, args=(q1,q2))
%\end{minted}
```

The use of the derivative features provided in Theano to take symbolic derivatives of a discrete integrator makes the implementation of the logarithm map extremely simple. The actual compiled code internally in Theano corresponds to a discrete backwards integration of the adjoint of the Hamiltonian system. An example of matching shapes by the logarithm map was shown in Fig. 7.1. Here two ellipsoids of 20000 landmarks were matched by applying the above `Log` function.

7.3 Christoffel Symbols

We here describe how Christoffel symbols can be computed and used in the Theano framework. A connection ∇ defines links between tangent spaces on $\mathcal{M}$ and describes how tangent vectors for different tangent spaces relate. Let (U, φ) denote a coordinate chart on $\mathcal{M}$ with basis coordinates ∂_i, $i = 1, \ldots, d$. The connection ∇ is uniquely described by its Christoffel symbols, Γ_{ij}^k, defined as $\nabla_{\partial_i}\partial_j = \Gamma_{ij}^k \partial_k$.

An example of a frequently used connection is the Levi-Civita connection for Riemannian manifolds. Based on the metric g on $\mathcal{M}$, the Levi-Civita Christoffel symbols are found by

$$\Gamma_{ij}^k = \frac{1}{2} g^{kl}(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}). \tag{7.2}$$

The Theano implementation below of the Christoffel symbols directly translates (7.2) into code:

```
"""
Christoffel Symbols
"""
## Cometric:
gsharp = lambda q: T.nlinalg.matrix_inverse(g(q))
## Derivative of metric:
Dg = lambda q: T.jacobian(g(q).flatten(),q).reshape((d,d,d))
## Christoffel symbols:
Gamma_g = lambda q: 0.5*(T.tensordot(gsharp(q),Dg(q),axes =
    [1,0])\
    + T.tensordot(gsharp(q),Dg(q),axes = [1,0]).dimshuffle
        (0,2,1)\
    - T.tensordot(gsharp(q),Dg(q),axes = [1,2]))
%\end{minted}
```

The connection, ∇, and Christoffel symbols, Γ_{ij}^k, can be used to define parallel transport of tangent vectors on $\mathcal{M}$. Let $\gamma\colon I \to \mathcal{M}$ be a curve and let $t_0 \in I$. A vector field V is said to be parallel along γ if the covariant derivative of V along γ is zero, i.e. $\nabla_{\dot\gamma_t} V = 0$. For a tangent vector $v_0 = v_0^i \partial_i \in T_{\gamma_{t_0}}\mathcal{M}$, there exists a unique parallel vector field V along γ s.t. $V_{t_0} = v_0$. Assume $V_t = v^i(t)\partial_i$, then the vector field V is parallel along γ if the coordinates follows the differential equation,

$$\dot v^k(t) + \Gamma_{ij}^k(\gamma_t)\dot\gamma_t^i v^j(t) = 0, \tag{7.3}$$

with initial values $v^i(0) = v_0^i$. In Theano code the ODE can be written as,

```
"""
Parallel transport
"""
def ode_partrans(gamma,dgamma,t,x):
   dpt = - T.tensordot(T.tensordot(dgamma, Gamma_gM(gamma),
                              axes = [0,1]),x, axes = [1,0])
   return dpt

pt = lambda v,gamma,dgamma: integrate(ode_partrans,v,gamma,
    dgamma)
%\end{minted}
```

Let q_0 be the mean CC plotted in Fig. 7.3 and consider $v_1, v_2 \in T_{q_0}\mathcal{M}$ s.t. v_1 is the vector consisting of 39 copies (one for each landmark) of $e_1 = (1, 0)$ and

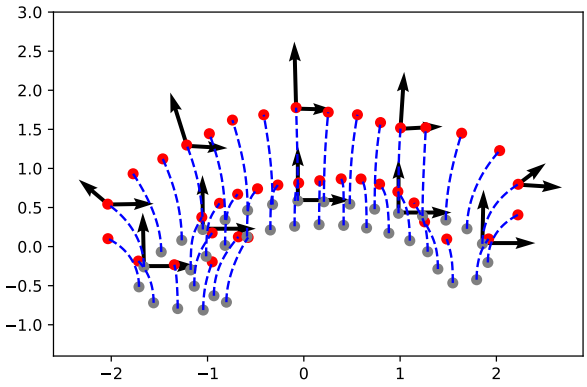


Figure 7.4: Example of parallel transport of basis vectors v_1, v_2 along the geodesic with initial values q_0, v_2. The parallel transported vectors are only plotted for 5 landmarks.

v_2, the vector of 39 copies of $e_2 = (0, 1)$. The tangent vector v_2 is shown in Fig. 7.3. Define γ as the geodesic calculated in Section 7.2 with initial values (q_0, v_2). The parallel transport of v_1, v_2 along γ is visualized in Fig 7.4. To make the plot easier to interpret, the parallel transported vectors are only shown for five landmarks.

7.4 Fréchet Mean

The Fréchet mean [37] is a generalization of the Euclidean mean value to manifolds. Let d be a distance map on $\mathcal{M}$. The Fréchet mean set is defined as $F(x) = \arg\min_{y \in \mathcal{M}} \mathbb{E}d(y, x)^2$. For a sample of data points $x_1, \ldots, x_n \in \mathcal{M}$, the empirical Fréchet mean is

$$F_{\bar{x}} = \arg\min_{y \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} d(y, x_i)^2. \tag{7.4}$$

Letting d be the Riemannian distance function determined by the metric g, the distance can be formulated in terms of the logarithm map, defined in Section 7.2, as $d(x, y) = \| \log(x, y) \|^2$. In Theano, the Fréchet mean can be obtained by optimizing the function implemented below, again using symbolic derivatives.

```
"""
Frechet Mean
"""
def Frechet_mean(q,y):
    (cout,updates) = theano.scan(fn=loss, non_sequences=[v0,q
        ],
```

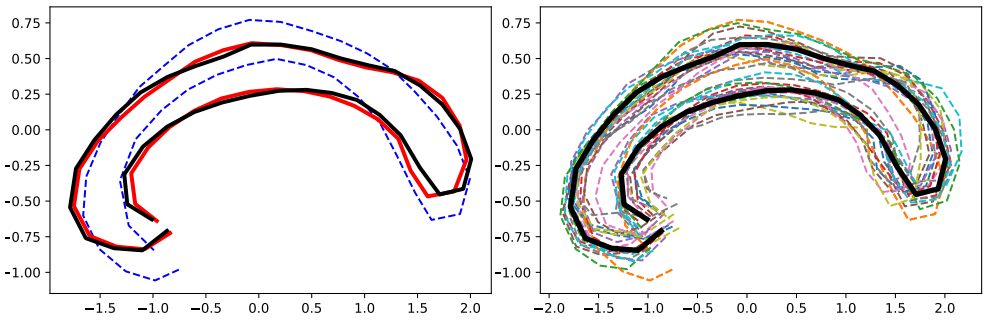


Figure 7.5: (left) The estimated empirical Fréchet mean (black), the initial value (blue) and the Euclidean mean of the 20 samples (red). (right) Plot of the 20 samples of CC, with the Fréchet mean shown as the black curve.

```
            sequences=[y], n_steps=n_samples)
    return 1./n_samples*T.sum(cout)
dFrechet_mean = lambda q,y: T.grad(Frechet_mean(q,y),q)
%\end{minted}
```

The variable `v0` denotes the optimal tangent vector found with the `Log` function in each iteration of the optimization procedure.

Consider a sample of 20 observations of the CC data shown in the right plot of Fig. 7.5. To calculate the empirical Fréchet mean on $\mathcal{M}$, the initial point $q_0 \in \mathcal{M}$ was set to one of the CC observations plotted in the left plot of Fig. 7.5. The result of the optimization is shown in Fig. 7.5 (bold outline).

So far we have shown how Theano can be used to implement simple and frequently used concepts in differential geometry. In the following sections, we will exemplify how Theano can be used for stochastic dynamics and for implementation of more complex concepts from sub-Riemannian geometry on the frame bundle of $\mathcal{M}$.

7.5 Normal Distributions and Stochastic Development

We here consider Brownian motion and normal distributions on manifolds. Brownian motion on Riemannian manifolds can be constructed in several ways. Here, we consider two definitions based on stochastic development and coordinate representation of Brownian motion as an Itô SDE. The first approach [34] allows anisotropic generalizations of the Brownian motion [122, 118] as we will use later.

Stochastic processes on $\mathcal{M}$ can be defined by transport of processes from $\mathbb{R}^m$, $m \leq d$ to $\mathcal{M}$ by the stochastic development map. In order to describe stochastic development of processes onto $\mathcal{M}$, the frame bundle has to be considered.

The frame bundle, $F\mathcal{M}$, is the space of points $u = (q, \nu)$ s.t. $q \in \mathcal{M}$ and ν is a frame for the tangent space $T_q\mathcal{M}$. The tangent space of $F\mathcal{M}$, $TF\mathcal{M}$, can be split into a vertical subspace, $VF\mathcal{M}$, and a horizontal subspace, $HF\mathcal{M}$, i.e. $TF\mathcal{M} = VF\mathcal{M} \oplus HF\mathcal{M}$. The vertical space, $VF\mathcal{M}$, describes changes in the frame ν, while $HF\mathcal{M}$ defines changes in the point $x \in \mathcal{M}$ when the frame ν is fixed in the sense of having zero acceleration measured by the connection. The frame bundle can be equipped with a sub-Riemannian structure by considering the distribution $HF\mathcal{M}$ and a corresponding degenerate cometric $g^*_{F\mathcal{M}} \colon TF\mathcal{M}^* \to HF\mathcal{M}$. Let (U, φ) denote a chart on $\mathcal{M}$ with coordinates $(x^i)_{i=1,\dots,d}$ and coordinate frame $\partial_i = \frac{\partial}{\partial x^i}$ for $i = 1, \dots, d$. Let ν_α $\alpha = 1, \dots, d$ denote the basis vectors of the frame ν. Then (q, ν) have coordinates (q^i, ν^i_α) where $\nu_\alpha = \nu^i_\alpha \partial_i$ and ν^α_i defines the inverse coordinates of ν_α. The coordinate representation of the sub-Riemannian cometric is then given as

$$(g_{F\mathcal{M}})^{ij} = \begin{pmatrix} W^{-1} & -W^{-1}\Gamma^T \\ -\Gamma W^{-1} & \Gamma W^{-1}\Gamma^T \end{pmatrix}, \tag{7.5}$$

where W is the matrix with components $W_{ij} = \delta_{\alpha\beta}\nu^\alpha_i \nu^\beta_j$ and $\Gamma = (\Gamma^{h\gamma}_j)$ for $\Gamma^{h\gamma}_j = \Gamma^h_{ji}\nu^i_\gamma$ with Γ^h_{ji} denoting the Christoffel symbols for the connection, ∇. The sub-Riemannian structure restricts infinitesimal movements to be only along horizontal tangent vectors. Let $\pi^*_\nu \colon T_\nu\mathcal{M} \to H_\nu F\mathcal{M}$ be the lift of a tangent vector in $T\mathcal{M}$ to its horizontal part and let $e \in \mathbb{R}^d$ be given. A horizontal vector at $u = (q, \nu)$ can be defined as the horizontal lift of the tangent vector $\nu e \in T_q\mathcal{M}$, i.e. $H_e(u) = (\nu e)^*$. A basis for the horizontal subspace at $u \in F\mathcal{M}$ is then defined as $H_i = H_{e_i}(u)$, where $e_1, \dots, e_d$ denote the canonical basis of $\mathbb{R}^d$.

Let W_t denote a stochastic process on $\mathbb{R}^m$, $m \leq d$. A stochastic process U_t on $F\mathcal{M}$ can be obtained by the solution to the stratonovich stochastic differential equation, $dU_t = \sum_{i=1}^m H_i(U_t) \circ dW^i_t$, with initial point $u_0 \in F\mathcal{M}$. A stochastic process on $\mathcal{M}$ can then be defined as the natural projection of U_t to $\mathcal{M}$. In Theano, the stochastic development map is implemented as

```
"""
Stochastic Development
"""
def sde_SD(dWt,t,q,nu):
   return T.tensordot(Hori(q,nu), dWt, axes = [1,0])
stoc_dev = lambda q,u,dWt: integrate_sde(sde_SD,
                  integrator_stratonovich,q,u,dWt)[1]
%\end{minted}
```

Here, `integrate_sde` is a function performing stochastic integration of the SDE. The `integrate_sde` is defined in a similar manner as `integrate` described in Section 7.2. In Fig. 7.6 is given an example of stochastic development of a stochastic process W_t in $\mathbb{R}^2$ to the landmark manifold. Notice that for $m < d$, only the first m basis vectors of the basis H_i is used in the stochastic development.

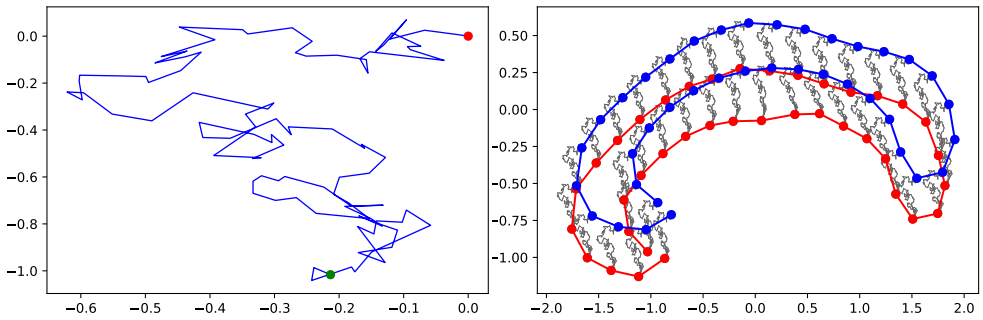


Figure 7.6: (left) Stochastic process W_t on $\mathbb{R}^2$. (right) The stochastic development of W_t on $\mathcal{M}$. The blue points represents the initial point chosen as the mean CC. The red points visualize the endpoint of the stochastic development.

Given the stochastic development map, Brownian motions on $\mathcal{M}$ can be defined as the projection of the stochastic development of Brownian motions in $\mathbb{R}^d$. Defining Brownian motions by stochastic development makes it possible to consider Brownian motions with anisotropic covariance by choosing the initial frame as not being orthonormal.

However, if one is only interested in isotropic Brownian motions, a different definition can be applied. In [120], the coordinates of a Brownian motion is defined as solution to the Itô integral,

$$dq_t^i = -\frac{1}{2} g_q^{kl} \Gamma_{kl}^i dt + \sqrt{g_q^*}^i dW_t. \tag{7.6}$$

This stochastic differential equation is implemented in Theano by the following code.

```
"""
Brownian Motion in Coordinates
"""
def sde_Brownian_coords(dW,t,q):
    gMsharpq = gMsharp(q)
    X = theano.tensor.slinalg.Cholesky()(gMsharpq)
    det = T.tensordot(gMsharpq,Gamma_gM(q),((0,1),(0,1)))
    sto = T.tensordot(X,dW,(1,0))
    return (det,sto,X)
Brownian_coords = lambda x,dWt: integrate_sde(
    sde_Brownian_coords,
                                        integrator_ito,x,dWt)
%\end{minted}
```

An example of an isotropic Brownian motion found by the solution of (7.6) is shown in Fig. 7.7.

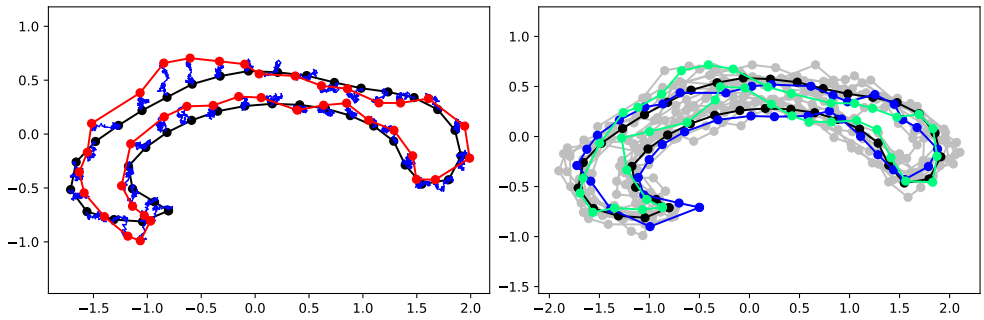


Figure 7.7: (left) Brownian motion on $\mathcal{M}$. (right) Samples drawn from an isotropic normal distribution defined as the transition distribution of a Brownian motion obtained as a solution to (7.6).

In Euclidean statistical theory, a normal distribution can be considered as the transition distribution of a Brownian motion. A similar definition was described in [122]. Here a normal distribution on $\mathcal{M}$ is defined as the transition distribution of a Brownian motion on $\mathcal{M}$. In Fig. 7.7 is shown samples drawn from a normal distribution on $\mathcal{M}$ with mean set to the average CC shown in Fig. 7.2 and isotropic covariance. The Brownian motions are in this example defined in terms of (7.6).

7.6 Fréchet Mean on Frame Bundle

A common task in statistical analysis is to estimate the distribution of data samples. If the observations are assumed to be normally distributed, the goal is to estimate the mean vector and covariance matrix. In [122], it was proposed to estimate the mean and covariance of a normal distribution on $\mathcal{M}$ by the Fréchet mean on the frame bundle.

Consider Brownian motions on $\mathcal{M}$ defined as the projected stochastic development of Brownian motions on $\mathbb{R}^d$. A normal distribution on $\mathcal{M}$ is given as the transition distribution of a Brownian motion on $\mathcal{M}$. The initial point for the stochastic development, $u_0 = (q_0, \nu_0) \in F\mathcal{M}$, corresponds to the mean and covariance, i.e. $q_0 \in \mathcal{M}$ denotes the mean shape and ν_0 the covariance of the normal distribution. As a consequence, normal distributions with anisotropic covariance can be obtained by letting ν_0 be a non-orthonormal frame.

In Section 7.4, the Fréchet mean on $\mathcal{M}$ was defined as the point $y \in \mathcal{M}$, minimizing the average geodesic distance to the observations. However, as only a sub-Riemannian structure is defined on $F\mathcal{M}$, the logarithm map does not exist and hence the geodesic distance cannot be used to define the Fréchet mean on $F\mathcal{M}$. Instead, the distance function will be defined based on the

most probable paths (MPP) defined in [118]. In this section, a slightly different algorithm for estimating the mean and covariance for a normal distribution is proposed compared to the one defined in [122].

Let $u = (q, \nu) \in FM$ be given such that q, ν is the mean and covariance of a normal distribution on $\mathcal{M}$. Assume that observations $y_1, \ldots, y_n \in \mathcal{M}$ have been observed and let $p_1, \ldots, p_n \in H^*FM$. The Fréchet mean on FM can then be obtained by optimizing,

$$F_{FM} = \underset{(u,p_1,\ldots,p_n)}{\arg\min} \frac{1}{n}\sum_{i=1}^{n} \|p_i\|^2_{g^*_{FM}} + \frac{\lambda}{n}\sum_{i=1}^{n} d_{\mathcal{M}}(\pi(\exp_u(p_i^\sharp)), y_i)^2 - \frac{1}{2}\log(\det \nu),$$

where $\sharp$ denotes the sharp map on FM changing a momentum vector in T^*FM to the corresponding tangent vector in TFM. The point of minimizing with respect to the momentum vector $p_1, \ldots, p_n$ is that the geodesics, $\exp_u(p_i^\sharp)$, becomes MPPs on FM, i.e. the first term penalizes the momentum vector. The second term decreases the distance of the mean to each data point as in the empirical Fréchet mean on $\mathcal{M}$, while the last term ensures that the covariance frame does not tend to 0.

The Fréchet mean on FM is implemented in Theano and numpy as,

```
"""
Frechet Mean on FM
"""
detg = lambda q,nu: T.nlinalg.Det()(T.tensordot(nu.T,
             T.tensordot(gM(q),nu,axes=(1,0)),axes=(1,0)))

lossf = lambda q1,q2: 1./d.eval()*np.sum((q1-q2)**2)

def Frechet_meanFM(u,p,y0):
    q = u[0:d.eval()]
    nu = u[d.eval():].reshape((d.eval(),rank.eval()))
    for i in range(n_samples):
       distv[i,:] = lossf(Expfmf(u,p[i,:])[0:d.eval()],y0)
       normp[i,:] = 2*Hfm(u,p[i,:]) # Hamiltonian on FM
    return 1./n_samples*np.sum(normp)
     +lambda0/n_samples*np.sum(distv**2)-1./2*np.log(detgf(x,
        u))
%\end{minted}
```

7.7 Conclusion

In the paper, it has been shown how different concepts in differential geometry and non-linear statistics can be implemented using the Theano framework. Integration of geodesics, computation of Christoffel symbols and parallel transport, stochastic development and Fréchet mean estimation were considered and demonstrated on landmark shape manifolds. In addition, we

showed how the Fréchet mean on the frame bundle $F\mathcal{M}$ can be computed for estimating the mean and covariance of an anisotropic normal distribution on $\mathcal{M}$.

Theano has, for the cases shown in this paper, been a very efficient framework for implementation of differential geometry concepts and for non-linear statistics in a simple and concise way yet allowing efficient and fast computations. We emphasize that Theano is able to perform calculations on high-dimensional manifolds using either CPU or GPU computations. In the future, we plan to extend the presented ideas to derive Theano implementations of differential geometry concepts in more general fiber bundle and Lie group geometries.

Acknowledgements. This work was supported by Centre for Stochastic Geometry and Advanced Bioimaging (CSGB) funded by a grant from the Villum foundation.

CHAPTER 8

Latent Space Non-Linear Statistics

The following manuscript, made in collaboration with Tom Fletcher, Sarang Joshi and Stefan Sommer, is available at ArXiv as submission,

- L. Kühnel, T. Fletcher, S. Joshi, and S. Sommer. Latent space non-linear statistics. *arXiv preprint arXiv:1805.07632*, 2018

Based on a pre-trained latent space of a variational autoencoder, we perform analyses of the projection of newly observed data to the lower dimensional latent space representation. It was shown simultaneously by [114, 24, 8] that the latent space has a non-linear Riemannian structure as the pull-back of the metric structure of the Euclidean ambient space. Hence, non-linear statistical methods should be used to analyse the latent space data representations.

Latent Space Non-Linear Statistics

Line Kühnel[1], Tom Fletcher[2], Sarang Joshi[3], and Stefan Sommer[1]

Department of Computer Science (DIKU), University of Copenhagen, Denmark[1]

Department of Electrical and Computer Engineering / Department of Computer Science, University of Virginia, USA[2]

Department of Biomedical Engineering, University of Utah, USA[3]

Abstract

Statistical analysis of high-dimensional data is usually infeasible and computationally difficult to conduct. Deep generative models, e.g., variational autoencoders and generative adversarial networks, train a lower dimensional latent representation of the data space primarily used for generating data samples. However, the low dimensionality of latent space makes the space optimal for analysing high-dimensional data. The linear Euclidean geometry of the high-dimensional data space pulls back to a nonlinear Riemannian geometry on latent space where classical linear statistical techniques are no longer applicable. The paper shows how analysis of data in their latent space representation is performed using techniques from the field of nonlinear manifold statistics. Nonlinear manifold statistics provide generalisations of Euclidean statistical notions including means, principal component analysis, and maximum likelihood estimation of parametric distributions. Introduction to estimation procedures on latent space are considered, and the computational complexity of using geometric algorithms with high-dimensional data addressed by training a separate neural network to approximate the Riemannian metric and cometric tensor capturing the shape of the learned data manifold.

8.1 Introduction

The Riemannian geometry of latent models, provided by deep generative models, have recently been explored in [114], [24] and [8]. The mapping $f : Z \to X$, from latent space Z to the data space X, constitutes an embedding of Z into X under mild assumptions on the network architecture. The embedding allows the image $f(Z)$ to inherit the Riemannian metric and hence the geometry from the Euclidean ambient space X. Equivalently, the

metric structure of X pulls back via f to a nonlinear Riemannian structure on Z. The above papers explore aspects of this geometry including numerical schemes for geodesic integration, parallel transport, Fréchet mean estimation, simulation of Brownian motion, and interpolation. With this paper, we focus on performing subsequent statistics after learning the latent representation including the embedding f.

Variational autoencoders learn a latent space Z in which training data follows a normal distribution. Performing statistical analysis on the data used for training the latent space Z is hence unnatural. On the contrary, we aim at learning the latent space representation Z given a training dataset and then use constructions, tools, and methods from nonlinear statistics [101] to perform statistical analysis on newly observed data in the latent representation.

Deep generative models are excellent tools for learning the intrinsic geometry of a low-dimensional data manifold $f(Z)$, subspace of the data space X. When the highest modes of data variation can be expressed in a few intrinsic dimensions, statistical analyses exploiting the lower dimensionality can be more efficient than conducting analyses directly in the high-dimensional data space. Performing statistical analysis in lower-dimensional manifolds learned with deep generative models, we simultaneously adapt the statistics to the intrinsic geometry of the data manifold, exploit the compact representation, and avoid unnecessary dimensions in the high-dimensional space X affecting the statistical analysis. As an example, we compare two-sample test in the full data space with a generalised two-sample test in the non-linear lantent space. The presented example shows that the test in the latent space results in a more significant test of the two generated populations than the test in the high-dimensional data space.

Exemplified on three datasets, synthetic data on the sphere $\mathbb{S}^2$ for visualization, the MNIST digits dataset, and landmark representation of diatoms, we show how statistical procedures such as principal component analysis can be performed on the latent space. We will subsequently define and infer parameters of geometric distributions allowing the definition and inference of maximum likelihood estimates via simulation of diffusion processes. Both VAEs and GANs themselves learn distributions representing the input training data. The aim is to perform nonlinear statistical analyses for data independent of the training data and with a different distribution, but which are elements of the same low-dimensional manifold of the data space. The latent representation can in this way be learned unsupervised from large numbers of unlabeled training samples where subsequently the low-dimensional space can be used to perform statistical analysis on datasets of a small sample size. This setting occurs for example in medical imaging where brain MR scans are abundant while controlled disease progression studies are of a much smaller sample size. The approach resembles the common task of using principal component analysis to represent data in the span of fewer principal eigenvectors, with the critical difference that in the present case a

nonlinear manifold is learned using deep generative models instead of standard linear subspace approximation.

The field of nonlinear statistics provide generalizations of statistical constructions and tools from linear Euclidean vector spaces to Riemannian manifolds. Such constructs, e.g., the mean value, often have many equivalent definitions in Euclidean space. However, nonlinearity and curvature generally break this equivalence leading to a plethora of different generalizations. For this reason, we here focus on a subset of selected methods to exemplify the use of nonlinear statistical tools in the latent space setting: Principal component analysis on manifolds with the principal geodesic analysis (PGA, [36]), inference of maximum likelihood means from intrinsic diffusion processes [122] and a generalisation of Hotelling two-sample test [16].

The learned manifold defines a Riemannian metric on the latent representation. The often high dimensionality of the data manifold makes it computationally costly to evaluate the metric. The computational cost is severely amplified when calculating higher-order derivatives needed for geometric concepts such as the curvature tensor and the Christoffel symbols that are crucial for numerical integration of geodesics and simulation of sample paths for Brownian motions. We present a new method for handling the computational complexity of evaluating the metric by training a second neural network to approximate the local metric tensor of the latent space thereby achieving a massive speed up in the implementation of the geometric and nonlinear statistical algorithms.

The paper thus presents the following contributions:

1. we couple tools from nonlinear statistics with deep end-to-end differentiable generative models for analyzing data using a pre-trained low-dimensional latent representation,

2. we show how an additional neural network can be trained to learn the metric tensor and thereby greatly speed up the computations needed for the nonlinear statistics algorithms,

3. we develop a method for maximum likelihood estimation of diffusion processes in the latent geometry and use this to estimate ML means from Riemannian Brownian motions.

4. we give an example of two-sample test for which the latent space representation results in a more significant test than the two-sample test in the high-dimensional data space.

We show examples of the presented methods on latent geometries learned from synthetic data in $\mathbb{R}^3$, on the MNIST dataset and on data of diatoms. The statistical computations are implemented in the Theano Geometry package [70] that using the automatic differentiation features of Theano [127] allows for easy and concise expression of differential geometry concepts.

The paper starts with a brief description on latent space geometry based on the papers [114], [24], and [8]. We then discuss the definition of mean values in the nonlinear latent geometry, the use of the principal geodesic analysis (PGA) procedure, and make a description of a generalised two-sample test on nonlinear spaces. We end the paper with developing a scheme for maximum likelihood estimation of parameters with Riemannian Brownian motion using a diffusion bridge sampling scheme before performing experiments on the described datasets.

8.2 Latent Space Geometry

Deep generative models such as generative adversarial networks (GANs, [42]) and autoencoders/variational autoencoders (VAEs, [13]) learn mappings from a latent space Z to the data space X. In the VAE case, the decoder mapping $f\colon Z \to X$ describes the mean of the data distribution, $P(X|z) = \mathcal{N}(X \mid f(z), \sigma(z)^2 I)$, and is complemented by an encoder $h\colon X \to Z$. Both Z and X are Euclidean spaces, with dimension d and n respectively and generally $d \ll n$. When the push forward f_*, and the differential df of f, is of rank d for any point z, the image $f(Z)$ in X is an embedded differentiable manifold of dimension d. We denote this manifold by M. Generally, for deep models, f is nonlinear making M a nonlinear manifold. An example of a trained manifold with a VAE is shown in Figure 8.1. Here we simulate synthetic data on the sphere $\mathbb{S}^2$ by the transition distribution of a Riemannian Brownian motion starting at the north pole. The learned submanifold approximates $\mathbb{S}^2$ on the northern hemisphere containing the greatest concentration of samples.

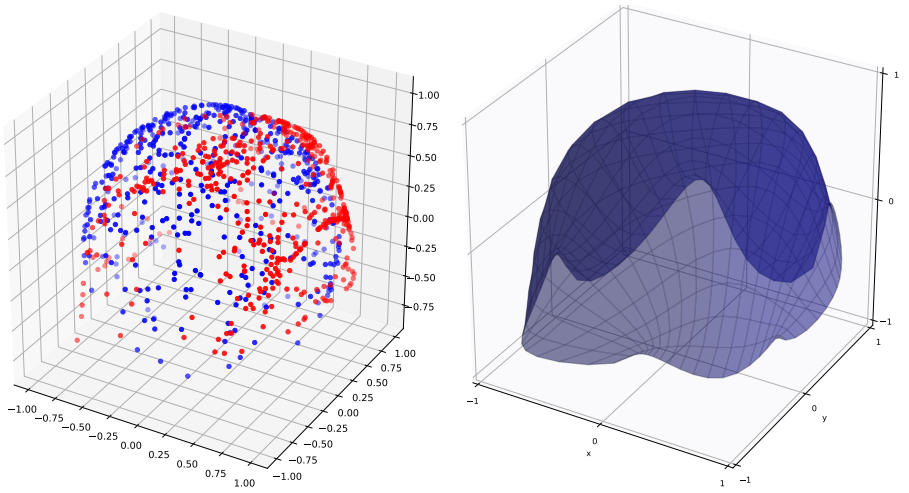


Figure 8.1: (left) Samples from the data distribution (blue) with corresponding predictions from the VAE (red). (right) The trained manifold.

The learned manifold M inherits differential and geometric structure from X. In particular, the standard Euclidean inner product restricts to tangent spaces T_xM for $x \in M$ to give a Riemannian metric g on M, i.e. for $v, w \in T_xM$, $g(v,w) = \langle v, w \rangle = v^T w$. Locally, we invert f to obtain charts on M, and get the standard expression $g_{ij}(z) = \langle \partial_{z_i} f, \partial_{z_j} f \rangle$ for the metric tensor in Z coordinates. Using Jacobian matrix $Jf = (\partial_{z_i} f^j)^i_j$, the matrix expression of $g(z)$ is $g(z) = (Jf(z))^T Jf(z)$. The metric tensor on Z can be seen as the pullback f^*g of the Riemannian metric on X.

The geometry of latent spaces was explored in [114]. In addition to setting up the geometric foundation, the paper developed efficient algorithms for geodesic integration, parallel transport, and Fréchet mean estimation on the latent space. The algorithms make particular use of the encoder function $h\colon X \to Z$ trained as part of the VAEs. Instead of explicitly computing Christoffel symbols for geodesic integration, the presence of h allows steps of the integration algorithm to be taken in X and then subsequently mapped back to Z. Execution speed increases significantly when avoiding computation of Christoffel symbols, a critical improvement for the heavy computations involved with the typically high dimensions of X. [24] provides additional views on the latent geometry and interpolation examples on the MNIST dataset and robotic arm movements. [8] includes the z-variability of the variance $\sigma(z)$ of VAEs resulting in the inclusion of the Jacobian of σ in the expected metric. In addition, the paper explores random walks in the latent geometry and how to enable meaningful extrapolation of the latent representation beyond the training data.

8.2.1 Latent Data Representations

Given sampled data $y_1, \ldots, y_N$ in X, the aim is here to perform statistical analysis on the data after mapping to the low-dimensional latent space Z. Note that the mapping f can thus be trained unsupervised and afterwards used to perform statistics on new data in the low-dimensional representation. Therefore, the data $y_1, \ldots, y_N$ are generally different from the training data used to train f. In particular, N can be much lower than the size of the training set.

For VAEs, the mapping of y_i to corresponding points in the latter representation z_i is directly available from the encoder function h, i.e. $z_i = h(y_i)$. In more general settings where h is not present, we need to construct z_i from y_i. A natural approach is to define z_i from the optimization problem

$$z_i = \underset{z \in Z}{\arg\min} \, \|f(z) - y_i\|^2 \,. \tag{8.1}$$

This can be seen as a projection from X to M using the Euclidean distance in X.

8.2.2 Geodesics and Brownian Motions

The pullback metric f^*g on Z defines geometric concepts such as geodesics, exponential and logarithm map, and Riemannian Brownian motions on Z. Using f, each of these definitions is equivalently expressed on M viewing it as a submanifold of X with inherited metric. Given $z \in Z$ and $v \in T_zZ$, the exponential map $\mathrm{Exp}_z \colon T_zZ \to Z$ is defined as the geodesic γ_t^v at time $t = 1$ with starting point z and initial velocity v, i.e. $\mathrm{Exp}_z(v) = \gamma_1^v$. The logarithm map $\mathrm{Log} \colon Z \times Z \to TZ$ is the local inverse of Exp: Given two points $z_1, z_2 \in Z$, $\mathrm{Log}_{z_1}(z_2)$, returns the tangent vector $v \in T_{z_1}Z$ defining the minimizing geodesic between z_1 and z_2. The Riemannian metric defines the geodesic distance expressed from the logarithm map by $d(z_1, z_2) = \|\mathrm{Log}_{z_1}(z_2)\|_g$. Using Z as coordinates for M by local inverses of f, the Riemannian Brownian motions on Z, and equivalently on M, is defined by the coordinate expression

$$dz_t^j = -\frac{1}{2} g(z)^{kl} \Gamma_{kl}^j dt + \sqrt{(g(z))^{-1}}^j dB_{t,j}, \tag{8.2}$$

where Γ_{kl}^j denotes Christoffel symbols, g^{-1} the cometric, i.e. the inverse of the metric tensor g, and B_t a standard Brownian motion in $\mathbb{R}^d$. Notice that Einstein notation is used for index summation.

8.3 Computational Representation

While metric computation is easily expressed using automatic differentiation to compute the Jacobian Jf of the embedding map f, the high dimensionality of the data space has a computational cost when evaluating the metric. The computational cost is particularly emphasised when computing higher-order differential concepts such as Christoffel symbols, used for geodesic integration, curvature, and Brownian motion simulation. The reason being the multiple derivatives and metric inverse computations involved. For integration of geodesics and Brownian motion, one elegant way to avoid the computation of Christoffel symbols is to take each step of the integration in the ambient data space of M and map the result back to the latent space using the encoder mapping h [114]. This procedure requires h to be close to the inverse of f restricted to M and limits the method to VAEs where h is trained along with the decoder, f.

We here propose an additional way to allow efficient computations without using the encoder map h. The approach, therefore, works for both GANs and VAEs. The latent space Z is of low dimension, and the only entity needed for encoding the geometry is the metric $g : Z \to \mathrm{Sym}_+(d)$ which to each z assigns a positive symmetric $d \times d$ matrix. $\mathrm{Sym}_+(d)$ has dimension $d(d+1)/2$. Hence, the high dimensionality of the data space does not appear directly when defining the geometry, and X is only used for the actual computation of $g(z)$. We therefore train a second neural network $\tilde{g}$ to act as a function

approximator for g, i.e. we train $\tilde{g}$ to produce an element of $\mathrm{Sym}_+(d)$ that is close to $g(z)$ for each z. Notice that this network does not evaluate a Jacobian matrix when computing $g(z)$, and no derivatives are hence needed for evaluating the metric. The lack of complexity and due to both input and output space of the network being of low dimensions, d and $d(d+1)/2$ respectively, makes the computational effort of evaluating $\tilde{g}$ and Christoffel symbols, computed from $\tilde{g}$, orders of magnitude faster than evaluating g directly. The speedup is especially present when the dimensionality n of X is high compared to d: Integration of the geodesic equation with 100 timesteps in the MNIST case presented later takes ≈ 30 s., when computing the metric from Jf, compared to ≈ 30 ms., when using the second neural network to predict g.

Inverting $\tilde{g}(z)$ is sensitive to the approximation of g provided by $\tilde{g}$. The cometric tensor g^{-1} is therefore more sensitive to the approximation when computed from $\tilde{g}$ than from g itself. This is emphasized when $g(z)$ has small eigenvalues. As a solution, we let the second neural network predict both the metric $g(z)$ and cometric $g(z)^{-1}$. Defining the loss function for training the network, we balance the norm between predicted matrices $\tilde{g}$ and $\tilde{g}^{-1}$. In addition, we ensure that the predicted $\tilde{g}$ and $\tilde{g}^{-1}$ are close to being actual inverses. These observations are expressed in the loss function

$$\begin{aligned}
&\mathrm{loss}_{g,g^{-1}\text{-approximator}}(g_{\text{true}}, g^{-1}_{\text{true}}, g_{\text{predicted}}, g^{-1}_{\text{predicted}}) \\
&\quad = \|g_{\text{true}} - g_{\text{predicted}}\|^2 / \|g_{\text{true}}\|^2 \\
&\qquad + \|g^{-1}_{\text{true}} - g^{-1}_{\text{predicted}}\|^2 / \|g^{-1}_{\text{true}}\|^2 \\
&\qquad + \|g^{-1}_{\text{predicted}} g_{\text{predicted}} - \mathrm{Id}_d\,\|^2,
\end{aligned} \tag{8.3}$$

using Frobenius matrix norms. We train a neural network with two dense hidden layers to minimize (8.3), and use this network for the geometry calculations. The network predicts the upper triangular part of each matrix, and this part is symmetrized to produce $g_{\text{predicted}}$ and $g^{-1}_{\text{predicted}}$. Note that additional methods could be employed to ensure the predicted metric being positive definite, see e.g. [53]. For the presented examples, it is our observations that the loss (8.3) ensures positive definiteness without further measures.

8.4 Non-Linear Latent Space Statistics

We now discuss aspects of nonlinear statistics applicable to the latent geometry setting. We start by focusing on means, particularly Fréchet and maximum likelihood (ML) means, before modeling variation around the mean with the principal geodesic analysis procedure and ending the section with a description of a generalised two-sample test.

8.4.1 Fréchet and ML Means

Fréchet mean [38] of a distribution on M, and its sample equivalent, minimise the expected squared Riemannian distance: $\hat{x} = \arg\min_{x \in M} \mathbb{E}[d(x,y)^2]$ and $\hat{x} = \arg\min_{x \in M} \frac{1}{N} \sum_{i=1}^{N} d(x,y_i)^2$. The standard way to estimate a sample Fréchet mean is to employ an iterative optimisation to minimise the sum of squared Riemannian distances. The Riemannian gradient of the squared distance can be expressed using the Riemannian Log map [101] by $\nabla_x d(x,y)^2 = 2\operatorname{Log}_x(y)$.

The Fréchet mean generalises the Euclidean concept of a mean value as a distance minimiser. In Euclidean space, this is equivalent to the standard Euclidean estimator $\hat{x} = \frac{1}{N}\sum_i y_i$. From a probabilistic viewpoint, the equivalence between the log-density function of a Euclidean normal distribution and the squared distance results in $\hat{x}$ as an ML fit of a normal distribution to data:

$$\hat{x} = \arg\min_x \log p_{\mathcal{N},x}(y), \tag{8.4}$$

with $p_{\mathcal{N},x}(y) \propto \exp(-\frac{1}{2}\|x-y\|^2)$ being the density of a normal distribution with mean x. While the normal distribution does not have a canonical equivalent on Riemannian manifolds, an intrinsic generalisation comes from the transition density of a Riemannian Brownian motion. This density on M arise as the solution to the heat PDE, $\frac{\partial}{\partial t} p_{x,t} = \frac{1}{2}\Delta_g p_{x,t}$, using the Laplace-Beltrami operator Δ_g, or, equivalently, from the law of the Brownian motion started at M. In [122], [95], and [120], this density is used to generalise the ML definition of the Euclidean mean,

$$\hat{x} = \arg\min_x \log p_{x,T}(y). \tag{8.5}$$

for at fixed $T > 0$. We will develop approximation schemes for evaluating the log-density and for solving the optimisation problem (8.5) in Section 8.5.

8.4.2 Principal Component Analysis

Euclidean principal component analysis (PCA) estimates subspaces of the data space that explain the majority of variation in the data, either by maximising variance or minimising residuals. PCA builts around the linear vector space structure and the Euclidean inner product. Defining procedures that resemble PCA for manifold-valued data hence become challenging, as neither inner products between arbitrary vectors nor the concept of linear subspaces are defined on manifolds.

[36] presented a generalised version of Euclidean PCA denoted principal geodesic analysis (PGA). PGA estimates nested geodesic submanifolds of M that capture the most variation of the data projected to each submanifold. The geodesic subspaces hence take the place of the linear subspaces found with the Euclidean PCA.

Let $z_1, \ldots, z_N \in Z$ be latent space representations of the data $y_1, \ldots, y_N$ in M, and let μ be a Fréchet mean of the samples $z_1, \ldots, z_N$. We assume the observations are located in a neighbourhood U of μ where Exp_μ is invertible and the logarithm map, Log_μ, thus well-defined. We search for an orthonormal basis of tangent vectors in $T_\mu Z$ such that for each nested submanifold, $H_k = \text{Exp}_\mu(\text{span}\{v_1, \ldots, v_k\})$, the variance of the data projected on H_k is maximised. The projection map used is based on the geodesic distance, d, and is defined by, $\pi_H(z) = \underset{z_1 \in H}{\arg\min}\, d(z, z_1)^2$.

The tangent vectors $v_1, \ldots, v_k$ in the orthonormal basis of $T_\mu Z$ are found by optimising the Fréchet variance of the projected data on the submanifold H, i.e.

$$v_k = \underset{\|v\|=1}{\arg\max} \sum_{i=1}^{n} d(\mu, \pi_H(z_i))^2, \tag{8.6}$$

where $H = \text{Exp}_\mu(\text{span}\{v_1, \ldots, v_{k-1}, v\})$. For a more detailed description of the PGA procedure, including computational approximations of the projection map in the tangent space of μ, see [36]. In the experiment section, we perform PGA on the manifold defined by the latent space of a deep generative model for the MNIST dataset.

8.4.3 Generalised Two-Sample Test

This section describes another example of the usage of the latent space representation to perform statistical analyses on high-dimensional data. More specifically, we apply a permutation test based on the test statistic from the generalised Hotelling two-sample test presented in [16].

Given two populations, $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_m)$, in the latent space Z, we test the null hypothesis $H_0 \colon \mu_1 = \mu_2$ of equal Fréchet mean against the alternative hypothesis, $H_1 \colon \mu_1 \neq \mu_2$. The test relies on several assumptions including the existence of an element $p \in \mathcal{M}$ such that the exponential chart $\phi^{-1} = \text{Exp}_p$ contains both populations, X and Y, i.e. $Y, X \subset \text{Exp}_p(V)$ for $V \subset \mathbb{R}^d$. The test statistic for the generalised Hotelling two-sample test is given as

$$T_{mn} = (n+m)(\hat{\mu}_1 - \hat{\mu}_2)^T \hat{\Sigma}^{-1} (\hat{\mu}_1 - \hat{\mu}_2) \tag{8.7}$$

where $\hat{\mu}_i$ is the Fréchet sample mean for the i'th population and $\hat{\Sigma}$ denotes the pooled sample covariance,

$$\hat{\Sigma} = (m+n)\left(\frac{1}{n}\hat{\Lambda}_1^{-1}\hat{\Sigma}_1\hat{\Lambda}_1^{-1} + \frac{1}{m}\hat{\Lambda}_2^{-1}\hat{\Sigma}_2\hat{\Lambda}_2^{-1}\right). \tag{8.8}$$

The pooled covariance matrix is based on

$$\hat{\Lambda}_1 = \frac{1}{n}\sum_{i=1}^{n} D_y^2(\|\mathrm{Log}(\phi^{-1}x_i, \phi^{-1}y)\|^2)\,|_{y=\phi\hat{\mu}_1} \text{ and}$$
$$\Sigma_1 = \mathrm{Cov}\left(D_y(\|\mathrm{Log}(\phi^{-1}x_i, \phi^{-1}y)\|^2)\,|_{y=\phi\mu_1}\right)$$

such that $\hat{\Sigma}_1$ denotes the sample covariance of Σ_1. Calculations of the derivatives of the geodesic distance can be found in [102], and [121].

Based on the test statistic T_{mn} the significance of the test is determined based on a permutation test. The permutation test uses the assumption of equal means under the null-hypothesis, i.e. if the null-hyppothesis is true permuting the samples between populations would not change the population means. Running several permutations under the null-hypothesis creates a distribution of the test statistic under the null for which it is possible to obtain a p-value.

8.5 Maximum Likelihood Inference of Diffusions

As in Euclidean statistics, parameters of distributions on manifolds can be inferred from data by maximum likelihood or, from a Bayesian viewpoint, maximum a posteriori. These methods can even be used to define statistical notions as exemplified by the ML mean in Section 8.4. The probabilistic viewpoint relies on the existence of parametric families of distributions in the geometric spaces, and the ability to evaluate likelihoods. One example of such a distribution is the transition distribution of the Riemannian Brownian motion, see, e.g. [52]. In this section, we show how likelihoods of data in the latent space Z under the transition distribution can be evaluated by Monte Carlo sampling of conditioned diffusion bridges. As previous, assume that a separate training dataset has been used to train the geometry of Z. We wish to perform statistical analysis on newly observed data represented by z_i. To determine the transition distribution of a Brownian motion on the data manifold, we apply a conditional diffusion bridge simulation procedure defined in [32]. The following section makes a description of this procedure. The sampling scheme has previously been used for geometric spaces in [5], and [120].

8.5.1 Bridge Simulation and Parameter Inference

Let $z_1, \ldots, z_N \in Z$ be N observations in Z. We assume z_i are time T observations from a Brownian motion, z_t defined by (8.2), on Z started at $x \in Z$. The aim is to optimise for the initial point x by maximising the likelihood of the observed data and thereby find the ML mean (8.5). The mean value of the data distribution is thus defined as the starting point of the process maximising the data likelihood, $L_\theta(z_1, \ldots, z_N) = \prod_{i=1}^{N} p_{T,\theta}(z_i)$, where $p_{T,\theta}(z_i)$ is the

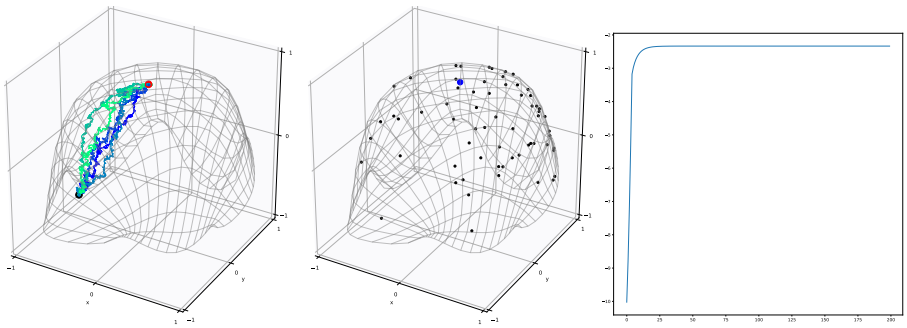


Figure 8.2: (left) Brownian bridge sample paths on the trained data manifold. (middle) The estimated ML mean (blue) from the data (black points). (right) The likelihood values from the MLE procedure.

time T transition density of z_t evaluated at z_i. The difficulty is to determine the transition density $p_{T,\theta}(z_i)$, i.e. the time T density conditional on $z_T = z_i$. In [32] it was shown that this conditional probability can be calculated based on the notion of a guided process

$$d\tilde{z}_t^j = -\frac{1}{2}g(z)^{kl}\Gamma_{kl}^j dt - \frac{\tilde{z}_t^j - z_i^j}{T-t}dt + \sqrt{(g(z))^{-1}}^j dB_t, \tag{8.9}$$

which, without conditioning, almost surely hits the observation z_i at time $t = T$. In fact, the conditional process $z_t|z_T = z_i$ is absolutely continuous with respect to the guided process with Radon-Nikodym derivative, $dP_{z|z_i}/dP_{\tilde{z}} = \varphi(\tilde{z})/E_{\tilde{z}}[\varphi(\tilde{z}_t)]$. Based on the above arguments, an expression of the transition density is

$$p_{T,\theta}(z_i) = \sqrt{\frac{|g(z_i)|}{(2\pi T)^d}} e^{-\frac{\|(x-z_i)^T g(x)(x-z_i)\|^2}{2T}} \mathbb{E}_{\tilde{z}_t}[\varphi(\tilde{z}_t)], \tag{8.10}$$

see [32], and [120] for more details. We use Monte Carlo sampling of $\tilde{z}_t$ to approximate $\mathbb{E}_{\tilde{z}_t}[\varphi(\tilde{z}_t)]$ and hence determine $p_{T,\theta}(z_i)$ by (8.10). The likelihood can then be iteratively optimised to find the ML mean by computing gradients with respect to x.

Figure 8.2 shows sample paths of a Brownian bridge on the trained manifold for the synthetic data on $\mathbb{S}^2$ in addition to the ML estimated mean (middle). The likelihood values at each iteration are plotted in the same figure and illustrates convergence for the MLE procedure.

8.6 Experiments

We give examples of the analyses described above for the MNIST dataset [74] and a dataset of landmark representations of diatoms [54]. The computations

are performed with the Theano Geometry package http://bitbucket.com/stefansommer/theanogeometry/ described in [70]. The package contains implementations of differential geometry concepts and corresponding statistical algorithms.

8.6.1 MNIST

The MNIST dataset consists of images of handwritten digits from 0 to 9 with each observation of dimension 28×28. A VAE is trained on the full dataset providing a 2-dimensional latent space representation. The VAE [13] has one hidden dense layer for both encoder and decoder, each layer containing 256 neurons, and results in a 2d latent space Z.

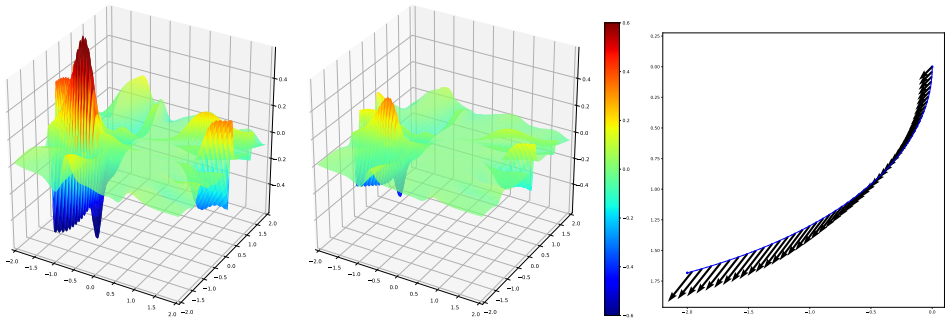


Figure 8.3: (left) Scalar curvature of space Z. (middle) Min. eigenvalue of the Ricci curvature tensor. (right) Parallel transport of a tangent vector in Z. The transported vectors have constant length measured by the Riemannian metric.

Figure 8.3 shows the scalar (left) and minimum Ricci curvature (middle) in a neighbourhood of the origin of Z. Moreover, an example of parallel transport of a tangent vector along a curve in the latent space is presented in the same figure (right). Note that the transported vector has a constant length as measured by the metric g which is not the case for the Euclidean $\mathbb{R}^2$ norm.

The top row of Figure 8.4 shows samples of Brownian motions and Brownian bridges in the latent space Z. Each of these Brownian bridges corresponds to a bridge in the data manifold of the MNIST data. Examples of bridges in the high dimensional space X are shown in the bottom row of Figure 8.4.

We now perform PGA on the latent space representation of the subset of the MNIST data consisting of even digits. PGA is a nonlinear coordinate change of the latent space around the Fréchet mean. PGA is applied to the data in Figure 8.5(a) where the resulting data represented in the PGA basis is shown in Figure 8.5(b). The variation along the two principal component

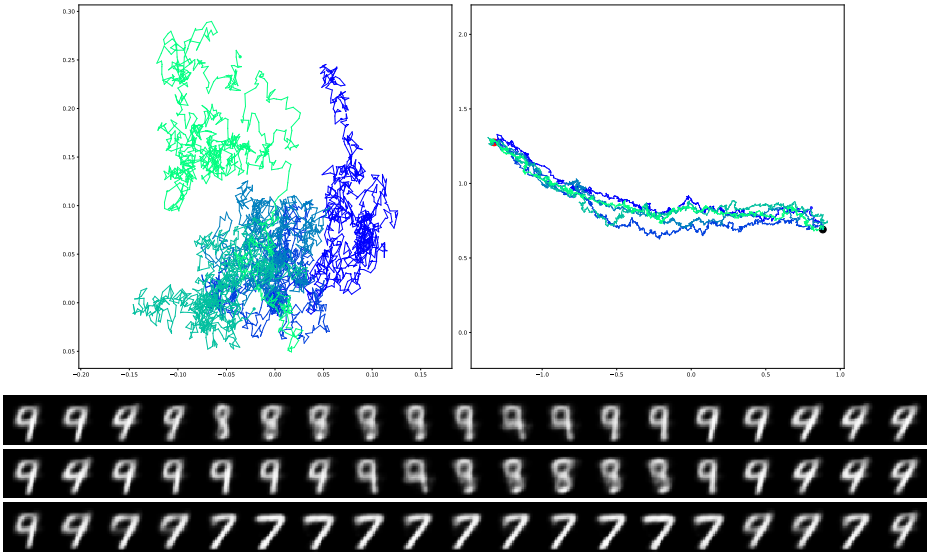


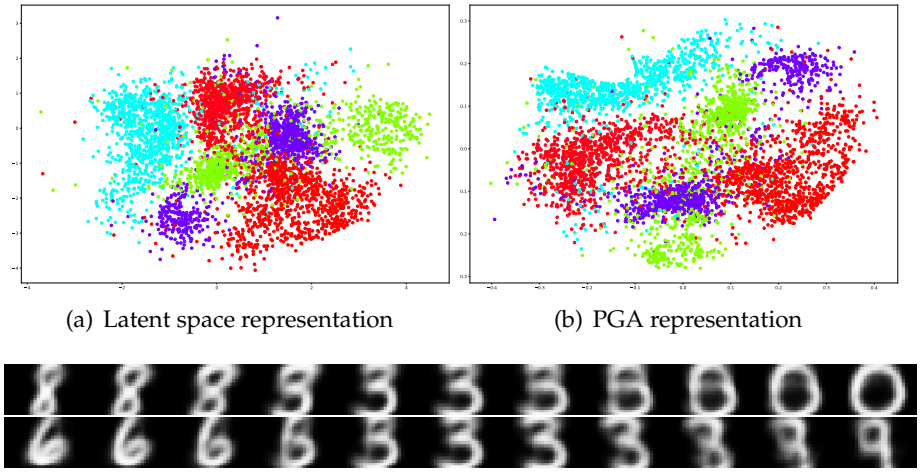
Figure 8.4: (top left) Samples from a Riemannian Brownian motion in latent space. (top right) Samples from a Brownian bridge simulated by (8.9). (bottom) Examples of Brownian bridges of MNIST data between two fixed 9s (left-/rightmost). The variance of the Brownian motion has been increased to visually emphasize the image variation.

directions are visualised in the full dimensional data space in the bottom row of Figure 8.5.

Figure 8.6 (bottom left) shows the maximum likelihood mean image for a subset of 256 even digits estimated by the ML procedure described in Section 8.5. Figure 8.6 (bottom right) shows the corresponding Fréchet mean. The iterations, for both ML and Fréchet mean in latent space, are presented in Figure 8.6 (upper left), with the upper right plot showing the likelihood values for each step of the ML optimisation.

8.6.2 Diatoms

As a final experiment we compare two-sample tests for equal mean for two populations of diatoms in the full data space against a nonlinear two-sample test in the latent space. The diatom dataset consists of 780 observations each a landmark representation of a diatom with 45 landmark points. A VAE has been trained on the diatom data with one hidden layer in both encoder and decoder. The hidden layer consists of 20 neurons and the dimension of the latent space is again set to 2. In Figure 8.7 is shown exsamples of data observations. The goal is to test the hypothesis of equal mean for two populations of diatoms shown in Figure 8.7. The two populations have been generated such that the populations overlap, but are not drawn from the same distribu-



(a) Latent space representation (b) PGA representation

Figure 8.5: (top left) Latent space representation of data. (top right) PGA analysis on the sub-space of even digits. (bottom) Variation along first (1. row) and second (2. row) principal components.

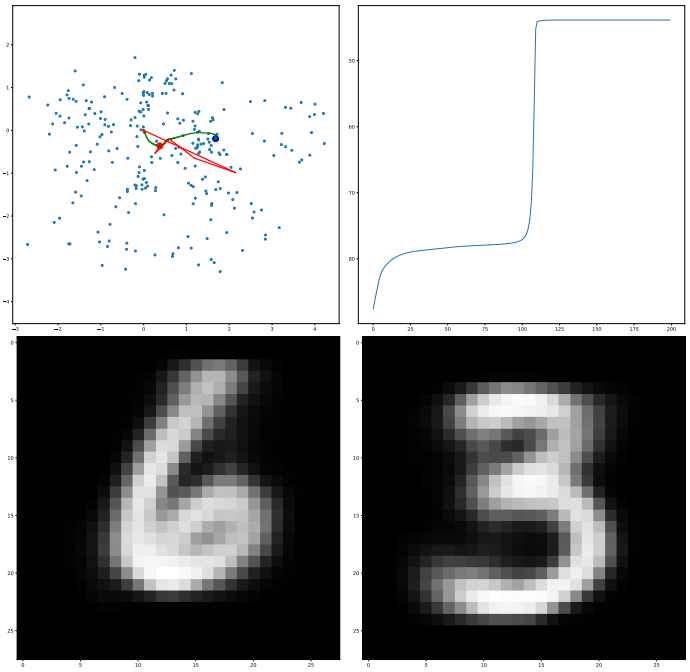


Figure 8.6: From top left row-wise: (1.) Iterations of ML (green) and Fréchet mean (red) for subset of even MNIST digits. (2.) Likelihood evolution during the MLE. Estimated ML mean (3.) and Fréchet mean (4.).

tion. We compare a normal Hotelling test in the full data space with a generalised Hotelling two-sample test in the latent space using the non-linear geometry induced by the decoder mapping. The generalised Hotelling test was presented in Section 8.4.3.

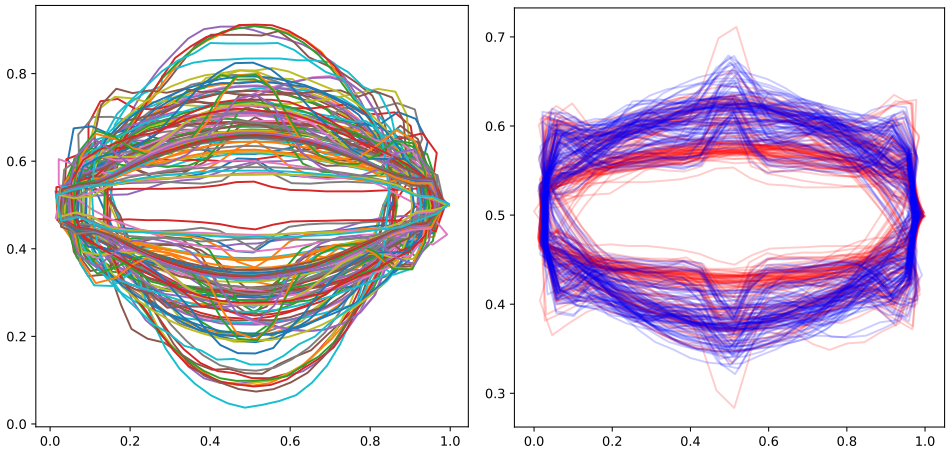


Figure 8.7: (left) Examples of data observations. (right) The two populations tested for equal mean. The populations have been generated such that they overlap but still differ in distribution.

For the diatom example, we investigate whether the test using the dimensionality reduction is better at seperating the two populations compaired to the two-sample test in the full data space. Performing statistical analysis of high-dimensional data often leads to difficulties if the sample size is too small. Similar to the dimensionality reduction based on principal component analysis, we use the low-dimensional latent representation to resolve the curse of dimensionality. On the contrary, when considering the low-dimensional representation of data, important information might have been excluded resulting in no difference between the two populations in latent space. The presented test with the diatom data shows an example where performing two-sample test in the latent space results in a more significant test of the null-hypothesis of equal mean than conducting the analysis in the high-dimensional data space.

Figure 8.8 shows the test statistic for the sample data and a histogram for the permutation based statistics. We notice that the normal Hotelling two-sample test in the full data space (left of Figure 8.8) results in a p-value of approximately 0.042, compaired to the generalised Hotelling test in the latent space which significantly rejects the hypothesis of equal Fréchet mean for the two populations with a p-value of approximately 0.0025.

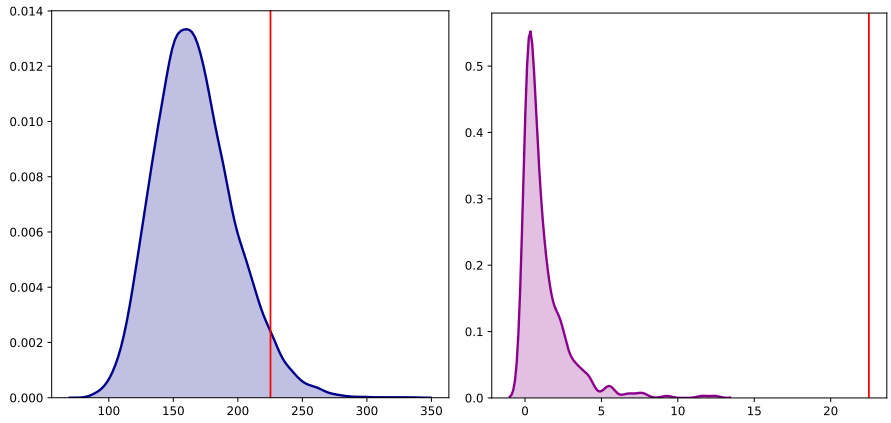


Figure 8.8: (left) Estimated density plot of the test statistic for the two-sample test in high-dimensional data space. The data statistic is shown as the red line. (right) Density plot for the test in latent space with the red line defining the data statistic.

8.7 Conclusion

Deep generative models define an embedding of a low dimensional latent space Z to a high dimensional data space X. The embedding can be used to reduce data dimensionality and move statistical analysis from X to the low-dimensional latent representation in Z. This method can be seen as a non-linear equivalent to the dimensionality reduction commonly performed by PCA. Some versions of generative models project data used for training the low dimensional structure to a specific distribution on the latent space. Performing statistical analysis on the training data is hence unnatural. We proposed to learn the low dimensional structure of the latent space as a predefined step and subsequently perform statistical analysis on newly observed data. The nonlinear structure of data can be represented compactly, and the induced geometry necessitates the use of nonlinear statistical tools. We considered principal geodesic analysis on the latent space, maximum likelihood estimation of the mean using simulations of conditioned diffusion processes and performed a generalised Hotelling two-sample test. The test resulted in a more significant rejection of the null hypothesis for the test in the latent space compared to the two-sample test in the high-dimensional data space. To enable fast computation of the geometric algorithms that involve high-order derivatives of the metric, we fit a second neural network, to predict the metric g and its inverse, which vastly speeds up computations. We visualised examples on 3D synthetic data simulated on $\mathbb{S}^2$ and performed analyses on the MNIST dataset and shape contours of diatoms based on a trained VAE with a 2D latent space.

CHAPTER 9

Conclusion and Future Work

Throughout the thesis, we presented generalised methods for analysing non-linear data structures. The focus has been on incorporating estimation of uncertainty and variation for distributions on manifolds in non-linear statistical methods. Commonly, distributions on Euclidean spaces are described by a closed-form expression of a density function with respect to the Lebesgue measure. An alternative definition is to consider the limit distribution of a stochastic diffusion process. In this thesis, we introduced variation and uncertainty to the generalised statistical methods by applying stochastic theory on non-linear data spaces.

Uncertainty Estimation in Images

In Chapter 4 and 5 ([68, 69]), we considered the task of modelling uncertainty in deformation of images. Two different approaches were developed, one for which deformations were modelled by spatial displacements of a discrete lattice and the other applying the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework, where deformations are defined as diffeomorphisms.

Paper [68] (Chapter 4) proposed a mixed-effects model for separating uncertainty into a warp effect and a spatial intensity effect. Deformations of a template image were modelled as random displacement fields discretised on a regular lattice. The fixed template and the parameters for the random variation effects were simultaneuosly estimated by maximum likelihood optimisation. We presented an efficient likelihood method based on spatial correlated intensity effects modelled as Gaussian Markov random fields. An approximation of the model likelihood was determined via linearisation of the deformation model. The applied linearisation is an effective method for determining a likelihood approximation, but how much information it costs is uncertain. For future work, we wish to explore the effect of the linearisation and apply new methods for determining the likelihood of the non-linear

model. An alternative option for obtaining the likelihood function is to use Brownian bridges as described in [32, 120]. Simulating Brownian motions conditioned on hitting, in this case, an observation y_i can approximate the density function of the data distribution at exactly the observed data points. The obtained density values will, hence, jointly provide an approximation of the likelihood function [120]. Investigating the application of this method to the presented deformation model is left for future work.

The work presented in [69] (Chapter 5) regarded stochastic deformation of images in the LDDMM framework. We proposed to combine the stochastic LDDMM framework introduced in [5] with the fast LDDMM solver [140] to create a computationally efficient method for modelling uncertainty in medical images. The noise was described by parametric fields located on the image domain. Optimisation of the noise field parameters was based on matching the moments of the stochastic image deformations with the observed data moments.

To estimate the moments for the transition distribution of the deformed images, we considered a coarse assumption of any higher-order moment of a stochastic process being the product of the first order moments. The severity of the approximation is unsure, but it will most definitely affect the optimisation results. Future investigation should focus on examining the lost information and find alternative methods for determining the moments of the deformed images. Another interesting goal is to model uncertainty in the Fourier domain. By considering Fourier representation of noise, we remove the spatial location of noise fields and can, therefore, define more global variation in data populations.

Regression on Manifolds

In Chapter 2 and 3 ([66, 67]) the stochastic development regression was presented, generalising the notion of regression in the case of a manifold-valued response variable. The relation between Euclidean covariates and the manifold-valued response was modelled by stochastic development of a semimartingale dependend on the involved covariate variables. The regression model was able to include multiple covariates, model non-geodesic relations, and introduce uncertainty in the relation between variables. Two different optimisation procedures for parameter inference were considered. The first approach approximated the likelihood function by a Laplace approximation. This method was found computationally infeasible as it required calculation of a high-dimensional Hessian matrix. The second approach used the method of moments for the extrinsic additive noise as the moments of this random variable was known. This procedure was computationally faster but resulted in a stochastic objective function. Due to the stochastic objective function, multiple predictions had to be calculated to obtain a stable optimisation procedure.

After working with the stochastic development model in [69], we would like to investigate the opportunity for performing parameter inference by matching data moments to the limiting moments of the stochastic development of the Euclidean semi-martingale. Based on the Fokker-Planck equations, we can obtain an ordinary differential equation for the evolution of the moments for the stochastic development and use the solution to match with the observed data moments. The procedure would result in a more stable and computationally efficient optimisation procedure as predictions are not required. It would, moreover, be natural to incorporate uncertainty directly in the semi-martingale and thereby introduce intrinsically defined noise on the manifold.

As a final note, it would be interesting to use the model for analysing longitudinal data. By modelling the relation as transportation of a semi-martingale, we naturally introduce a time parameter in the model. This time parameter could be used to model the time evolution of patients following a treatment program or to model the evolution of an anatomical object over time as was discussed in Chapter 5.

Applying stochastic theory to describe uncertainty in data populations is just one approach for considering distributions on manifolds. For Euclidean distributions, it is not the most common application as we usually obtain a closed form expression of the probability density function, or at least an approximation of it. However, as probability density functions are hard to define for distributions on manifolds, we may need to take into account alternative methods for describing variation in data populations.

Automatic Differentiation and Non-linear statistics

In Chapter 6 and 7 ([63, 65]) we proposed to implement concepts from differential geometry and non-linear statistics in numerical frameworks primarily developed for Deep Learning tasks. Using the symbolic and automatic calculations available in these numerical frameworks, such as automatic differentiation and symbolic loop functions, we were able to perform concise implementations of the mathematical theory. The task of implementing theoretical concepts came down to a direct translation of mathematical equations making implementations less error-prone.

Many of the deep learning numerical frameworks can run on GPUs and make parallel computing for optimisation of the computation time of the considered task. However, like all software packages, the deep learning numerical frameworks have their limitations. When considering simple tasks, where for example expressions for derivatives are easily obtained, making explicit implementation of the expression can be faster than using automatic differentiation. For complex constructions, which rely on multiple order derivatives or nested symbolic loops, the compilation time for the symbolic functions can

be extensive and memory consuming. However, by choosing the right setup for the numerical software, these restrictions can be considerably limited.

Applying the Deep learning numerical frameworks can be a fast way to verify new ideas and methods without spending unnecessary time on the derivation of equations and gradients for ideas that may not provide the desired outcome.

The software library is based on the Deep Learning framework Theano, a package of Python. As the development of Theano has stopped, a natural next step is to adapt the developed software library in another symbolic deep learning framework, e.g. PyTorch or Tensorflow.

In the paper [64] (Chapter 8), we proposed to use the lower dimensional non-linear latent space representation trained by a Variational Autoencoder (VAE) to perform subsequent analysis on newly observed data. As the latent space is shown to inherit a non-linear geometry from the VAE embedding, non-linear statistical methods have to be used for analysing data in the latent space representation. We performed Principal Geodesic analysis in latent space, maximum likelihood estimation of the mean of the data sample, and a Hotelling two-sample test for the difference in two populations. The two sample test was shown to provide a more significant test for data in the latent space representation, compared to the two sample test in the full dimensional data space. We only performed one example of the Hotelling two-sample test of significance between populations. However, it could be interesting to investigate whether the projection of the high-dimensional data to the latent space representation of a pre-trained VAE, in general, keep sufficient information of the data distributions to result in more significant tests in the latent space compared to the high-dimensional data space.

Bibliography

[1] M. Abadi et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] S. Allassonniere, S. Durrleman, and E. Kuhn. Bayesian mixed effect atlas estimation with a diffeomorphic deformation model. *SIAM Journal on Imaging Sciences*, 8(3):1367–1395, 2015.

[3] A. Arnaudon, A. L. Castro, and D. D. Holm. Noise and dissipation on coadjoint orbits. *Journal of Nonlinear Science*, 28(1):91–145, 2018.

[4] A. Arnaudon, D. D. Holm, A. Pai, and S. Sommer. A Stochastic Large Deformation Model for Computational Anatomy. In *Information Processing in Medical Imaging*, Lecture Notes in Computer Science, pages 571–582. Springer, 2017.

[5] A. Arnaudon, D. D. Holm, and S. Sommer. A Geometric Framework for Stocastic Shape Analysis. *Foundations of Computational Mathematics*, July 2018.

[6] A. Arnaudon, D. D. Holm, and S. Sommer. String Methods for Stochastic Image and Shape Matching. *J. Math. Imaging Vis.*, 60(6):953–967, July 2018.

[7] M. Arnaudon, X. Chen, and A. B. Cruzeiro. Stochastic euler-poincaré reduction. *Journal of Mathematical Physics*, 55(8):081507, 2014.

[8] G. Arvanitidis, L. K. Hansen, and S. Hauberg. Latent Space Oddity: on the Curvature of Deep Generative Models. *ICLR 2018, arXiv:1710.11379*, October 2017.

[9] A. Aswani, P. Bickel, and C. Tomlin. Regression on manifolds: Estimation of the exterior derivative. *The Annals of Statistics*, 39(1):48–81, February 2011. arXiv: 1103.1457.

[10] M. Banerjee, R. Chakraborty, E. Ofori, M. S. Okun, D. E. Vaillancourt, and B. C. Vemuri. A Nonlinear Regression Technique for Manifold Valued Data with Applications to Medical Image Analysis. In *2016 IEEE Conference on CVPR*, pages 4424–4432, June 2016.

[11] M. Banerjee, R. Chakraborty, E. Ofori, D. Vaillancourt, and B. C. Vemuri. Nonlinear regression on Riemannian manifolds and its applications to Neuro-image analysis. *MICCAI*, 9349:719–727, October 2015.

[12] M. F. Beg, M. I. Miller, A. Trouvé, and L. Younes. Computing Large Deformation Metric Mappings via Geodesic Flows of Diffeomorphisms. *International Journal of Computer Vision*, 61(2):139–157, February 2005.

[13] Y. Bengio. Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, 2(1):1–127, November 2009.

[14] C. Bernardi, Y. Maday, J. F. Blowey, J. P. Coleman, and A. W. Craig. *Theory and numerics of differential equations: Durham 2000*. Universitext. Springer-Verlag Berlin Heidelberg, 1 edition, 2001.

[15] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98, January 2017.

[16] A. Bhattacharya and R. Bhattacharya. *Nonparametric statistics on manifolds with applications to shape spaces*. Institute of Mathematical Statistics, 2008.

[17] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, 1996.

[18] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[19] A. Bloch, J. Baillieul, P. Crouch, J. E. Marsden, D. Zenkov, P. S. Krishnaprasad, and R. M. Murray. *Nonholonomic mechanics and control*, volume 24. Springer, 2003.

[20] A. Bruhn, J. Weickert, and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International Journal of Computer Vision*, 61(3):211–231, 2005.

[21] T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

[22] M. do Carmo. *Riemannian Geometry*. Mathematics: Theory & Applications. Birkhäuser Basel, 1992.

[23] B. Carpenter, A. Gelman, M. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. A. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of Statistical Software*, 2016.

[24] N. Chen, A. Klushyn, R. Kurle, X. Jiang, J. Bayer, and P. van der Smagt. Metrics for Deep Generative Models. In *AISTAT 2018*, November 2017. arXiv: 1711.01204.

[25] M.-Y. Cheng and H.-T. Wu. Local Linear Regression on Manifolds and Its Geometric Interpretation. *Journal of the American Statistical Association*, 108(504):1421–1434, December 2013.

[26] G. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 1: Classical Results and Geometric Methods. Applied and Numerical Harmonic Analysis*. Birkhäuser, 2009.

[27] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media, 2011.

[28] E. Cornea, H. Zhu, P. Kim, J. G. Ibrahim, and the Alzheimer's Disease Neuroimaging Initiative. Regression models on Riemannian symmetric spaces. *Journal of the Royal Statistical Society: Series B*, 79:463–482, March 2017.

[29] J. G. Cragg. Using Higher Moments to Estimate the Simple Errors-in-Variables Model. *The RAND Journal of Economics*, 28:S71–S91, 1997.

[30] A. B. Cruzeiro, D. D. Holm, and T. S. Ratiu. Momentum maps and stochastic clebsch action principles. *Communications in Mathematical Physics*, pages 1–40, 2017.

[31] B. C. Davis, P. T. Fletcher, E. Bullitt, and S. Joshi. Population Shape Regression From Random Design Data. In *ICCV*, pages 1–7. IEEE Computer Society, 2007.

[32] B. Delyon and Y. Hu. Simulation of conditioned diffusion and application to parameter estimation. *Stochastic Processes and their Applications*, 116(11):1660–1675, November 2006.

[33] O. Demetz, D. Hafner, and J. Weickert. The complete rank transform: A tool for accurate and morphologically invariant matching of structures. In *Proc. 2013 British Machine Vision Conference, Bristol, UK*, 2013.

[34] D. Elworthy. Geometric aspects of diffusions on manifolds. In Paul-Louis Hennequin, editor, *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, number 1362 in Lecture Notes in Mathematics, pages 277–425. Springer Berlin Heidelberg, 1988.

[35] P. T. Fletcher. Geodesic Regression and the Theory of Least Squares on Riemannian Manifolds. *International Journal of Computer Vision*, 105(2):171–185, November 2012.

[36] P. T. Fletcher, C. Lu, S. M. Pizer, and S. Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *Medical Imaging, IEEE Transactions on*, 2004.

[37] M. Fréchet. *L'intégrale abstraite d'une fonction abstraite d'une variable abstraite et son application a la moyenne d'un élément aléatoire de nature quelconque*. La Revue Scientifique, 1944.

[38] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancie. *Ann. Inst. H. Poincaré*, 10:215–310, 1948.

[39] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008.

[40] T. Fujita and S.-I. Kotani. The Onsager-Machlup function for diffusion processes. *Journal of Mathematics of Kyoto University*, 22(1):115–130, 1982.

[41] R. Furrer, S. R. Sain, et al. spam: A sparse matrix R package with emphasis on MCMC methods for Gaussian Markov random fields. *Journal of Statistical Software*, 36(10):1–25, 2010.

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[43] D. Hafner, O. Demetz, and J. Weickert. Why is the census transform good for robust optic flow computation? In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 210–221. Springer, 2013.

[44] D. Hafner, O. Demetz, J. Weickert, and M. Reißel. Mathematical foundations and generalisations of the census transform for robust optic flow computation. *Journal of Mathematical Imaging and Vision*, 52(1):71–86, 2015.

[45] M. L. Hazelton. Methods of Moments Estimation. In *International Encyclopedia of Statistical Science*, pages 816–817. Springer Berlin Heidelberg, 2011.

[46] C. R. Henderson. Estimation of genetic parameters. *Biometrics*, 6(2):186–187, 1950.

[47] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. Variational methods for multimodal image matching. *International Journal of Computer Vision*, 50(3):329–343, 2002.

[48] J. Hinkle, P. Muralidharan, P. T. Fletcher, and S. Joshi. Polynomial Regression on Riemannian Manifolds. In *ECCV 2012*, pages 1–14. Springer, January 2012.

[49] D. D. Holm. Variational principles for stochastic fluid dynamics. *Proc. Mathematical, Physical, and Engineering Sciences / The Royal Society*, 471(2176), April 2015.

[50] D. D. Holm and T. M. Tyranowski. Variational principles for stochastic soliton dynamics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 472, 2016.

[51] Y. Hong, R. Kwitt, N. Singh, N. Vasconcelos, and M. Niethammer. Parametric Regression on the Grassmannian. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(11):2284–2297, November 2016.

[52] E. P. Hsu. *Stochastic Analysis on Manifolds*. American Mathematical Soc., 2002.

[53] Z. Huang and L. Van Gool. A Riemannian Network for SPD Matrix Learning. *AAAI-17, arXiv:1608.04233*, August 2016. arXiv: 1608.04233.

[54] A. C. Jalba, M. H. F. Wilkinson, and J. B. T. M. Roerdink. Shape representation and recognition through morphological curvature scale spaces. *IEEE Transactions on Image Processing*, 15(2):331–341, February 2006.

[55] G. M. James, T. J. Hastie, and C. A. Sugar. Principal component models for sparse functional data. *Biometrika*, 87(3):587–602, 2000.

[56] A. Jorstad, D. Jacobs, and A. Trouvé. A deformation and lighting insensitive metric for face recognition based on dense correspondences. In *CVPR 2011*, pages 2353–2360. IEEE, June 2011.

[57] S. Joshi, B. Davis, M. Jomier, and G. Gerig. Unbiased diffeomorphic atlas construction for computational anatomy. *NeuroImage*, 23:151–160, 2004.

[58] H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30(5):509–541, September 1977.

[59] R. E. Kass and D. Steffey. Approximate Bayesian Inference in Conditionally Independent Hierarchical Models (Parametric Empirical Bayes Models). *Journal of the American Statistical Association*, 84(407):717–726, September 1989.

[60] W. S. Kendall. Probability, Convexity, and Harmonic Maps with Small Image I: Uniqueness and Fine Existence. *Proceedings of the London Mathematical Society*, s3-61(2):371–406, September 1990.

[61] I. Kolář, J. Slovák, and P. W. Michor. *Natural Operations in Differential Geometry*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[62] N. V. Krylov. On the itô–wentzell formula for distribution-valued processes and related topics. *Probability Theory and Related Fields*, 150(1):295–319, Jun 2011.

[63] L. Kühnel, A. Arnaudon, and S. Sommer. Differential geometry and stochastic dynamics with deep learning numerics. *arXiv 1712.08364*, 2017.

[64] L. Kühnel, T. Fletcher, S. Joshi, and S. Sommer. Latent space non-linear statistics. *arXiv preprint arXiv:1805.07632*, 2018.

[65] L. Kühnel and S. Sommer. *Computational Anatomy in Theano*, pages 164–176. Springer International Publishing, 2017.

[66] L. Kühnel and S. Sommer. Stochastic development regression on non-linear manifolds. In *Information Processing in Medical Imaging*, pages 53–64, Cham, 2017. Springer International Publishing.

[67] L. Kühnel and S. Sommer. Stochastic development regression using method of moments. In *International Conference on Geometric Science of Information*, pages 3–11. Springer, Cham, 2017.

[68] L. Kühnel, S. Sommer, A. Pai, and L. L. Raket. Most likely separation of intensity and warping effects in image registration. *SIAM Journal on Imaging Sciences*, 10(2):578–601, 2017.

[69] L. Kühnel, A. Arnaudon, T. Fletcher, and S. Sommer. Stochastic Image Deformation in Frequency Domain and Parameter Estimation using Moment Evolutions. *arXiv:1812.05537 [cs, math, stat]*, December 2018. arXiv: 1812.05537.

[70] L. Kühnel, A. Arnaudon, and S. Sommer. Differential geometry and stochastic dynamics with deep learning numerics. *arXiv:1712.08364 [cs, stat]*, December 2017. arXiv: 1712.08364.

[71] B. A. Landman, A. Ribbens, B. Lucas, C. Davatzikos, B. Avants, C. Ledig, D. Ma, D. Rueckert, D. Vandermeulen, F. Maes, G. Erus, J. Wang, H. Holmes, H. Wang, J. Doshi, J. Kornegay, J. Manjon, A. Hammers, A. Akhondi-Asl, A. J. Asman, and S. K. Warfield. *MICCAI 2012 Workshop on Multi-Atlas Labeling*. 2012.

[72] K. Lange. Diffusion Processes. In *Applied Probability*, Springer Texts in Statistics, pages 269–295. Springer New York, NY, 2010.

[73] S. L. Lauritzen. *Graphical models*. Oxford University Press, 1996.

[74] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[75] J. M. Lee. *Introduction to smooth manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2003.

[76] J. M. Lee. *Riemannian manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006.

[77] M. Liao. *Lévy processes in Lie groups*. Cambridge University Press, Cambridge; New York, 2004.

[78] L. Lin, B. St Thomas, H. Zhu, and D. B. Dunson. Extrinsic local regression on manifold-valued data. *arXiv:1508.02201 [math, stat]*, August 2015. arXiv: 1508.02201.

[79] M. J. Lindstrom and D. M. Bates. Nonlinear mixed effects models for repeated measures data. *Biometrics*, 46(3):673–687, 1990.

[80] J.-M. Loubes and B. Pelletier. A kernel-based classifier on a Riemannian manifold. *Statistics & Decisions International mathematical journal for stochastic methods and models*, 26(1):35–51, 2009.

[81] J. Ma, M. I. Miller, A. Trouvé, and L. Younes. Bayesian template estimation in computational anatomy. *NeuroImage*, 42(1):252–261, August 2008.

[82] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *Medical Imaging, IEEE Transactions on*, 16(2):187–198, 1997.

[83] B. Markussen. Functional data analysis in an operator-based mixed-model framework. *Bernoulli*, 19:1–17, 2013.

[84] J. E. Marsden and T. S. Ratiu. *Introduction to Mechanics and Symmetry*, volume 17 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 1999.

[85] S. Marsland and T. Shardlow. Langevin Equations for Landmark Image Registration with Uncertainty. *SIAM Journal on Imaging Sciences*, 10(2):782–807, January 2017.

[86] M. Miller, A. Banerjee, G. Christensen, S. Joshi, N. Khaneja, U. Grenander, and L. Matejic. Statistical methods in computational anatomy. *Statistical Methods in Medical Research*, 6(3):267–299, 1997.

[87] M. I. Miller, A. Trouvé, and L. Younes. Geodesic shooting for computational anatomy. *Journal of Mathematical Imaging and Vision*, 24(2):209–228, Mar 2006.

[88] M. A. Mohamed and B. Mertsching. TV-L1 optical flow estimation with image details recovering based on modified census transform. In *Advances in Visual Computing*, pages 482–491. Springer, 2012.

[89] K.-P. Mok. On the differential geometry of frame bundles of Riemannian manifolds. *Journal Für Die Reine Und Angewandte Mathematik*, 1978(302):16–31, 1978.

[90] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer, 1998.

[91] S. Negahdaripour. Revised definition of optical flow: integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–979, September 1998.

[92] E. G. Ng and B. W. Peyton. Block sparse Cholesky algorithms on advanced uniprocessor computers. *SIAM Journal on Scientific Computing*, 14(5):1034–1056, 1993.

[93] M. Niethammer, Y. Huang, and F.-X. Vialard. Geodesic Regression for Image Time-Series. *MICCAI*, 14(0 2):655–662, 2011.

[94] J. Nilsson, F. Sha, and M. I. Jordan. Regression on Manifolds Using Kernel Dimension Reduction. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 697–704, New York, NY, USA, 2007. ACM.

[95] T. Nye. Construction of Distributions on Tree-Space via Diffusion Processes. Mathematisches Forschungsinstitut Oberwolfach, 2014.

[96] A. Pai, S. Sommer, L. Sorensen, S. Darkner, J. Sporring, and M. Nielsen. Kernel bundle diffeomorphic image registration using stationary velocity fields and Wendland basis functions. *IEEE Transactions on Medical Imaging*, PP(99):1–1, 2015.

[97] M. Pal. Consistent moment estimators of regression coefficients in the presence of errors in variables. *Journal of Econometrics*, 14(3):349–364, December 1980.

[98] G. Panin. Mutual information for multi-modal, discontinuity-preserving image registration. In *Advances in Visual Computing*, pages 70–81. Springer, 2012.

[99] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. Highly accurate optical flow computations with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006.

[100] X. Pennec. Probabilities and Statistics on Riemannian Manifolds : A Geometric approach. Technical report, January 2004.

[101] X. Pennec. Intrinsic Statistics on Riemannian Manifolds: Basic Tools for Geometric Measurements. *J. Math. Imaging Vis.*, 25(1):127–154, 2006.

[102] X. Pennec. Barycentric Subspace Analysis on Manifolds. *arXiv:1607.02833 [math, stat]*, July 2016. arXiv: 1607.02833.

[103] J. Pinheiro and D. Bates. *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media, 2006.

[104] J. Pinheiro, D. Bates, S. DebRoy, and D. Sarkar. Linear and nonlinear mixed effects models. *R package version*, 3:57, 2007.

[105] T. Pock, M. Urschler, C. Zach, R. Beichel, and H. Bischof. A duality based algorithm for TV-L^1-optical-flow image registration. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2007*, pages 511–518. Springer, 2007.

[106] L. L. Raket. *pavpop version 0.10*, 2016. https://github.com/larslau/pavpop/.

[107] L. L. Raket and B. Markussen. Approximate inference for spatial functional data on massively parallel processors. *Computational Statistics & Data Analysis*, 72:227 – 240, 2014.

[108] L. L. Raket, S. Sommer, and B. Markussen. A nonlinear mixed-effects model for simultaneous smoothing and registration of functional data. *Pattern Recognition Letters*, 38:1–7, March 2014.

[109] G. K. Robinson. That BLUP is a good thing: The estimation of random effects. *Statistical Science*, 6(1):15–32, 02 1991.

[110] A. Roche, G. Malandain, X. Pennec, and N. Ayache. The correlation ratio as a new similarity measure for multimodal image registration. In *Medical Image Computing and Computer-Assisted Interventation—MICCAI'98*, pages 1115–1124. Springer, 1998.

[111] H. Rue and H. Tjelmeland. Fitting Gaussian Markov random fields to Gaussian fields. *Scandinavian journal of Statistics*, 29(1):31–49, 2002.

[112] F. S. Samaria. *The Database of Faces*. AT&T. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[113] T. Schaffter. Numerical integration of sdes: a short tutorial. Technical report, École Polytechnique Fédérale de Lausanne (EPFL), 2010.

[114] H. Shao, A. Kumar, and P. T. Fletcher. The Riemannian Geometry of Deep Generative Models. *arXiv:1711.08014 [cs, stat]*, November 2017. arXiv: 1711.08014.

[115] X. Shi, M. Styner, J. Lieberman, J. G. Ibrahim, W. Lin, and H. Zhu. Intrinsic regression models for manifold-valued data. *Medical image computing and computer-assisted intervention: MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, 12(Pt 2):192–199, 2009.

[116] N. Singh, J. Hinkle, S. Joshi, and P. T. Fletcher. Hierarchical Geodesic Models in Diffeomorphisms. *Int. Journal of Computer Vision*, 117:70–92, March 2016.

[117] N. Singh, F.-X. Vialard, and M. Niethammer. Splines for diffeomorphisms. *Medical Image Analysis*, 25(1):56–71, October 2015.

[118] S. Sommer. Anisotropic Distributions on Manifolds: Template Estimation and Most Probable Paths. *Information Processing in Medical Imaging: Proceedings of the ... Conference*, 24:193–204, 2015.

[119] S. Sommer. Anisotropically Weighted and Nonholonomically Constrained Evolutions on Manifolds. *Entropy*, 18(12):425, November 2016.

[120] S. Sommer, A. Arnaudon, L. Kühnel, and S. Joshi. Bridge Simulation and Metric Estimation on Landmark Manifolds. In *Graphs in Biomedical Image Analysis, Computational Anatomy and Imaging Genetics*, Lecture Notes in Computer Science, pages 79–91. Springer, Cham, September 2017. DOI: 10.1007/978-3-319-67675-3_8.

[121] S. Sommer, F. Lauze, and M. Nielsen. Optimization over geodesics for exact principal geodesic analysis. *Advances in Computational Mathematics*, 40(2):283–313, April 2014.

[122] S. Sommer and A. M. Svane. Modelling anisotropic covariance using stochastic development and sub-Riemannian frame bundle geometry. *Journal of Geometric Mechanics*, 9(3):391–410, June 2017.

[123] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable Medical Image Registration: A Survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, July 2013.

[124] F. Steinke and M. Hein. Non-parametric Regression Between Manifolds. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1561–1568. Curran Associates, Inc., 2009.

[125] R. S. Strichartz. Sub-Riemannian geometry. *Journal of Differential Geometry*, 24(2):221–263, 1986.

[126] L. Su, A. M. Blamire, R. Watson, J. He, B. Aribisala, and J. T. O'Brien. Cortical and subcortical changes in Alzheimer's disease: A longitudinal and quantitative MRI study. *Current Alzheimer Research*, 13(5):534–544, 2016.

[127] The Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv:1605.02688 [cs]*, May 2016. arXiv: 1605.02688.

[128] A. Trouvé and F.-X. Vialard. Shape splines and stochastic shape evolutions: a second order point of view. *Quarterly of Applied Math.*, 70(2):219–251, 2012.

[129] A. Trouvé and L. Younes. Local geometry of deformable templates. *SIAM Journal on Mathematical Analysis*, 37(1):17–59, January 2005.

[130] A. Trouvé and L. Younes. Metamorphoses through Lie group action. *Foundations of Computational Mathematics*, 5(2):173–198, February 2005.

[131] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P.A. Yushkevich, and J. C. Gee. N4itk: Improved N3 Bias Correction. *IEEE Transactions on Medical Imaging*, 29(6):1310–1320, June 2010.

[132] F.-X. Vialard. Extension to infinite dimensions of a stochastic second-order model associated with shape splines. *Stochastic Processes and their Applications*, 123(6):2110–2157, 2013.

[133] R. Wolfinger. Laplace's approximation for nonlinear mixed models. *Biometrika*, 80(4):791–795, 1993.

[134] X. Xie and K.-M. Lam. Face recognition using elastic local reconstruction based on a single face image. *Pattern Recognition*, 41(1):406–417, January 2008.

[135] L. Younes. *Shapes and Diffeomorphisms*. Springer, 2010.

[136] L. Younes, F. Arrate, and M. I. Miller. Evolutions equations in computational anatomy. *NeuroImage*, 45(1, Supplement 1):S40–S50, March 2009.

[137] Y. Yuan, H. Zhu, W. Lin, and J. S. Marron. Local Polynomial Regression for Symmetric Positive Definite Matrices. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 74(4):697–719, September 2012.

[138] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *Computer Vision—ECCV'94*, pages 151–158. Springer, 1994.

[139] M. Zhang and P. T. Fletcher. Bayesian principal geodesic analysis for estimating intrinsic diffeomorphic image variability. *Medical Image Analysis*, 25(1):37–44, 2015.

[140] M. Zhang and P. T. Fletcher. Finite-dimensional lie algebras for fast diffeomorphic image registration. In *IPMI*, pages 249–260. Springer, 2015.

[141] M. Zhang, N. Singh, and P. T. Fletcher. Bayesian Estimation of Regularization and Atlas Building in Diffeomorphic Image Registration. In *Information Processing for Medical Imaging (IPMI)*, Lecture Notes in Computer Science, pages 37–48. Springer, 2013.