Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik

Master Thesis Computer Science

Bayesian Quadrature on Riemannian Data Manifolds

Christian Fröhlich

13.04.2021

Reviewers

Prof. Dr. Philipp Hennig
(Methods of Machine Learning)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Prof. Dr. Jakob Macke
(Machine Learning in Science)
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen

Fröhlich, Christian:
Bayesian Quadrature on Riemannian Data Manifolds
Master Thesis Computer Science
Eberhard Karls Universität Tübingen
Thesis period: 28.10.2020 - 27.04.2021
Matriculation number: 4089098

Abstract

Riemannian manifolds provide a principled way to model nonlinear geometric structure inherent in data. A Riemannian metric on said manifolds determines geometry-aware shortest paths and provides the means to define statistical models accordingly. However, these operations are typically computationally demanding. To ease this computational burden, we advocate probabilistic numerical methods for Riemannian statistics. In particular, we focus on Bayesian quadrature (BQ) to numerically compute integrals over normal laws on Riemannian manifolds learned from data. In this task, each function evaluation relies on the solution of an expensive initial value problem. We show that by leveraging both prior knowledge and an active exploration scheme, BQ significantly reduces the number of required evaluations and thus outperforms Monte Carlo methods on a wide range of integration problems. As a concrete application, we highlight the merits of adopting Riemannian geometry with our proposed framework on a nonlinear dataset from molecular dynamics.

Zusammenfassung

Riemannsche Mannigfaltigkeiten bieten eine Möglichkeit zur systematischen Modellierung, wenn sich Daten durch eine nicht-lineare Struktur auszeichnen. Eine Riemannsche Metrik auf solchen Mannigfaltigkeiten erlaubt die Berechnung von kürzesten Wegen, welche die Geometrie respektieren, und zudem die Definition geeigneter statistischer Modelle. Allerdings sind diese Operationen typischerweise rechnenaufwändig. Um diese Rechenlast zu erleichtern, plädieren wir für probabilistische numerische Methoden in der Riemannschen Statistik. Wir fokussieren uns spezifisch auf Bayesianische Integration, mit der wir numerisch Integrale über normale Dichtefunktionen auf durch Daten gelernten Mannigfaltigkeiten berechnen. In dieser Problemstellung benötigt jede Funktionsauswertung die Lösung eines aufwändigen Anfangswertproblems. Wir zeigen, dass Bayesianische Integration durch das Ausnutzen von Vorwissen und eine aktive Explorationsstrategie die Anzahl der benötigten Auswertungen verringert und dadurch auf einer breiten Palette an Integrationsproblemen bessere Leistung als Monte Carlo Methoden erbringt. Als konkrete Anwendung demonstrieren wir den Nutzen Riemannscher Methoden (innerhalb unseres Ansatzes) auf einem nicht-linearen Datensatz aus einer Molekulardynamik-Simulation.

Acknowledgements

I am grateful for the continual support of my supervisors Alexandra Gessner (*Bayesian Quadrature*) and Georgios Arvanitidis (*Geometry*). Both of their expertise was critical for this project, a synthesis of their subject areas. They patiently guided me through the process of writing a paper. I am thankful for all the inspiring coffee breaks with Nina Effenberger, Jonathan Schmidt and Marvin Pförtner and I am indebted to Rahel Gerrens for her encouragement. I also thank Nicholas Krämer, Agustinus Kristiadi, and Dmitry Kobak for helpful discussions and Bernhard Schölkopf for feedback. Finally, I thank Philipp Hennig and Jakob Macke for reviewing this.

Contents

List of Figures

List of Tables

List of Abbreviations

ADK	Adenylate Kinase
BQ	Bayesian Quadrature
BVP	Boundary Value Problem
DCV	Directional Cumulative Acquisition
FP	Fixed-Point Geodesic Solver
GMM	Gaussian Mixture Model
GP	Gaussian Process
IVP	Initial Value Problem
LAND	Locally Adaptive Normal Distribution
MNIST	"MNIST" Handwritten Digits Database
ODE	Ordinary Differential Equation
PCA	Principal Component Analysis
PNM	Probabilistic Numerical Methods
RBF	Radial Basis Function (Kernel)
SPD	Symmetric Positive Definite
VAE	Variational Auto-Encoder
VMD	Visual Molecular Dynamics
WSABI	Warped Sequential Active Bayesian Integration
WSABI-L	Linearized WSABI
WSABI-M	Moment-Matched WSABI

Chapter 1

Introduction

The structure of space matters. This is what I have learned from this thesis in a nutshell. Yet, in statistics and machine learning there is the ubiquitous and tacit assumption of a Euclidean geometry, implying that distances can be measured along straight lines. This flat space model is inadequate when data follows a nonlinear trend, which is known as the *manifold hypothesis*. As a result, probability distributions based on a flat geometry may poorly model the data and fail to capture its underlying structure. Generalized distributions that account for curvature of the data space have been put forward to alleviate this issue. In particular, Pennec [2006] proposed an extension of the normal distribution on Riemannian manifolds such as the sphere.

The key strategy to use such distributions on general data manifolds is by replacing Euclidean straight lines with continuous shortest paths, known as geodesics, which respect the nonlinear structure of the data. This is achieved by introducing a Riemannian metric in the data space that specifies how distance and volume are distorted locally.

To this end, Arvanitidis et al. [2016] proposed a maximum likelihood estimation scheme based on a data-induced metric to learn the parameters of a *Locally Adaptive Normal Distribution* (LAND). However, it relies on a computationally expensive optimization task that entails the repeated numerical integration of the unnormalized density on the manifold, for which no closed-form solution exists. Hence we are interested in techniques to improve the efficiency of the numerical integration scheme.

Probabilistic numerics [Hennig et al., 2015, Cockayne et al., 2019] frames computations that are subject to noise and approximation error as active inference. A probabilistic numerical routine treats the computer as a data source, acquiring evaluations according to a policy to decrease its uncertainty about the result. Amongst probabilistic numerical methods, we focus on Bayesian quadrature (BQ) to compute intractable integrals on data manifolds. Bayesian quadrature [O'Hagan, 1991, Briol et al., 2019] treats numerical integration as an inference problem by constructing posterior measures over integrals given observations, i.e., evaluations of the integrand. Besides providing sound uncertainty estimates, the probabilistic approach permits the inclusion of prior knowledge about properties of the function to be integrated, and leverages active learning schemes for node selection as well as transfer learning schemes, for example when multiple similar integrals have to be jointly estimated

[Xi et al., 2018, Gessner et al., 2019]. These properties make BQ especially relevant in settings where the integrand is expensive to evaluate, and make it a suitable tool for integration on Riemannian data manifolds.

Contributions

- The uptake of Riemannian methods in machine learning is principally hindered by prohibitive computational costs. We here address a key aspect of this bottleneck by improving the efficiency of integration on data manifolds.
- We customize Bayesian quadrature to curved spaces by exploiting knowledge about their structure. To this end, we introduce a tailored acquisition function that minimizes the number of expensive computations by selecting informative *directions* (instead of single points) on the manifold. Adopting a probabilistic numerical integration scheme enables efficient exploration of the integrand's support under a suitable prior.
- We demonstrate accuracy and performance of our approach on synthetic and real-world data manifolds, where we observe speedups by factors of up to 20. In these examples we focus on the LAND model, which provides a wide range of numerical integration problems of varying geometry and difficulty. We highlight molecular dynamics as a promising application area for Riemannian machine learning models. The results support the use of probabilistic numerical methods within Riemannian geometry to achieve significant speedup.

As geometric methods with data-induced metrics are not (yet) well known in the machine learning community, we begin with a motivating example, which will guide the subsequent conceptual developments. This also includes a philosophical introduction, which emphasizes that this framework has little to do with a priori structured manifolds, such as spheres or hyperbolic spaces.

Remarks

- Most of this material has been published in [Fröhlich et al., 2021], so this source is not cited further. As this project involved collaborators, some of the sentences here I did not write myself.
- The geometric framework of data-driven metrics in machine learning has been developed by Hauberg et al. [2012], Arvanitidis [2019] and is thus not a contribution of this thesis.
- This thesis is best read on screen instead of in print due to colorful figures.

1.1 A Motivating Example: Adenylate Kinase

In molecular dynamics, biophysical systems are simulated on the atomic level. This approach is useful to understand the conformational changes of a protein, that is,

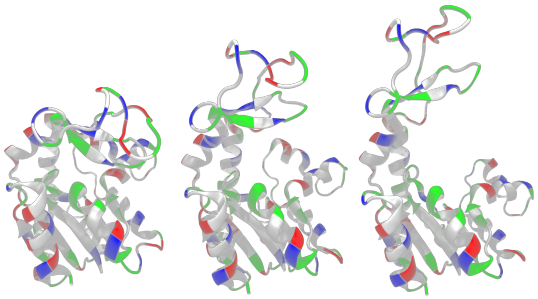


Figure 1.1: ADK in a closed, an intermediate and an open state. The intermediate state is the representative mean resulting from the LAND model.

Figure 1.2: ADK trajectory data (2,038 points), colored according to the *radius of gyration*, a measure which indicates how open the protein is. This visually corroborates the analysis.

the structural changes it undergoes. A conformation is the spatial arrangement of its constituent atoms. Here we consider the example of *adenylate kinase* (ADK). According to Seyler et al. [2015], "ADK's closed/open transition [..] is a standard test case that captures general, essential features of conformational changes in proteins". This well-studied transition involves the movement of the *LID* and *NMP* domains against the rather stable core domain. As a consequence, it can be described by two angles θ_{LID} and θ_{NMP}. In Fig. 1.1, it is visible how the *LID* opens to the top, whereas the *NMP* domain moves towards the bottom right (from this particular perspective). In general, clustering conformations and finding representative substates are scientifically interesting (see e.g., Papaleo et al., 2009, Wolf and Kirschner, 2013, Spellmon et al., 2015, Tribello and Gasparotto, 2019). For the ADK example, assume that we want to find a mean state of the transition motion, which is well situated between the closed and the open state.

We obtained ADK trajectory data from Seyler et al. [2015] [1], specifically, the DIMS variant, a dataset which comprises 200 trajectories and select a subset consisting of trajectories $160 - 200$, which contain in total 2,038 data points. Each observation consists of the Cartesian (x, y, z) coordinates for each of the 3,341 atoms, yielding a 10,023 dimensional vector. As is common in the field, we use principal component analysis (PCA) to extract the *essential dynamics* [Amadei et al., 1993]. A typical

[1] https://www.mdanalysis.org/MDAnalysisData/adk_transitions.html#adk-dims-transitions-ensemble-dataset

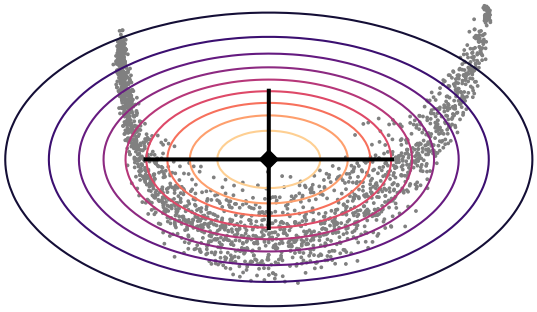


Figure 1.3: A normal distribution fitted to ADK data, showing mean and eigenvectors.

assumption is that the trajectory actually takes place on a low-dimensional subspace and thus it is sufficient and more instructive to analyze is there. Indeed, the first two eigenvectors already explain 65% of the total variance and suffice to capture the transition motion. Figure 1.2 shows the resulting projected data and provides a visual argument for the low-dimensional hypothesis.

In this context, the advantage of using PCA over a complicated non-linear method such as t-SNE is that PCA is well understood, interpretable and comes with a (lossy) inverse transform. See Tribello and Gasparotto [2019] for an overview on dimensionality reduction of protein trajectories. Using Cartesian coordinates has also been criticized [Sittel et al., 2014]. Instead, backbone dihedral angles may be used. I use Cartesian coordinates to keep the analysis simple, however.

To find a representative mean, a natural approach is to fit a 2D normal distribution to the data. As the data evidently follows a nonlinear trend, this model is inappropriate, however. The mean is placed outside the data and the eigenvectors of the covariance do not align with the intrinsic nonlinearity (Figure 1.3).

This raises the question of what it is that is problematic about the familiar normal distribution in this setting. As *the* maximum entropy distribution, characterized by mean $\boldsymbol{\mu} \in \mathbb{R}^D$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$, the normal distribution is a conceptually well-founded generative model. Its density

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{1.1}$$

is based on the Mahalanobis distance $D(\mathbf{x}) = \sqrt{(\mathbf{x}-\boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$, which is essentially the Euclidean distance after applying an affine transformation to remove shift and correlation of the data. In the ADK example, however, the Euclidean metric should be considered inappropriate: a realistic trajectory from the closed to the open state should follow the U-shape of the data, instead of crossing the no man's land inbetween. This is prior knowledge that we have about the data generating process. For example, a protein clearly does not self-intersect, so we cannot expect all regions of the data space to be meaningfully inhabitable.

A principled way to model this phenomenon is by introducing a *Riemannian metric*, which curves the data space in such a way that shortest paths and distance measurements, consequently, respect the data geometry. This allows an extension of

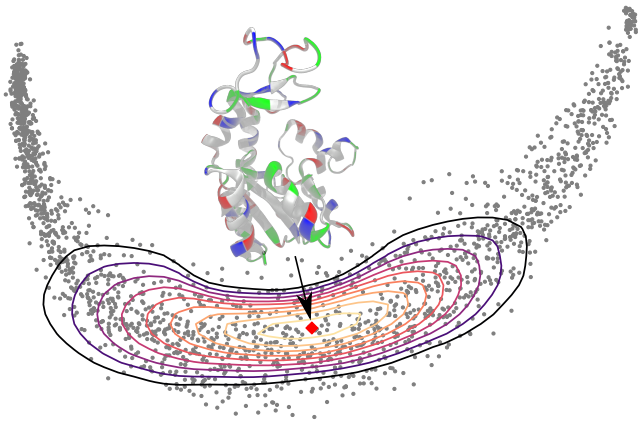


Figure 1.4: A LAND on ADK trajectory data. The conformation corresponding to the LAND mean (◆) is visualized. The contours show the density on the manifold.

the normal distribution to *data manifolds*. Figure 1.4 shows the resulting fit of such a *Locally Adaptive Normal Distribution* [Arvanitidis et al., 2016] on the ADK data.

1.2 The Concept of Space: Kant, Riemann, Einstein and Machine Learning

The typical textbook approach to differential and Riemannian geometry is to state technical definitions of manifolds, charts, etc. without much motivation, relying on pretheoretical intuitions about concrete manifolds like the sphere. Yet this approach leaves many questions regarding the *why* unanswered and is at risk of missing to demonstrate the scope of the implications. We will therefore begin with a philosophico-historical digression on conceptions about space, making our way from Kant and Riemann towards contemporary machine learning.

Whereas in the Newtonian worldview, space is an actual, mind-independent entity, filled with objects that move in it, Leibniz conceptualized space as consisting in relations of those objects. Kant radically departs from these ideas by placing space in the realm of the subject's intuition [Janiak, 2020, Jost, 2013], that is, space is merely a feature through which we perceive objects [Stang, 2021], uninformative of the nature of the objects in themselves. The idea that geometric statements are *synthetic a priori* judgments is central to Kant's doctrine of transcendental idealism. Space is a priori in the sense that, according to Kant, it is possible to imagine space devoid of objects, but impossible to imagine the absence of space itself. This assertion of ontological priority removes geometry from the realm of empirical investigation. Instead, geometry concerns our own representations of objects. Kant further argues for the synthetic character of geometric statements. Consider the exemplary postulate, which we meet again in Chapter 2, that the shortest distance between two points is a straight line. This is a synthetic judgment, because the quantitative concept of length is prima facie unrelated to the qualitative concept of straightness, and thus this statement cannot be obtained by analytic reasoning

alone [Jost, 2013]. Instead, it requires synthetic construction by the subject. So if space is actually a non-empirical representation in the mind of the perceiving subject, what makes Euclidean geometry the "right" one? Kant answers this by pointing out that Euclidean space is the only kind which is intuitively accessible to humans. By necessity, we conceive of space as three-dimensional Euclidean in nature. Whether we follow this line of argumentation or not, one important lesson to take from Kant is to consider the subjective component, for instance the inductive bias that is introduced by the spatial preconceptions of a machine learning engineer.

Kant's successor, J. Herbart, turns against the a priori conception of space and instead proposed a grounding in sensory physiology [Gray, 2019]. Without innate structural priors, space is constructed through experience, i.e., unconscious inference about hidden causes in the world. This theory already hints at modern developments in cognitive science within the Bayesian framework of predictive processing [Wiese and Metzinger, 2017] and was an important influence on Riemann.

In his seminal habilitation lecture, Riemann [1854] goes even further than Herbart and replaces the Kantian concept of space with a completely novel approach. Riemann considers geometry an a posteriori matter: Euclidean space loses its unique characteristic and instead, geometric judgments are hypotheses about the world, subject to empirical validation and falsification. This already hints at the possibility of doing (probabilistic) inference about space itself. Also, Riemann sees no reason for a three-dimensional restriction and invented an abstract geometric metatheory. The present framework of data-aware geometry in machine learning is best seen as abandoning Kantian ideas and as an outcome of the philosophico-mathematical line of work starting with Riemann.

Riemann sets out to question whether the postulates of Euclidean geometry are necessary and thus seeks a more general understanding of geometry. Remarkably, the lecture is almost devoid of formulae. He introduces the notion of a manifold in an abstract fashion as follows:

> Größenbegriffe sind nur da möglich, wo sich ein allgemeiner Begriff vorfindet, der verschiedene Bestimmungsweisen zulässt. Je nachdem unter diesen Bestimmungsweisen von einer zu einer andern ein stetiger Uebergang stattfindet oder nicht, bilden sie eine stetige oder discrete Mannigfaltigkeit; die einzelnen Bestimmungsweisen heissen im erstern Falle Punkte, im letztern Elemente dieser Mannigfaltigkeit. [Riemann, 1854]

> Magnitude-notions are only possible where there is an antecedent general notion which admits of different specialisations. According as there exists among these specialisations a continuous path from one to another or not, they form a continuous or discrete manifoldness; the individual specialisations are called in the first case points, in the second case elements, of the manifoldness. (translated by Clifford, 2013)

Riemann further characterizes a manifold as allowing local coordinate assignment (through *charts*, in modern jargon). For example, the surface of the earth can be

described with multiple charts, together constituting an atlas. Riemann also makes the import distinction between topology, describing only qualitative neighborhood properties, and geometry, which requires imposing additional quantitative metric structure. As opposed to Kant, this makes space an a posteriori concept subject to empirical investigation, which carries significant meaning as opposed to Newton's passive view. From a parsimonious set of assumptions, for instance that the Pythagorean theorem is valid infinitesimally, Riemann then derives a form for the metric tensor [Jost, 2013]. The metric, together with notions of curvature, turn out to be intrinsic quantities. Although their description depends on choosing a specific coordinate system, the objects themselves do not in the sense that they transform according to general, predictable laws under a change of coordinates. This is the subject matter of tensor calculus. A strength of this framework is that it realizes the arbitrariness of coordinate descriptions, yet is still able to carve out substantial invariants. In machine learning, we often have the situation that we cannot interpret coordinates in an intrinsically meaningful way, for example in a latent space of a deep generative model. Therefore we are interested in characterizing invariants, which are unaffected by reparameterization.

Einstein elaborates on the Riemannian postulate that the structure of space is intimately tied to its content and employs the mathematical tools to systematically study applications and implications in physics. Whereas for Newton, objects under gravitational influence move along curved paths in a passive space, in Einstein's general relativity the objects move along straight paths in a curved space [Jost, 2013]. This is an important point which deserves emphasis. The paths *are* straight, it is only the case that they may not appear as such. Perhaps the statement also conjures an inappropriate mental representation, in which the universe is just a Euclidean space that is then embedded in a higher dimensional space with a distortion due to gravity. However, the curvature is *intrinsic*. Unlike a sphere, which is typically seen as embedded in a higher-dimensional Euclidean space, the universe cannot curve into a higher dimension. It curves intrinsically.

Among the most important equations in physics are the Einstein-Field equations

$$\underbrace{R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} + \Lambda g_{\mu\nu}}_{geometry} = \underbrace{\kappa T_{\mu\nu}}_{matter},$$

which relate the curvature of spacetime, expressed by $R_{\mu\nu}$ and R (see Chapter 2), to its content: matter. Although it may look innocent, it is a groundbreaking idea to equate geometry with matter, that is, to view them as two incarnations of a single common core.

Einstein's ideas offer a helpful analogy to what we attempt in machine learning. We are here not investigating embedded manifolds with a priori structure such as the sphere, but instead we consider intrinsic curvature due to the data itself. We propose to jointly infer the geometry of the data space and a statistical model based on the distribution of matter within the space, i..e, the data. Figure 1.5 shows two different paths with the same endpoints on the ADK data. Is one of them straight? Is one of them the shortest path between the points? Answering these subtly different questions requires knowledge of a lot of structure, which is hidden in the Figure. To

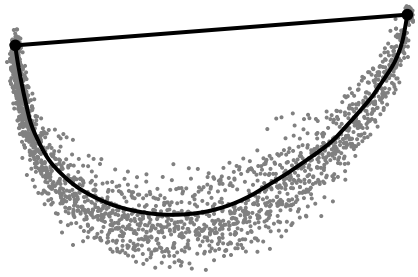


Figure 1.5: A "curved" vs. a "straight" line on ADK. The bottom path is computed as the straightest and shortest path under a data-driven Riemannian metric.

a Kantian data scientist, the answer would be clear, as they would not question the Euclidean assumption, or more precisely, they would state that it holds necessarily. On the other hand, from the Riemannian perspective, this is a well-posed question. The bottom path, which appears curved to a Euclidean observer, might very well be a straight (and the shortest) path, as it follows the trend of the data in a meaningful way. In contrast, the seemingly straight line at the top cannot meaningfully be considered the shortest path, as it crosses no man's land. Domain knowledge tells us that this is dangerous, as the coordinates in this region without data may not correspond to physically meaningful protein conformations and the path thus does not represent a realistic trajectory between the closed and open state.

With this conceptual motivation at hand, we now turn to the technical side. As we are interested in data-induced metrics, topology will only play a minor role and we focus on geometrical aspects. This philosophical introduction also served to emphasize that we consider the space itself, not merely embeddings of surfaces in higher-dimensional spaces. Thus, the wrong example to have in mind is that of a sphere or a torus. Instead, the general relativity metaphor of a dataverse, where mass intrinsically curves the space, is a more suitable intuition to keep in mind in the following.

Chapter 2

Riemannian Geometry

This chapter begins with a brief, intuitive summary of Riemannian geometry by introducing the necessary objects and operations from the bird's eye view. Thereafter we zoom in and make a second, more rigorous pass on the concepts, yet still retaining a structural approach to avoid getting lost in technical details. This exposition primarily draws from the introductory lecture course on differential geometry and relativity by Schuller [2015]. We will use plain face to denote objects (v), whereas coordinate vector/matrix representations of the same objects are set in bold ($\mathbf{v}$). We sometimes switch freely between these viewpoints.

2.1 A Primer on Riemannian Geometry

In our applied setting, we view $\mathbb{R}^D$ as a *smooth manifold* $\mathcal{M}$ with a changed notion of distance and volume measurement as compared to the Euclidean case. This view arises from the assumption that data have a general underlying nonlinear structure in $\mathbb{R}^D$ (see Fig. 2.1), and thus, the following discussion excludes manifolds with structure known a priori, e.g., spheres and tori. In our case, the *tangent space* $\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$ at a point $\boldsymbol{\mu} \in \mathcal{M}$ is again $\mathbb{R}^D$, but centered at $\boldsymbol{\mu}$. This is a vector space that allows to represent points of the manifold as tangent vectors $\mathbf{v} \in \mathbb{R}^D$. Pictorially, a vector $\mathbf{v} \in \mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$ is tangential to some curve passing through $\boldsymbol{\mu}$. Together, these vectors give a linearized view of the manifold with respect to a base point $\boldsymbol{\mu}$.

A *Riemannian metric* is a positive definite matrix $\mathbf{M} : \mathbb{R}^D \to \mathbb{R}_+^{D\times D}$ that varies smoothly across the manifold. Therefore, we can define a local inner product between tangent vectors $\mathbf{v}, \mathbf{w} \in \mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$ as $\langle \mathbf{v}, \mathbf{w} \rangle_{\boldsymbol{\mu}} = \langle \mathbf{v}, \mathbf{M}(\boldsymbol{\mu})\mathbf{w} \rangle$, where $\langle \cdot, \cdot \rangle$ is the Euclidean inner product. This inner product makes the smooth manifold a *Riemannian manifold* [do Carmo, 1992, Lee, 2018].

A Riemannian metric locally scales the infinitesimal distances and volume. Consider a curve $\boldsymbol{\gamma} : [0, 1] \to \mathcal{M}$ with $\boldsymbol{\gamma}(0) = \boldsymbol{\mu}$ and $\boldsymbol{\gamma}(1) = \mathbf{x}$. The length of this curve on the Riemannian manifold $\mathcal{M}$ is computed as

$$L(\boldsymbol{\gamma}) = \int_0^1 \sqrt{\langle \dot{\boldsymbol{\gamma}}(t), \mathbf{M}(\boldsymbol{\gamma}(t))\dot{\boldsymbol{\gamma}}(t) \rangle}\, \mathrm{d}t,$$

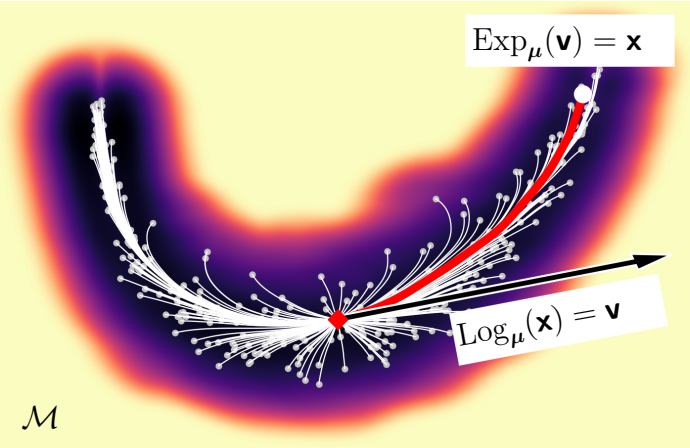


Figure 2.1: A protein trajectory manifold. A subset of the geodesics is shown with respect to a fixed point $\boldsymbol{\mu}$ (◆). The background is colored according to the volume element $\sqrt{|\mathbf{M}|}$ (Sec. 2.10) on a log scale. We omit a colorbar, since the values are not easily interpreted. Darker color indicates regions with small metric, to which shortest paths are attracted. For one geodesic (━), we show the downscaled tangent vector and the Log and Exp operations.

where $\dot{\boldsymbol{\gamma}}(t) = \frac{d}{\mathrm{d}t}\boldsymbol{\gamma}(t) \in \mathcal{T}_{\boldsymbol{\gamma}(t)}\mathcal{M}$ is the velocity of the curve. The $\boldsymbol{\gamma}^*$ that minimizes this functional is the shortest path between the points. To overcome the invariance of L under reparametrization of $\boldsymbol{\gamma}$, shortest paths can equivalently be defined as minimizers of the energy functional. This enables application of the Euler-Lagrange equations, which result in a system of 2^{nd} order nonlinear ordinary differential equations (ODEs). The shortest path is obtained by solving this system as a boundary value problem (BVP) with boundary conditions $\boldsymbol{\gamma}(0) = \boldsymbol{\mu}$ and $\boldsymbol{\gamma}(1) = \mathbf{x}$. Such a length-minimizing curve is known as *geodesic*.

To perform computations on $\mathcal{M}$ we introduce two operators. The first is the *logarithmic map* $\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}) = \mathbf{v}$, which represents a point $\mathbf{x} \in \mathcal{M}$ as a tangent vector $\mathbf{v} \in \mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$. This can be seen as the initial velocity of the geodesic that reaches $\mathbf{x}$ at $t = 1$ with starting point $\boldsymbol{\mu}$. The inverse operator is the *exponential map* $\mathrm{Exp}_{\boldsymbol{\mu}}(t \cdot \mathbf{v}) = \boldsymbol{\gamma}(t)$ that takes an initial velocity $\dot{\boldsymbol{\gamma}}(0) = \mathbf{v} \in \mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$ and generates a unique geodesic with $\boldsymbol{\gamma}(0) = \boldsymbol{\mu}$ and $\boldsymbol{\gamma}(1) = \mathbf{x}$. Note that $\mathrm{Log}_{\boldsymbol{\mu}}(\mathrm{Exp}_{\boldsymbol{\mu}}(\mathbf{v})) = \mathbf{v}$, and also, $\|\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x})\|_2 = \|\mathbf{v}\|_2 = L(\boldsymbol{\gamma})$. Computationally, the logarithmic map amounts to solving a BVP, whereas the exponential map corresponds to an initial value problem (IVP). For general data manifolds, analytic solutions of the geodesic equations do not exist, so we rely on specialized approximate numerical solvers for the BVPs; however, finding shortest paths still remains a computationally expensive problem [Hennig and Hauberg, 2014, Arvanitidis et al., 2019b]. In contrast, the exponential map as an IVP is an easier problem and solutions are significantly more efficient.

We illustrate our applied manifold setting in Fig. 2.1, where we show geodesics starting from a point $\boldsymbol{\mu}$, as well as the Log and Exp operators between $\boldsymbol{\mu}$ and a point $\mathbf{x}$. Note that Figures {2.1,3.1,4.1} are in correspondence, that is, they illustrate different aspects of the same setting. After this primer on Riemannian geometry, we now introduce the concepts in more depth.

2.2 Topological and Smooth Manifolds

At the set-level structure, we cannot talk about continuity of maps. For this, we require a *topology* on the set. Such a topology defines which subsets are considered as *open sets*. The intuition for this may be derived from open intervals of the real numbers $\mathbb{R}$.

Definition 1. *A topology $\mathcal{O}$ on a set $\mathcal{M}$ is a collection of subsets of $\mathcal{M}$ such that $\emptyset, \mathcal{M} \in \mathcal{O}$, with the requirement that $\mathcal{O}$ be closed under arbitrary union of elements and finite intersections. We call $(\mathcal{M}, \mathcal{O})$ a topological space.*

This allows a definition of continuous maps between topological spaces.

Definition 2. *A map $f : \mathcal{M} \longrightarrow \mathcal{N}$ is called continuous if preimages of open sets are again open, i.e., $\mathcal{U} \in \mathcal{N}$ open $\Rightarrow f^{-1}(\mathcal{U})$ open, where "open" is to be understood with respect to the, possibly different, topologies on $\mathcal{M}$ and $\mathcal{N}$.*

With the standard topology on $\mathbb{R}^D$, which is generated by the open balls $B(x, r) = \{y \in \mathbb{R}^D : \|x - y\| < r\}$ $\forall x \in \mathbb{R}^D$ $\forall r \in \mathbb{R}^+$, this coincides with the familiar $\epsilon - \delta$ criterion for continuity.

Definition 3. *A topological space $(\mathcal{M}, \mathcal{O})$ is a D-dimensional topological manifold if it is Hausdorff*[1] *and second-countable*[2] *and can be covered with charts, defined locally on open neighborhoods. Formally, we require*
$\forall p \in \mathcal{M} : \exists \mathcal{U} \in \mathcal{O} : p \in \mathcal{U} : \exists x : \mathcal{U} \subset \mathcal{M} \longrightarrow x(\mathcal{U}) \subset \mathbb{R}^D$*, where x is*

- *continuous with respect to the topology on $\mathcal{M}$ and the standard topology on $\mathbb{R}^D$*
- *invertible*
- *and its inverse x^{-1} is also continuous.*

We call $(\mathcal{U}, x)$ a chart and x a chart map, which assigns a list of coordinates to each point within its domain. An atlas is a collection of charts which covers the whole manifold.

This is already sufficient topology for us, since we work with a particularly simple setting: to keep the analysis tractable, given data in $\mathcal{M} = \mathbb{R}^D$ we inherit the standard topology and work with the global identity chart $x : \mathbb{R}^D \longrightarrow \mathbb{R}^D$, $x = id$, which directly covers the whole manifold. This also avoids a technical issue: if a manifold is covered with multiple overlapping charts, one has to require transition maps between those charts to be smooth, i.e., infinitely differentiable, in order to do calculus on the manifold. Such a manifold is then called a *smooth manifold*. Since we have only one chart, our topological manifold is immediately a smooth manifold. We write $f \in C^\infty$ for a smooth[3] function and call an atlas fulfilling this condition a *smooth atlas* or C^∞-atlas.

[1] A topological space is Hausdorff if for any two points x, y there exist disjoint neighborhoods for x and y, respectively. Thus two points can always be separated.

[2] A topological space is second-countable if it can be generated from a countable basis.

[3] To define smooth maps between two manifolds, one makes use of the chart maps to inherit the familiar smoothness concept by pre- and postcomposition of the respective (inverse) chart maps.

2.3 The Tangent Space

After this short excursion into topology, we now introduce the central construction of differential geometry to enable calculus, the *tangent space*.

Let $\mathcal{M}$ be a smooth manifold, i.e., a set $\mathcal{M}$ together with a topology and a C^∞ atlas. The idea of the tangent space is to construct a vector space at each point on the manifold, where vectors correspond to directional derivatives of curves passing through that point.

First, we define a helper vector space $C^\infty(\mathcal{M}) \coloneqq \{f : \mathcal{M} \longrightarrow \mathbb{R} \mid f \in C^\infty\}$ of smooth functions from a manifold into the reals, equipped with pointwise operations $(f \oplus g)(p) = f(p) +_{\mathbb{R}} g(p)$ and $(\lambda \otimes f)(p) = \lambda \cdot_{\mathbb{R}} f(p), \quad \forall \lambda \in \mathbb{R}$.

Let $\gamma : \mathbb{R} \longrightarrow \mathcal{M}$ a smooth curve on the manifold. W.l.o.g. assume that $\gamma(0) = p$, where p is our point of interest (otherwise reparameterize the curve).

Definition 4. *The velocity of γ at p is the linear map $v : C^\infty(M) \xrightarrow{\sim} \mathbb{R}$, where $f \mapsto v(f) \coloneqq (f \circ \gamma)'(0)$. We denote linearity with $\xrightarrow{\sim}$ and $'$ is the familiar derivative operator w.r.t. the curve parameter. Note that v implicitly depends on both γ and p.*

Importantly, vectors are *functions* in differential geometry. The action of a vector v on a function f yields a directional derivative $vf \coloneqq v(f)$. Why is such a complicated construction necessary? The insight is that $f \circ \gamma : \mathbb{R}^D \longrightarrow \mathbb{R}^D$ is a function for which we have the familiar notion of a derivative available, so we exloit it to lift calculus to the manifold.

Definition 5. *The tangent space $\mathcal{T}_p\mathcal{M}$ at a point p contains the velocities (tangent vectors) of all curves passing through p. It is endowed with a vector space structure using the pointwise operations $(v_1 \oplus v_2)(f) = v_1(f) +_{\mathbb{R}} v_2(f)$ and $(\lambda \otimes v)(f) = \lambda \cdot_{\mathbb{R}} v(f)$.*

It can be checked that these operations are well-defined, so that their result is again a tangent vector. Conceptually, what happens is that by choosing a chart, we can exploit linearity of the familiar derivative and thereby induce a local vector space structure on the manifold.

If we have a chart $x : \mathcal{U} \subset \mathcal{M} \longrightarrow \mathbb{R}^D$, we are interested in chart-dependent components for a vector. To obtain them, we first consider the action of a tangent vector on a function $f \in C^\infty(\mathcal{M})$:

$$\begin{aligned} v(f) &\coloneqq (f \circ \gamma)'(0) \\ &= \left((f \circ x^{-1}) \circ (x \circ \gamma)\right)'(0). \end{aligned}$$

We now write this component-wise, using Einstein-summation convention. Repeated indices are implicitly summed over. Furthermore, a summation index "up" is always accompanied by an index "down". We explore the reason for this later. From the chain rule we obtain:

$$\begin{aligned} &= \left((x \circ \gamma)'\right)^i(0) \cdot (f \circ x^{-1})'_i(x(p)) \\ &= \left((x \circ \gamma)'\right)^i(0) \cdot \left(\partial_i (f \circ x^{-1})\right)(x(p)) \\ &=: \dot{\gamma}^i_x(0) \cdot \left(\frac{\partial f}{\partial x^i}\right)_p, \end{aligned}$$

where we have the components in the first term and defined the suggestive shorthand notation $\left(\frac{\partial}{\partial x^i}\right) = \left((\_ \circ x^{-1})'\right)^i$, which still takes as argument a function f and a point p. Now we can express a tangent vector at p (without acting on a function) in the chart x as

$$v = \dot{\gamma}_x^i(0) \cdot \left(\frac{\partial}{\partial x^i}\right)_p .$$

It can be shown that $\left(\frac{\partial}{\partial x^i}\right)_p, .., \left(\frac{\partial}{\partial x^i}\right)_p$ constitute a basis, the *chart-induced basis*, for $\mathcal{T}_p\mathcal{U}$, where $\mathcal{U} \in \mathcal{M}$ is the domain of the chart x. Furthermore, the vector space dimension of the tangent space coincides with the topological dimension of the manifold.

There is a natural correspondence between vectors and *covectors*, which are linear maps from a vector space into the reals. In our case, the *cotangent space* $\mathcal{T}_p\mathcal{M}^* \coloneqq \{\varphi : \mathcal{T}_p\mathcal{M} \xrightarrow{\sim} \mathbb{R}\}$. The most important example is the differential df of a function $f \in C^\infty(\mathcal{M})$ at a point p, defined as $(df)_p : \mathcal{T}_p\mathcal{M} \xrightarrow{\sim} \mathbb{R}$, $v \mapsto (df)_p(v) \coloneqq vf$, which takes a vector v and then applies it to the function f to obtain the directional derivative of f in the v direction. A basis for this space is given by $(dx^1)_p, ..., (dx^D)_p$, where x_i are the component functions. This basis is indeed a *dual basis*, because $(dx^i)_p \left(\frac{\partial}{\partial x^j}\right)_p = \delta^i_j$.

The geometric treatment makes it clear that gradient and differential are not the same object. While the gradient is a vector (corresponding to the direction of greatest increase), the differential is a covector. In general, converting between them is not as simple as transposing - this is only valid with a Euclidean metric structure.

Vectors and Covectors follow different transformation laws, when the basis is transformed, for instance when changing from cartesian to polar coordinates. The vector space basis transforms (by definition) following a covariant rule, where the "co" indicates that it is the same direction as the basis transformation. On the other hand, vector components transform contravariantly. Thus, vector components carry an upper (contravariant) index, while the basis carries a lower (covariant) index. In the "denominator", the position is switched, so $\frac{\partial}{\partial x^i}$ should be read as having a lower index. For covectors, the opposite is true: the dual basis transforms in a contravariant mode, whereas covector components are covariant. The intuition is that, if we upscale the basis, we have to downscale the dual basis accordingly to keep their relation intact.

2.4 Smooth Vector and Tensor Fields

To define shortest paths on smooth manifolds, we require the notion of a derivative along a curve. As a prerequisite for this, we have to introduce tangent bundles and vector fields, so we can talk about assigning a vector to each point on the manifold.

Definition 6. *The tangent bundle of a smooth manifold is the disjoint union of all tangent spaces on the manifold:* $\mathcal{T}\mathcal{M} \coloneqq \dot{\bigcup} \mathcal{T}_p\mathcal{M}$*, which comes with the projection* $\pi : \mathcal{T}\mathcal{M} \longrightarrow \mathcal{M}$ *that assigns the base point on the manifold, i.e.,* $v \in \mathcal{T}_p\mathcal{M} \mapsto p$.

The tangent bundle becomes a smooth manifold of dimension $2D$ by constructing a smooth atlas that combines two pieces of data, the coordinates of the base point and the coordinates of the tangent vector. Similarly, the cotangent bundle can be defined as the union of all cotangent spaces.

A smooth vector field v then assigns a vector to each point on the manifold, varying in a smooth fashion. Formally, it is a *smooth section* $v : \mathcal{M} \longrightarrow \mathcal{TM}$, requiring $\pi \circ v = id_{\mathcal{M}}$, where the smoothness property is to be understood with respect to the smooth atlases on $\mathcal{M}$ and $\mathcal{TM}$.

Definition 7. *We define the set of smooth vector fields*

$$\Gamma(\mathcal{TM}) = \{v : \mathcal{M} \longrightarrow \mathcal{TM} \mid v \text{ smooth vector field}\}, \tag{2.1}$$

which can be made into a *module* over the ring $(C^\infty(M), +, \cdot)$, where $+$ and $\cdot$ are point-wise operations defined in Section 2.3. Addition and scalar multiplication of the module are also defined pointwise as follows

$$\begin{aligned}(v \oplus \tilde{v})(f) &\coloneqq vf +_{C^\infty(\mathcal{M})} \tilde{v}f \quad \forall v, \tilde{v} \in \Gamma(\mathcal{TM}) \\ (g \odot v)(f) &\coloneqq g \cdot_{C^\infty(\mathcal{M})} vf \quad \forall g \in C^\infty(M), v \in \Gamma(\mathcal{TM})\end{aligned}$$

with a new implicit application $vf : \mathcal{M} \longrightarrow \mathbb{R}$, $p \mapsto v(p)f$, which takes care of assigning a vector to the point p on the manifold. This definition can be extended to accommodate general *tensor fields*.

Definition 8. *An (p, q)-tensor field T is a $C^\infty(\mathcal{M})$-linear map*

$$T : \underbrace{\Gamma(\mathcal{TM}^*) \times .. \times \Gamma(\mathcal{TM}^*)}_{p} \times \underbrace{\Gamma(\mathcal{TM}) \times .. \times \Gamma(\mathcal{TM})}_{q} \xrightarrow{\sim} C^\infty(\mathcal{M})$$

Important examples that we will encounter are the metric tensor and curvature tensors, which are all intrinsic quantities that characterize invariants of the manifold.

2.5 Affine Connections

We have seen that vectors and covectors, or more generally, tensors, follow well-behaved transformation laws. These objects do not change intrinsically under a change of chart, only their components do. Thus, for the derivative of a vector/tensor field in the direction of another vector field, we require an intrinsic construction, that respects the invariant semantics. This is achieved by choosing an *affine connection*, also called a *covariant derivative*. For simplicity, we state the definition using vector fields only.

Definition 9. *A connection $\nabla : \Gamma(\mathcal{TM}) \times \Gamma(\mathcal{TM}) \longrightarrow \Gamma(\mathcal{TM})$ is a bilinear map $(v, u) \mapsto \nabla_v u$, which satisfies $\forall w \in \Gamma(\mathcal{TM}), f \in C^\infty(\mathcal{M})$*

$$\begin{aligned}\nabla_{fv+u} w &= f\nabla_v w + \nabla_u w \\ \nabla_v(fu) &= v(f)u + f\nabla_v u \\ \nabla_v(u + w) &= \nabla_v u + \nabla_v w,\end{aligned}$$

which are sensible requirements for a derivative notion. Importantly, it suffices if v is defined along a curve instead of everywhere.

A connection is not uniquely determined by the smooth manifold structure. Instead, there are D^3 degrees of freedom to choose. Defining chart-dependent coefficients Γ, called the *Christoffel symbols*

$$\nabla_{\left(\frac{\partial}{\partial x^i}\right)} \frac{\partial}{\partial x^j} =: \Gamma_{ij}^k \frac{\partial}{\partial x^k}$$

which are D^3 many functions $\Gamma_{ij}^k : U \longrightarrow \mathbb{R}$, where U is the domain of the chart x. These coefficient functions are *not* tensorial, so their transformation law is more involved. Yet these symbols are often useful to simplify computations.

Using the Christoffel symbols together with the rules for the connection, the covariant derivative can be written as

$$(\nabla_v u)^i = v^m \cdot \left(\frac{\partial}{\partial x^m} u^i\right) + v^m \cdot u^n \cdot \Gamma_{nm}^i$$

The significance of a connection lies in the fact that it defines a *parallel transport*, which enables us to "glue" tangent spaces together.

Definition 10. *We call a vector field u parallely transported along a smooth curve $\gamma : \mathbb{R} \longrightarrow \mathcal{M}$, $\lambda \mapsto \gamma(\lambda)$ with corresponding velocity $v(\lambda)$ if $\nabla_v u = 0$.*

In Euclidean space, parallely transporting a vector is trivial. On a manifold, however, the intrinsic curvature will play a role in the result. A connection furthermore lets us define a notion of a straight line on the manifold.

Definition 11. *An affine geodesic is a curve which is parallely transported along itself (or: autoparallel), obeying $\nabla_v v = 0$.*

In practice, this amounts to solving an initial value problem, given a starting point on the manifold and an initial velocity. While an affine geodesic may be considered as a straight, we did not yet impose any metric structure and thus cannot measure the length of such a curve.

2.6 Riemannian Metrics

To measure angles and the lengths of curves, we need to impose additional metric structure on the smooth manifold, i.e., an inner product. This yields the concept of a shortest path between to points, which then also allows us to make the identification of straightest and shortest paths.

Definition 12. *A Riemannian metric $m : \Gamma(\mathcal{TM}) \times \Gamma(\mathcal{TM}) \xrightarrow{\sim} C^\infty(\mathcal{M})$ is a smooth symmetric positive definite $(0,2)$-tensor field on $\mathcal{M}$ that assigns to vector fields v and u a local inner product $\langle v, u\rangle_m = m(v,u) = m(v,u)$. Using the local matrix representation $\mathbf{M}$ of m, we write $\mathbf{v}^\intercal \mathbf{M} \mathbf{u}$.*

This definition makes the smooth manifold a *Riemannian manifold*. A metric has a dually corresponding structure, which we may call the "inverse" metric tensor $m^{-1} : \Gamma(\mathcal{T}\mathcal{M}^*) \times \Gamma(\mathcal{T}\mathcal{M}^*) \xrightarrow{\sim} C^\infty(\mathcal{M})$. This construction relies on the insight that the metric provides a canonical isomorphism between the tangent and the cotangent space:

Definition 13. *The flat map* $\flat : \Gamma(\mathcal{T}\mathcal{M}) \longrightarrow \Gamma(\mathcal{T}\mathcal{M}^*)$, $\flat(v) \coloneqq u \mapsto m(v,u)$ *can be used to "lower" an index. We require that the inverse exists and define* $\flat^{-1} =: \sharp$. *With this at hand, we can set* $m^{-1}(\omega,\sigma) \coloneqq \omega(\sharp\sigma)$.

As a corollary, we find that the gradient is related to the differential as follows: $\mathrm{grad} f \coloneqq \sharp df$. Consequently, $\langle \mathrm{grad} f, v\rangle_m = vf \quad \forall v \in \mathcal{T}_p\mathcal{M}$. While the differential is a covector field, the gradient is a vector field. In Euclidean space, the metric is the Kronecker delta, represented by the identity matrix, and thus it is sometimes confused whether the differential or gradient is a row or column vector. Note that a row vector is actually a covector, whose transpose - in Euclidean geometry - is a vector.

The Riemannian metric allows for a meaningful definition of the length of a curve $\gamma : \mathbb{R} \longrightarrow \mathcal{M}$, $t \mapsto \gamma(t)$ with tangent vectors v_t as

$$L(\gamma) = \int_0^1 \sqrt{m(v_t, v_t)}\ \mathrm{d}t,$$

where the speed $s = \sqrt{m(v_t, v_t)}$ of the curve is integrated locally. Intuitively, the metric at each point stretches or shrinks the infinitesimal length ruler.

2.7 Geodesics

Definition 14. *A geodesic is a curve* $\gamma : [0,1] \longrightarrow \mathcal{M}$, *which is a stationary curve of the length functional* L. *As such, it fulfills the Euler-Lagrange equations.*

We are interested in the minimizers, i.e., shortest paths, although maximizers are also geodesics. Working directly with the length functional is problematic, because its solution can be arbitrarily reparameterized. Instead, we minimize curve energy to ensure unit speed. For this, we move to the chart, where we can solve the resulting differential equations. Here, we denote the coordinate matrix representation of the metric at point $p \in \mathcal{M}$ as $\mathbf{M}(p)$. The inverse metric m^{-1}, represented by the matrix inverse $\mathbf{M}^{-1}(p)$, is by convention often denoted simply as $M^{ij}(p)$, but with upper indices.

Definition 15. *The energy or action functional of a curve* $\boldsymbol{\gamma}$ *with time derivative* $\dot{\boldsymbol{\gamma}}(t)$ *is defined as*

$$E(\boldsymbol{\gamma}) = \frac{1}{2}\int_0^1 \underbrace{\langle \dot{\boldsymbol{\gamma}}(t), \mathbf{M}(\boldsymbol{\gamma}(t))\dot{\boldsymbol{\gamma}}(t)\rangle}_{=:e}\ \mathrm{d}t.$$

We abbreviate the inner product as $e \coloneqq \langle \dot{\boldsymbol{\gamma}}(t), \mathbf{M}(\boldsymbol{\gamma}(t))\dot{\boldsymbol{\gamma}}(t)\rangle$. Let γ^i denote the i-th coordinate of the curve $\boldsymbol{\gamma}$ at time t and M_{ik} the metric component at row i and

column k, if it is represented as a matrix. Applying the Euler-Lagrange equations to the functional E results in a system of equations involving e

$$\frac{\partial e}{\partial \gamma^k} = \frac{\partial}{\partial t}\frac{\partial e}{\partial \dot{\gamma}^k}, \quad \text{for } k \in 1, \dots, D.$$

which is a system of 2^{nd} order differential equations. We first consider the left-hand side

$$\mathrm{I} \coloneqq \frac{\partial e}{\partial \gamma^k} = \frac{1}{2}\frac{\partial M_{ij}}{\partial \gamma^k}\dot{\gamma}^i\dot{\gamma}^j,$$

which holds due to independence of the coordinates. The right-hand side is

$$\mathrm{II} \coloneqq \frac{\partial}{\partial t}\left[M_{ik}\dot{\gamma}^i\right] = \frac{\partial M_{ik}}{\partial \gamma^j}\dot{\gamma}^i\dot{\gamma}^j + M_{ik}\ddot{\gamma}^i.$$

We expand this using a small index rearrangement trick

$$\mathrm{II} = \frac{1}{2}\frac{\partial M_{ik}}{\partial \gamma^j}\dot{\gamma}^i\dot{\gamma}^j + \frac{1}{2}\frac{\partial M_{jk}}{\partial \gamma^i}\dot{\gamma}^i\dot{\gamma}^j + M_{ik}\ddot{\gamma}^i.$$

This allows us to write $\mathrm{I} = \mathrm{II} \Leftrightarrow \mathrm{II} - \mathrm{I} = 0$ as

$$M_{ik}\ddot{\gamma}^i + \frac{1}{2}\left(\frac{\partial M_{ik}}{\partial \gamma^j} + \frac{\partial M_{jk}}{\partial \gamma^i} - \frac{\partial M_{ij}}{\partial \gamma^k}\right)\dot{\gamma}^j = 0.$$

the next step is to left multiply with the inverse metric tensor, which carries upper indices, and plug in the Christoffel symbols, chosen here as follows

$$\Gamma^k_{ij} = \frac{1}{2}M^{kh}\left(\frac{\partial M_{ih}}{\partial \gamma^j} + \frac{\partial M_{jh}}{\partial \gamma^i} - \frac{\partial M_{ij}}{\partial \gamma^h}\right), \tag{2.2}$$

so we finally obtain the geodesic equations in the canonical form

$$\ddot{\gamma}^k + \Gamma^k_{ij}\dot{\gamma}^j\dot{\gamma}^j = 0, \quad \text{for } k \in 1, \dots, D. \tag{2.3}$$

In this derivation, the metric has forced us to define the Christoffel symbols in a certain way. In fact, when writing out the autoparallelity equation $\nabla_v v = 0$ given by the affine connection, one arrives at the exact same form as Equation (2.3), except that the Christoffel symbols remain "free" in this case. Identifying straight with shortest curves thus amounts to choosing a specific connection in terms of the metric.

Definition 16. *The Levi-Civita connection is the unique connection ∇ on a given Riemannian manifold $\mathcal{M}$, which is*

1. *torsion-free: $\nabla_X Y - \nabla_Y X - [X, Y] = 0$,*
   with the Lie-Bracket $[X, Y]f \coloneqq X(Yf) - Y(Xf)$.
2. *and metric compatible: $\nabla m = 0$.*

Condition 1 can be intuitively stated as requiring that parallel transport along a curve does not involve any twist. Condition 2 demands that parallel transport preserves inner products. From this viewpoint, we now appreciate Kant's understanding that the identification of the straightness with the length-minimizing property is a non-trivial, synthetic judgment.

2.8 The Exponential and Logarithmic Maps

We assume our manifold to be *geodesically complete* [Pennec, 2006], which means that geodesics can be infinitely extended, i.e., their domain is $\mathbb{R}$. As a consequence, we can define two geodesic operations on the whole tangent space.

Definition 17. *The exponential map* Exp_p *on a Riemannian manifold* $\mathcal{M}$ *at a point* p *on the manifold takes an initial velocity* $v = \gamma'(0)$ *and maps to the point* x *reached in unit time along a geodesic. Formally,* $\operatorname{Exp}_p : \mathcal{T}_p\mathcal{M} \longrightarrow \mathcal{M}$, $v \mapsto \operatorname{Exp}_p(v) = \gamma(1) = x$, *where* $\gamma : \mathbb{R} \longrightarrow \mathcal{M}$ *is a geodesic emanating from* p, $\gamma(0) = p$.

The exponential map $\operatorname{Exp}_p(\cdot)$ realizes a *diffeomorphism*[4] in some open neighborhood around p and thus it admits a smooth inverse in said neighborhood. However, we assume this to be true on the whole manifold in practice to keep the analysis tractable.

Definition 18. *The inverse map,* Log_p, *takes a point* x *on the manifold and maps to the initial velocity* v *needed to reach* x *from* p *by following a geodesic. Formally,* $\operatorname{Log}_p : \mathcal{M} \longrightarrow \mathcal{T}_p\mathcal{M}$, $x \mapsto \operatorname{Log}_p(x) = v = \gamma'(0)$, *where* $\gamma : \mathbb{R} \longrightarrow \mathcal{M}$ *is a geodesic emanating from* p *with* $\gamma(0) = p$ *and* $\gamma(1) = x$.

Computing an exponential map requires the solution of an initial value problem, whereas the logarithmic map amounts to a more expensive and less robust boundary value problem.

These maps induce the *exponential chart*, in which a point on the manifold is mapped to the coordinates of $\operatorname{Log}_p(x) = v$ in the tangent space, presupposing a chosen basis for the tangent space. This allows to represent the manifold in terms of the exponential map, thereby giving a linearized view of it. In this chart, geodesics appear as straight lines going through the origin. Additionally, distance with respect to p is preserved, $\|\operatorname{Log}_p(x)\|_2 = \|v\|_2 = L(\gamma)$ [Pennec, 2006]. The Christoffel symbols vanish and the metric at p, but not in any neighborhood of p, is the Kronecker delta.

Recall that tangent vectors are technically functions. To represent the tangent space visually, we can use the exponential chart, which is in direct correspondence. We will later also use this chart for integration. Furthermore, we use the suggestive notation $\mathcal{T}_p\mathcal{M}$ to denote the exponential chart, since it allows us to view points on the manifold in terms of tangent vectors. Moreover, we may also refer to a point in the exponential chart as a tangent vector. It is clear from the context whether a tangent vector $v : C^\infty(\mathcal{M}) \xrightarrow{\sim} \mathbb{R}$ or a vector consisting of exponential coordinates $\mathbf{v} \in \mathbb{R}^D$ is meant. For instance, integrating over the tangent space is not meaningful in the vectors-as-functions view.

2.9 Integration on Riemannian Manifolds

To define probability distributions on manifolds, we are interested in measuring volume. A natural way to establish a *volume form* on a Riemannian manifold is to make use of the metric, instead of inventing a completely new object.

[4] An isomorphism between smooth manifolds.

Recall that the metric tensor provides an inner product structure. In matrix representation of local coordinates, $M_{ij} =: \langle \frac{\partial}{\partial x^i}, \frac{\partial}{\partial x^j} \rangle_m$ gives us a real number for the inner product of the basis vectors, which are the partial derivatives. In our applied setting, where we work with the identity chart and focus on diagonal metrics, we have a simple interpretation: the scalar M_{ii} specifies how to locally scale the familiar partial derivative in the direction of the i-th coordinate axis.

In this sense, we can consider the symmetric positive definite (SPD) matrix $\mathbf{M} = \mathbf{A}^\intercal \mathbf{A}$ as a Gram matrix. The volume of the parallelepiped spanned by the basis vectors is then $|\mathbf{A}| = \sqrt{|\mathbf{M}|}$. This motivates the definition of the *Riemannian volume form*, expressed in local coordinates as $d\mathcal{M} = \sqrt{|\mathbf{M}|}\, dx^1 \wedge ... \wedge dx^D$. The dx^i, called *1-forms*, are the coordinate covector fields. The *wedge product* $\wedge$ combines them to obtain a *differential form* of degree $D = dim(\mathcal{M})$, which is an alternating[5] tensor field. Differential forms of maximal degree are called *volume forms* and can be used for signed integration. A volume form also defines a measure μ on the Borel[6] sets U as $\mu(U) = \int_U d\mathcal{M}$. So we refer to $d\mathcal{M}$ as the Riemannian volume form or measure interchangeably. Our applied setting is further simplified by the fact that we use a global chart, otherwise a "partition of unity" of the manifold is required to account for overlapping charts. Also, orientability would be a concern. Our exponential chart covers the whole manifold by assumption and thus, we can equivalently integrate there [Sommer et al., 2020], building on the insight $\mathbb{R}^D \simeq \mathcal{T}_\mathbf{x}\mathcal{M}$

$$\int_\mathcal{M} f(x) d\mathcal{M} = \int_{\mathbb{R}^D} f(\mathrm{Exp}_\mathbf{x}(\mathbf{v})) \mathbf{M}(\mathrm{Exp}_\mathbf{x}(\mathbf{v}))\, \mathrm{d}\mathbf{v}, \tag{2.4}$$

where $\mathrm{d}\mathbf{v}$ denotes the Lesbegue measure on $\mathbb{R}^D$. After having assembled the geometric background, we now turn to the learning of data-driven metrics.

2.10 Constructing Riemannian Manifolds from Data

We shift our focus on the applied machine learning setting and thus the conversation is about vectors and matrices as coordinate lists now. We implicitly assume the standard topology on $\mathbb{R}^D$ and the identity as the global chart map.

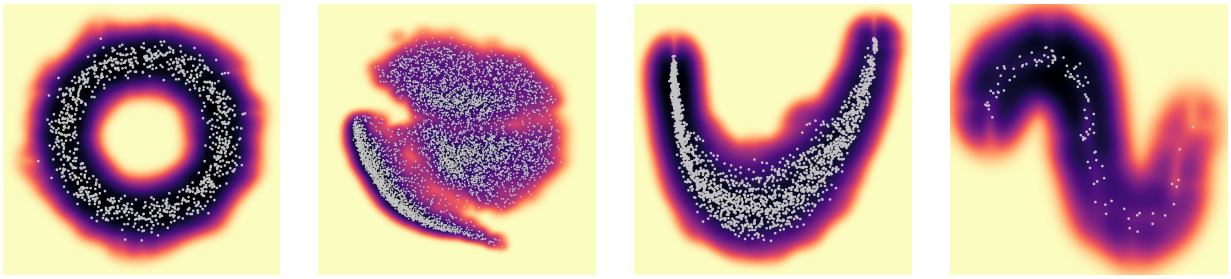
Intuitively, the Riemannian volume element or measure is a distortion of the infinitesimal standard Lebesgue measure $\mathrm{d}\mathbf{x}$. For a meaningful metric this quantity is small near the data and increases as we move away from them. This metric behavior pulls shortest paths near the data.

There are broadly two unsupervised approaches to learn such an adaptive metric from data. Given a dataset $\mathbf{x}_{1:N}$ of N points in $\mathbb{R}^D$, Arvanitidis et al. [2016] proposed a nonparametric metric to model nonlinear data trends as the inverse of a local diagonal covariance matrix with entries

$$M_{dd}(\mathbf{x}) = \left(\sum_{n=1}^{N} w_n(\mathbf{x})(x_{nd} - x_d)^2 + \rho \right)^{-1}, \tag{2.5}$$

[5] A multilinear map means that switching any of its arguments changes the sign.

[6] The Borel σ-algebra is the smallest σ-algebra containing the open sets of a topology. A σ-algebra is closed under countable unions, countable intersections and complements.



(a) CIRCLE (b) MNIST (c) ADK (d) CURLY

Figure 2.2: Exemplary data manifolds. Data scattered in grey, background colored according to the Riemannian measure on a log scale. Dark color corresponds to low values, light color to high values. Colorbars omitted on purpose, as the values are not easily interpreted. Note that the subplots do not share a common color scale. We use the kernel metric for CIRCLE, ADK and CURLY and the surrogate metric for MNIST.

where the weights w_n are obtained from an isotropic Gaussian kernel $w_n(\mathbf{x}) = \exp\left(-\frac{||\mathbf{x}_n - \mathbf{x}||^2}{2\sigma^2}\right)$. The lengthscale σ determines the curvature of the manifold, i.e., how fast the metric changes. The hyperparameter ρ controls the value of the metric components that is reached far from the data, so the measure there is $\sqrt{|\mathbf{M}|} = \rho^{-\frac{D}{2}}$. Typically, ρ is set to a small scalar to encourage geodesics to follow the data trend. However, this metric does not scale to higher dimensions due to the curse of dimensionality [Bishop, 2006, Ch. 1.4].

Another approach to metric learning relies on the *pullback metric*. Let $F : \mathcal{M} \longrightarrow \mathcal{N}$ a smooth immersion, i.e., a smooth map with an everywhere injective derivative. The map F induces a *pushforward* $dF_p : \mathcal{T}_p\mathcal{M} \longrightarrow \mathcal{T}_{F(p)}\mathcal{N}$, defined by [Lebanon, 2012]

$$dF_p(v)(g) = v(g \circ F) \quad \forall g \in C^\infty(\mathcal{N}), \tag{2.6}$$

which maps tangent vectors of $\mathcal{M}$ to tangent vectors of $\mathcal{N}$. Thus, if we have a metric on $\mathcal{N}$, we can use it to get the *pullback metric* on $\mathcal{M}$ as $(F^*m)_p(v,u) = g(dF_p v, dF_p u)$.

This idea can be applied to generative models so as to capture the geometry of high-dimensional data in a low-dimensional latent space [Tosi et al., 2014, Arvanitidis et al., 2018]. Assume a dataset $\mathbf{y}_{1:N} \in \mathbb{R}^{D'}$ with latent representation $\mathbf{x}_{1:N} \in \mathbb{R}^D$ and $D' > D$, such that $\mathbf{y}_n \approx \mathbf{g}(\mathbf{x}_n)$ where $\mathbf{g}$ is a stochastic function with Jacobian $\mathbf{J}_\mathbf{g}(\mathbf{x}) \in \mathbb{R}^{D' \times D}$. Then, the pullback metric $\mathbf{M}(\mathbf{x}) = \mathbb{E}[\mathbf{J}_\mathbf{g}^\intercal(\mathbf{x})\mathbf{J}_\mathbf{g}(\mathbf{x})]$ is naturally induced in the latent space, which enables the computation of lengths that respect the geometry of the data manifold in $\mathbb{R}^{D'}$. Even though this metric reduces the dimensionality of the problem and can be learned directly from the data by learning $\mathbf{g}$, it is computationally expensive to use due to the Jacobian.

To mitigate this shortcoming, we[7] propose a surrogate Riemannian metric. Consider a Variational Auto-Encoder (VAE) [Kingma and Welling, 2014, Rezende et al., 2014] with encoder $q_\phi(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_\phi(\mathbf{y}), \mathrm{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{y})))$, decoder $p_\theta(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y} \mid \boldsymbol{\mu}_\theta(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}^2_\theta(\mathbf{x})))$ and prior $p(\mathbf{x}) = \mathcal{N}(\mathbf{x} \mid \mathbf{0}, \mathbb{I}_D)$, with deep neural networks

[7] This contribution is due to G. Arvanitidis.

Figure 2.3: The "swiss roll', an intrinsically flat $2D$ manifold embedded in $3D$.

as the functions that parametrize the distributions. Then, the aggregated posterior is

$$q_\phi(\mathbf{x}) = \int_{\mathbb{R}^{D'}} q_\phi(\mathbf{x} \mid \mathbf{y}) p(\mathbf{y}) \, \mathrm{d}\mathbf{y} \approx \frac{1}{N} \sum_{n=1}^{N} q_\phi(\mathbf{x} \mid \mathbf{y}_n), \tag{2.7}$$

where the integral is approximated from the training data. This is a Gaussian mixture model that assigns non-zero density only near the latent codes of the data. Thus, motivated by Arvanitidis et al. [2020] we define a diagonal Riemannian metric in the latent space as

$$\mathbf{M}(\mathbf{x}) = (q_\phi(\mathbf{x}) + \rho)^{-\frac{2}{D}} \cdot \mathbb{I}_D. \tag{2.8}$$

This metric fulfills the desideratum of modeling the local behavior of the data in the latent space and it is more efficient than the pullback metric. The variance $\sigma^2_\phi(\cdot)$ of the components is typically small, so the metric adapts well to the data, which, however, may result in high curvature.

In Figure 2.2, four exemplary data manifolds are displayed, with the background colored according to the Riemannian measure. Further information about the construction of these data manifolds is in Chapter 6.

2.11 Curvature

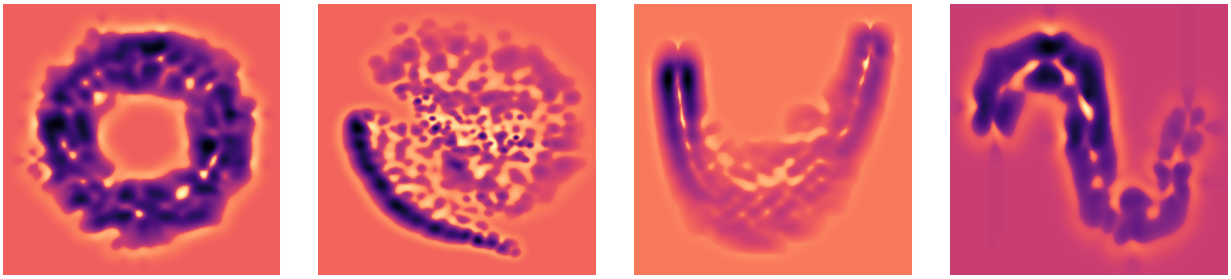
That we consider curvature of the data space itself is an important feature of this approach, which differentiates it from manifold-learning approaches assuming a flat manifold, such as *isomap* [Tenenbaum et al., 2000]. For instance, the famous swiss roll, depicted in Figure 2.3, is actually intrinsically flat, i.e., without curvature.

Gauss had already discovered in his *Theorema Egregium* that curvature of a surface is a intrinsic, independent of how it is embedded in a higher-dimensional space. This insight was elaborated on by Riemann and Ricci, who devised more expressive formulations of curvature. The derivations are quite technical, so we refer the reader to standard literature [Lee, 2018] and keep it brief here. The Riemann tensor measures the extent to which the covariant derivative fails to commute. Formally, it is defined as

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

The Ricci tensor is obtained by taking the trace

$$R(X,Y) = tr(Z \mapsto Riem(Z,X)Y),$$



(a) CIRCLE (b) MNIST (c) ADK (d) CURLY

Figure 2.4: Ricci scalar curvature of exemplary data manifolds, colored on a symmetric logarithmic color scale. Dark color corresponds to negative curvature, light color to positive curvature. Colorbars omitted on purpose, as the values are not easily interpreted. As a reference, the curvature far from the data is 0, as the constant color suggests. Note that the subplots do not share a common color scale. In densely populated areas, there is negative or low positive curvature (dark color).

of which we can again take the trace to obtain the Ricci scalar curvature, which gives us a single number that averages the curvature over all possible directions:

$$R = trR(\cdot, \cdot). \tag{2.9}$$

Intuitively, Ricci curvature measures volume growth along the flow of neighbouring geodesics. Positive curvature means that these geodesics accelerate towards each other, i.e., the second derivative of their separation is negative; contrariwise, negative curvature makes geodesics deviate away from each other.

To compute the Ricci curvature, we use the following coordinate expression:

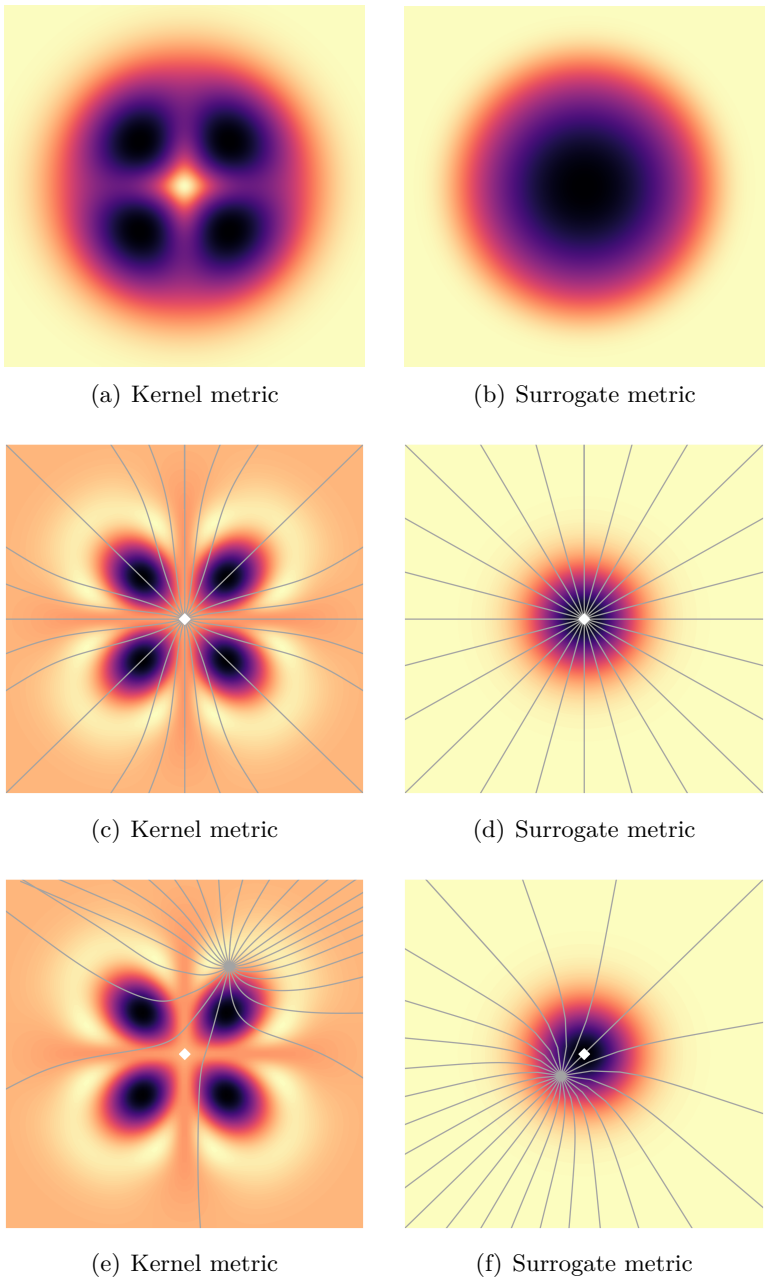
$$\begin{aligned} R_{ij} =& \frac{\partial \Gamma^a_{ij}}{\partial x^a} - \frac{\partial \Gamma^a_{aj}}{\partial x^i} + \Gamma^a_{ab}\Gamma^b_{ij} - \Gamma^a_{ib}\Gamma^b_{aj} \\ =& -\frac{1}{2}\left(\frac{\partial^2 M_{ij}}{\partial x^a x^b} + \frac{\partial^2 M_{ab}}{\partial x^i x^j} - \frac{\partial^2 M_{ib}}{\partial x^j x^a} - \frac{\partial^2 M_{jb}}{\partial x^i x^a}\right) M^{ab} \\ &+ \frac{1}{2}\left(\frac{1}{2}\frac{\partial M_{ac}}{\partial x^i}\frac{\partial M_{bd}}{\partial x^j} + \frac{\partial M_{ic}}{\partial x^a}\frac{\partial M_{jd}}{\partial x^b} - \frac{\partial M_{ic}}{\partial x^a}\frac{\partial M_{jb}}{\partial x^d}\right) M^{ab}M^{cd} \\ &- \frac{1}{4}\left(\frac{\partial M_{jc}}{\partial x^i} + \frac{\partial M_{ic}}{\partial x^j} - \frac{\partial M_{ij}}{\partial x^c}\right)\left(2\frac{\partial M_{bd}}{\partial x^a} - \frac{\partial M_{ab}}{\partial x^d}\right) M^{ab}M^{cd}, \end{aligned}$$

so finally, using the inverse metric to contract indices, we get the Ricci scalar

$$R = M^{ij}R_{ij}.$$

To check the correctness of the implementation, we use the sphere, which has known positive scalar curvature $\frac{2}{R^2}$.

It is instructive to begin studying the curvature of data manifolds by looking at the simplest case first: that of a single datum. Figure 2.5 shows the situation for the kernel metric ($\sigma = 0.1$) as well as the surrogate metric ($\sigma = 0.08$). Note that the surrogate metric is essentially the kernel metric with equal σ for all data,



(a) Kernel metric (b) Surrogate metric

(c) Kernel metric (d) Surrogate metric

(e) Kernel metric (f) Surrogate metric

Figure 2.5: The top row compares the Riemannian measure for the kernel metric ($\sigma = 0.1$) and the surrogate metric ($\sigma = 0.08$), with a single data point at the origin (white diamond). The middle and bottom rows show the corresponding Ricci scalar curvature, with geodesics emanating radially from two different fixed points. The angles of the initial tangent vectors (w.r.t. the Euclidean inner product) are equally spaced between 0 and 2π. It is clearly visible that negative curvature causes geodesics to spread out, whereas positive curvature causes them to close in. Light color indicates positive curvature, dark color negative curvature. The "background" curvature away from the data is always 0. Each plot has an own colorscale, omitted due to difficult interpretability.

where only the weights w_n are kept and the second term in the formula is dropped, so the metric becomes $M_{dd}(\mathbf{x}) = \left(\sum_{n=1}^{N} \frac{1}{Z} w_n(\mathbf{x}) + \rho\right)^{-\frac{2}{D}}$ Arguably the surrogate metric has a more meaningful profile, as it monotonically increases away from the datum at the origin. This is also reflected in the curvature, where the kernel metric displays a rather complex "butterfly" shape that unnecessarily introduces a more complicated curvature landscape. Actually we are more interested in a third order quantity, the change of curvature. As of now, this is just a speculation, but we expect the robustness and speed of the geodesic computations to depend not only on the curvature itself, but on how fast the curvature changes, as this means that the geodesics will be very "wiggly", resulting in an unstable ODE system.

Figure 2.4 shows the Ricci scalar curvature of our data manifolds. As expected, the curvature is negative in areas which is densely populated, although the kernel metric does show unwanted "canyon" effects, where a region of high data concentration is neighboured by two walls of high negative curvature. This is well visible in the left tail of the ADK U-shape. For MNIST we used the variance estimates of a variational auto-encoder (Section 6.3) for σ, which are typically very low, so that the curvature landscape is very fragmented, which results in unstable geodesic computations.

We also experimented with "t-SNE inspired" metric learning. Since the weights w_n are Gaussian, we can consider the perplexity $2^{\mathbb{H}}$ of the resulting probability distribution over neighbours [see Van der Maaten and Hinton, 2008 for details] and then choose σ_i for each datum individually, so as to achieve a desired perplexity. The hope was that this would improve the curvature landscape, yet it turns out difficult to choose a sensible perplexity value. In densely populated regions, this metric tends to overfit ($\downarrow \sigma$), whereas it behaves rather nicely in sparsely populated regions ($\uparrow \sigma$), which might improve the robustness of geodesic computations. Still, this could be an interesting direction for future research.

Chapter 3

Riemannian Statistics

3.1 Probability Distributions on Manifolds

The definitions are from Pennec [2006], a concise introduction to Riemannian statistics, with a focus on manifolds with a priori structure, however.

Definition 19. *Let $(\Omega, \mathcal{B}(\Omega), P)$ be a probability space with Borel algebra $\mathcal{B}(\Omega)$ (Footnote 6) and P a measure on that algebra, with $P(\Omega) = 1$. A random point x on the Riemannian manifold $\mathcal{M}$ is a measurable function $x : \Omega \longrightarrow \mathcal{M}$.*

From this, we can define a probability density function p with respect to the Borel σ-algebra of $\mathcal{M}$, where in our case we inherit the standard topology from $\mathbb{R}^D$.

Definition 20. *The random point $x : \Omega \longrightarrow \mathcal{M}$ has density p if it satisfies $P(x \in U) = \int_U p(\cdot) d\mathcal{M}$ under the constraint $P(\mathcal{M}) = \int_M p(\cdot) d\mathcal{M} = 1$.*

Furthermore, the concept of mean and variance can be generalized to Riemannian manifolds by Fréchet's formulation. Interestingly, the variance is conceptually prior to the mean in this approach.

Definition 21. *The Fréchet variance of a random point x with density $p(\cdot)$ with respect to a fixed point y is defined as*

$$\sigma_x^2(y) = \mathbb{E}\left[dist(y, x)^2\right] = \int_{\mathcal{M}} \| \operatorname{Log}_y(z) \|_2^2 p(z) d\mathcal{M}$$

Definition 22. *A Fréchet mean of the random point x with density $p(x)$ is every of the possibly multiple minimizer of the variance*

$$\mu = \arg\min_{y \in \mathcal{M}} \left[\sigma_x^2(y)\right]$$

While in theory, multiple Fréchet means might exist, in practice one finds local minima (*Karcher means*) by gradient descent on the function (22). For some data manifolds, for instance data sampled from a circle with noise, the optimization landscape will be almost flat, whereas for others it will be convex.

3.2 The Riemannian Normal Distribution

Assume that we place the constraints of knowing mean and covariance (see below) of a random point. We express this in the exponential chart centered at some $\boldsymbol{\mu} \in \mathcal{M}$:

$$\int_{\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}} \mathbf{v} p(\mathrm{Exp}_{\boldsymbol{\mu}}(\mathbf{v})) \mathbf{M}(\mathrm{Exp}_{\boldsymbol{\mu}}(\mathbf{v}))\, \mathrm{d}\mathbf{v} = \mathbf{0}$$
$$\int_{\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}} \mathbf{v}\mathbf{v}^\intercal p(\mathrm{Exp}_{\boldsymbol{\mu}}(\mathbf{v})) \mathbf{M}(\mathrm{Exp}_{\boldsymbol{\mu}}(\mathbf{v}))\, \mathrm{d}\mathbf{v} = \boldsymbol{\Sigma},$$

where $\boldsymbol{\Sigma}$ is an SPD matrix. Under these constraints, Pennec [2006] theoretically derived the *Riemannian normal distribution* as the maximum entropy distribution on a Riemannian manifold $\mathcal{M}$. The density on $\mathcal{M}$ is expressed using the mean $\boldsymbol{\mu}$ and the precision $\boldsymbol{\Gamma}$ as:

$$p(\mathbf{x} \in \mathcal{M} \mid \boldsymbol{\mu}, \boldsymbol{\Gamma}) = \frac{1}{\mathcal{C}(\boldsymbol{\mu}, \boldsymbol{\Gamma})} \exp\left(-\frac{1}{2}\left\langle \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}), \boldsymbol{\Gamma}^{-1} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}) \right\rangle\right).$$

This is reminiscent of the familiar Euclidean density, but with a Mahalanobis distance based on the nonlinear logarithmic maps. Analytic solutions for the normalization constant $\mathcal{C}$ can be given only for certain manifolds that are known a priori, like the sphere or the torus, since this requires analytic solutions for the logarithmic and exponential maps.

Why did we bother to use the precision in the density instead of the perhaps more intuitive covariance matrix, which was used to express the constraint in the chart? On a Riemannian manifold, the covariance is defined to be

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathcal{M}} &= \mathbb{E}\left[\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}) \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x})^\intercal\right] \\ &= \frac{1}{\mathcal{C}(\boldsymbol{\mu}, \boldsymbol{\Gamma})} \int_{\mathcal{M}} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}) \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x})^\intercal \exp\left(-\frac{1}{2}\left\langle \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}), \boldsymbol{\Gamma} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}) \right\rangle\right) \mathrm{d}\mathcal{M}(\mathbf{x}),\end{aligned}$$

so in general, we have $\boldsymbol{\Sigma}_{\mathcal{M}} \neq \boldsymbol{\Gamma}^{-1}$. However, we will simply denote $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}^{-1}$ in what follows, as the "tangent space covariance" is arguably more intuitive, even if it is not actually the covariance for this distribution. We here introduce the notation $\mathrm{d}\mathcal{M}(\mathbf{x})$ to make the dependence of the measure on the point $\mathbf{x}$ visible.

While this distribution has useful applications on a priori structured manifolds, for instance on the manifold of positive symmetric definite matrices in medical imaging [Said et al., 2017], we here consider this distribution on data-driven manifolds.

3.3 The LAND Model

Arvanitidis et al. [2016] extended the Riemannian normal distribution to general data manifolds (see Fig. 1.4) where the Riemannian metric is learned as discussed in Sec. 2.10. In this case, $\mathcal{M} = \mathbb{R}^D$ and $\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M} = \mathbb{R}^D$, so $\boldsymbol{\Sigma} \in \mathbb{R}^{D \times D}_+$. Given a dataset $\mathbf{x}_{1:N}$ assumed to be i.i.d., the log-likelihood of the *Locally Adaptive Normal Distribution* (LAND) mixture can be stated as [Arvanitidis et al., 2016]

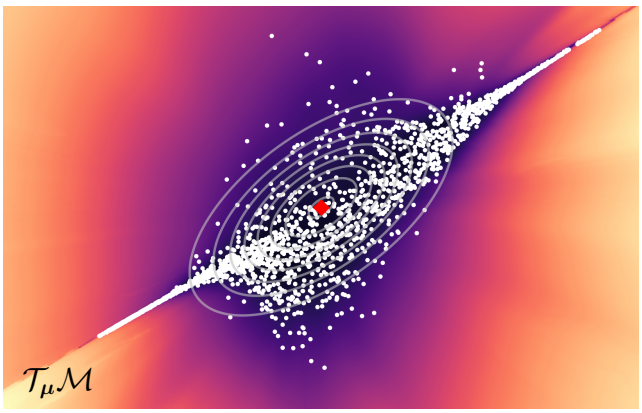


Figure 3.1: The function $g_{\boldsymbol{\mu}}$ on the tangent space, i.e., in the exponential chart, of the ADK manifold. The origin $\mathbf{0}$ (◆) corresponds to the point $\boldsymbol{\mu}$ on the manifold from which the exponential maps are computed. Contours of the integration measure $\mathcal{N}(\mathbf{v}; \mathbf{0}, \boldsymbol{\Sigma})$ are in light gray. Logarithmic maps $\mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}_n)$ of the data are scattered in white. The background is colored according to the volume element on a log scale.

$$\mathcal{L} = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} \left[\frac{1}{2} \langle \mathrm{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n), \boldsymbol{\Sigma}_k^{-1} \mathrm{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n) \rangle + \log\left(\mathcal{C}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) - \log(\pi_k) \right] \quad (3.1)$$

where π_k is the weight of the k^{th} component, with the constraint $\sum_{k=1}^{K} \pi_k = 1$ and $r_{nk} = \frac{\pi_k p(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^{K} \pi_l p(x_n|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$ is the responsibility of the k^{th} component for the n^{th} datum. The maximum likelihood solution can be obtained by non-convex optimization, alternating between gradient descent updates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and cycling through the components k.

The normalization constant is computed using a naïve Monte Carlo scheme as

$$\begin{aligned} \mathcal{C}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \int_{\mathcal{M}} \exp\left(-\frac{1}{2} \langle \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}), \boldsymbol{\Sigma}^{-1} \mathrm{Log}_{\boldsymbol{\mu}}(\mathbf{x}) \rangle \right) \mathrm{d}\mathcal{M}(\mathbf{x}), \\ &= \sqrt{(2\pi)^D |\boldsymbol{\Sigma}|} \int_{\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}} g_{\boldsymbol{\mu}}(\mathbf{v}) \, \mathcal{N}(\mathbf{v}; \mathbf{0}, \boldsymbol{\Sigma}) \, \mathrm{d}\mathbf{v}, \end{aligned} \quad (3.2)$$

where $g_{\boldsymbol{\mu}}(\mathbf{v}) = \sqrt{|\mathbf{M}(\mathrm{Exp}_{\boldsymbol{\mu}}(\mathbf{v}))|}$ gives the tangent space view on the volume element. An example of $g_{\boldsymbol{\mu}}$ is depicted in Fig. 3.1. Instead of having to solve BVPs for the logarithmic maps, we now integrate on the Euclidean tangent space, where we solve significantly faster exponential maps (IVPs).

The normalization constant is needed to estimate maximum likelihood parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For this, we use gradient descent in an alternating fashion, keeping $\boldsymbol{\mu}$ fixed while optimizing $\boldsymbol{\Sigma}$ and vice versa. Note that $\mathcal{C}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ acts as a regularizer, keeping $\boldsymbol{\mu}$ near the data manifold and penalizing an overestimated $\boldsymbol{\Sigma}$. Moreover, the constant enables the definition of a mixture of LANDs.

The MC estimator for this integral requires the evaluation of a large number of exponential maps and is ignorant about known structure of the integrand. We replace MC by BQ to drastically reduce the number of these costly evaluations needed to retain accuracy. Our foremost goal is to speed up numerical integration on data

manifolds since exponential maps are, albeit faster than the BVPs, still relatively slow. The runtime of exponential maps depends on the employed metric (see Sec. 2.10) and on other factors such as curvature or curve length.

For $\boldsymbol{\mu}$, we use the steepest descent direction as in Arvanitidis et al. [2016]

$$d_{\boldsymbol{\mu}_k}\mathcal{L} = \sum_{n=1}^{N} r_{nk} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n) - \frac{\mathcal{Z}_k \cdot R_k}{\mathcal{C}_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \int_{\mathcal{T}_{\boldsymbol{\mu}_k}\mathcal{M}} \mathbf{v} g_{\boldsymbol{\mu}_k}(\mathbf{v}) \mathcal{N}(\mathbf{v}; \mathbf{0}, \boldsymbol{\Sigma}_k) \, \mathrm{d}\mathbf{v}, \tag{3.3}$$

where the vector-valued integral stems from BQ and $R_k = \sum_{n=1}^{N} r_{nk}$ and we have the Euclidean normalization constant $\mathcal{Z}_k = \sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}$. To compute the descent direction for the covariance, we begin with the first term of $\mathcal{L}$

$$\begin{aligned} &\nabla_{\boldsymbol{\Sigma}_k} \left(\sum_{n=1}^{N} r_{nk} \left[\frac{1}{2} \langle \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n), \boldsymbol{\Sigma}_k^{-1} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n) \rangle \right] \right) \\ &\quad = -\frac{1}{2} \sum_{n=1}^{N} r_{nk} \boldsymbol{\Sigma}_k^{-\intercal} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n) \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n)^{\intercal} \boldsymbol{\Sigma}_k^{-\intercal}. \end{aligned}$$

For the gradient of the normalization constant, the second term of $\mathcal{L}$, with respect to the covariance, we get $\nabla_{\boldsymbol{\Sigma}_k} \log(\mathcal{C}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) =$

$$\begin{aligned} &= \frac{1}{\mathcal{C}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \int_{\mathcal{M}} \nabla_{\boldsymbol{\Sigma}_k} \exp\left(\frac{1}{2} \langle \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}), \boldsymbol{\Sigma}^{-1} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}) \rangle \right) \mathrm{d}\mathcal{M}_{\mathbf{x}} \\ &= \frac{1}{2 \cdot \mathcal{C}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \int_{\mathcal{M}} \boldsymbol{\Sigma}_k^{-\intercal} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}) \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x})^{\intercal} \boldsymbol{\Sigma}_k^{-\intercal} \exp\left(-\frac{1}{2} \langle \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}), \boldsymbol{\Sigma}^{-1} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}) \rangle \right) \mathrm{d}\mathcal{M}_{\mathbf{x}} \\ &= \frac{1}{2 \cdot \mathcal{C}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \int_{\mathcal{T}_{\boldsymbol{\mu}_k}\mathcal{M}} \boldsymbol{\Sigma}_k^{-\intercal} \mathbf{v}\mathbf{v}^{\intercal} g_{\boldsymbol{\mu}_k}(\mathbf{v}) \boldsymbol{\Sigma}_k^{-\intercal} \exp\left(-\frac{1}{2} \langle \mathbf{v}, \boldsymbol{\Sigma}^{-1} \mathbf{v} \rangle \right) \mathrm{d}\mathbf{v}. \end{aligned} \tag{3.4}$$

Taking this together, we obtain the gradient

$$\begin{aligned} \nabla_{\boldsymbol{\Sigma}_k}\mathcal{L} = &-\frac{1}{2} \sum_{n=1}^{N} r_{nk} \boldsymbol{\Sigma}_k^{-\intercal} \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n) \operatorname{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n)^{\intercal} \boldsymbol{\Sigma}_k^{-\intercal} \\ &+ \frac{R_k}{2 \cdot \mathcal{C}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)} \int_{\mathcal{T}_{\boldsymbol{\mu}_k}\mathcal{M}} \boldsymbol{\Sigma}_k^{-\intercal} \mathbf{v}\mathbf{v}^{\intercal} g_{\boldsymbol{\mu}_k}(\mathbf{v}) \boldsymbol{\Sigma}_k^{-\intercal} \exp\left(-\frac{1}{2} \langle \mathbf{v}, \boldsymbol{\Sigma}^{-1} \mathbf{v} \rangle \right) \mathrm{d}\mathbf{v}, \end{aligned} \tag{3.5}$$

where the matrix-valued integral again stems from BQ. We optimize with gradient descent and a deterministic manifold linesearch (Section 5.4) as a subroutine, which adaptively chooses its step lengths.

In sum, the optimization process is as follows: we cycle through the components K. After taking a single steepest-direction step for $\boldsymbol{\mu}_k$, we perform two gradient descent steps for $\boldsymbol{\Sigma}_k$, each of which may use up to 4 steps in the linesearch subroutine to satisfy a sufficient decrease criterion. Pseudocode for the LAND optimization is provided in Alg. 1.

Algorithm 1 LAND mixture main loop

Input: data $\mathbf{x}_{1:N}$, manifold $\mathcal{M}$ with Exp and Log operators, max. number of iterations t_{max},
initial stepsize $\alpha_{\boldsymbol{\mu}}^1 \in \mathbb{R}$, gradient tolerance $\epsilon_{\nabla_{\boldsymbol{\mu}}}$, likelihood tolerance $\epsilon_{\mathcal{L}}$
Output: estimates $(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \mathcal{C}_k, \pi_k)_{1:K}$
Initialize LAND parameters$\left(\boldsymbol{\mu}_k^1, \boldsymbol{\Sigma}_k^1, \mathcal{C}_k^1, \pi_k^1\right)_{1:K}$, $t \leftarrow 1$.
repeat
  Expectation step: $r_{nk} = \frac{\pi_k p(x_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{l=1}^K \pi_l p(x_n|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)}$
  Maximization step:
  for $k = 1$ **to** K **do**
    Compute $\mathcal{C}_k^t(\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t)$
    Compute $d_{\boldsymbol{\mu}_k}\mathcal{L}(\boldsymbol{\mu}_k^t, \boldsymbol{\Sigma}_k^t)$ using Eq. (3.3)
    if $||d_{\boldsymbol{\mu}_k}\mathcal{L}|| < \epsilon_{\nabla_{\boldsymbol{\mu}}}$ **then**
      Continue
    end if
    $\boldsymbol{\mu}_k^{t+1} \leftarrow \mathrm{Exp}_{\boldsymbol{\mu}_k^t}(\alpha_{\boldsymbol{\mu}}^t d_{\boldsymbol{\mu}_k}\mathcal{L})$
    Compute $\mathrm{Log}_{\boldsymbol{\mu}_k^{t+1}}(\mathbf{x}_{1:N})$
    Compute $\mathcal{C}_k^{t+1}(\boldsymbol{\mu}_k^{t+1}, \boldsymbol{\Sigma}_k^t)$
    $\boldsymbol{\Sigma}_k^{t+1} \leftarrow \mathrm{update}_{\boldsymbol{\Sigma}_k^t}$ using Alg. 2
    $\pi_k^t = \frac{1}{N}\sum_{n=1}^N r_{nk}$
  end for
  if $\mathcal{L}^{t+1} < \mathcal{L}^t$ **then**
    $\alpha_{\boldsymbol{\mu}}^{t+1} \leftarrow 1.1 \cdot \alpha_{\boldsymbol{\mu}}^t$ {optimism}
  else
    $\alpha_{\boldsymbol{\mu}}^{t+1} \leftarrow 0.75 \cdot \alpha_{\boldsymbol{\mu}}^t$ {pessimism}
  end if
  $t \leftarrow t + 1$
until $||\mathcal{L}^{t+1} - \mathcal{L}^t|| \leq \epsilon_{\mathcal{L}}$ **or** $t = t_{max}$

Chapter 4

Bayesian Quadrature

Bayesian quadrature (BQ) is a probabilistic approach to integration that performs Bayesian inference on the value of the integral given function evaluations and prior assumptions on the integrand (e.g., smoothness). The probabilistic model enables a decision-theoretic approach to the selection of informative evaluation locations. BQ is thus inherently sample-efficient, making it an excellent choice in settings where function evaluations are costly, as is the case in Eq. (3.2) where evaluations of the integrand rely on exponential maps. The key strategy for the application of BQ in a manifold context is to move the integration to the Euclidean tangent space. We review vanilla BQ and then apply adaptive BQ variants to the integration problem in Eq. (3.2).

4.1 Vanilla BQ

Bayesian quadrature seeks to compute otherwise intractable integrals of the form

$$\mathcal{C} = \int_{\mathbb{R}^D} f(\mathbf{v})\pi(\mathbf{v})\ \mathrm{d}\mathbf{v}, \tag{4.1}$$

where $f(\mathbf{v}) : \mathbb{R}^D \to \mathbb{R}$ is a function that is typically expensive to evaluate and $\pi(\mathbf{v})$ is a probability measure on $\mathbb{R}^D$. A Gaussian process (GP) prior is assumed on the integrand to obtain a posterior distribution on the integral by conditioning the GP on function evaluations. Such a GP is a distribution over functions with the characterizing property that the joint distribution of a finite number of function values is multivariate normal [Rasmussen and Williams, 2006]. It is parameterized by its mean function $m(\mathbf{v})$ and covariance function (or kernel) $k(\mathbf{v}, \mathbf{v}')$, and we write $f \sim \mathcal{GP}(m, k)$. The choice of kernel allows incorporating prior knowledge about the function, for example smoothness and lengthscale, and thereby specifies the inductive bias of the GP. After observing data $\mathcal{D}$ at input locations $\mathbf{V} = \mathbf{v}_{1:M}$ and evaluations $\mathbf{f} = f(\mathbf{v})_{1:M}$, the posterior is

$$\begin{aligned} m_{\mathcal{D}}(\mathbf{v}) &= m(\mathbf{v}) + k(\mathbf{v}, \mathbf{V})k(\mathbf{V}, \mathbf{V})^{-1}\left(\mathbf{f} - m(\mathbf{V})\right), \\ k_{\mathcal{D}}(\mathbf{v}, \mathbf{v}') &= k(\mathbf{v}, \mathbf{v}') - k(\mathbf{v}, \mathbf{V})k(\mathbf{V}, \mathbf{V})^{-1}k(\mathbf{V}, \mathbf{v}'). \end{aligned} \tag{4.2}$$

Due to linearity of the integral operator, the random variable $\mathcal{C}$ representing our belief about the integral follows a univariate normal posterior after conditioning on observations, with posterior mean $\mathbb{E}[\mathcal{C} \mid \mathcal{D}] = \int m_{\mathcal{D}}(\mathbf{v})\pi(\mathbf{v})\,\mathrm{d}\mathbf{v}$ and variance $\mathbb{V}[\mathcal{C} \mid \mathcal{D}] = \int k_{\mathcal{D}}(\mathbf{v}, \mathbf{v}')\pi(\mathbf{v})\pi(\mathbf{v}')\,\mathrm{d}\mathbf{v}\,\mathrm{d}\mathbf{v}'$. For certain choices of m, k and π, these expressions have a closed-form solution [Briol et al., 2019].

4.2 Warped BQ

The integration task we consider is Eq. (3.2), where the integration measure on the tangent space $\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M}$ is Gaussian of the form $\pi(\mathbf{v}) = \mathcal{N}(\mathbf{v}; \mathbf{0}, \boldsymbol{\Sigma})$. To encode the known positivity of the integrand $g \coloneqq g_{\boldsymbol{\mu}}(\mathbf{v}) > 0$, we model its square-root by a Gaussian process. Let $f = \sqrt{2(g - \delta)}$, where $\delta > 0$ is a small scalar and $f \sim \mathcal{GP}(m, k)$. Consequently, $g = \delta + \frac{1}{2}f^2$ is guaranteed to be positive. Assume noise-free observations of f as $\mathbf{f} = \sqrt{2(g(\mathbf{V}) - \delta)}$. Inference is done in f-space and induces a non-Gaussian posterior on g.

4.3 WSABI on Manifolds

To overcome non-Gaussianity, Gunter et al. [2014] proposed an algorithm dubbed *warped sequential active Bayesian integration* (WSABI). WSABI approximates g by a GP either via a local linearization on the marginal of g at every location $\mathbf{v}$ (WSABI-L) or via moment-matching (WSABI-M). The approximate mean and covariance function of g can be written in terms of the moments of f as

$$\begin{aligned} \tilde{m}_{\mathcal{D}}(\mathbf{v}) &= \delta + \frac{1}{2} m_{\mathcal{D}}(\mathbf{v})^2 + \frac{\eta}{2} k_{\mathcal{D}}(\mathbf{v}, \mathbf{v}), \\ \tilde{k}_{\mathcal{D}}(\mathbf{v}, \mathbf{v}') &= \frac{\eta}{2} k_{\mathcal{D}}(\mathbf{v}, \mathbf{v}') + m_{\mathcal{D}}(\mathbf{v}) k_{\mathcal{D}}(\mathbf{v}, \mathbf{v}') m_{\mathcal{D}}(\mathbf{v}'), \end{aligned} \tag{4.3}$$

where $\eta = 0$ for WSABI-L and $\eta = 1$ for WSABI-M. For suitable kernels, these expressions permit closed-form integrals for BQ. Chai and Garnett [2019] discuss these and other possible transformations. Kanagawa and Hennig [2019] showed that the algorithm is consistent for $\delta > 0$.

In the present setting, WSABI offers three main advantages over vanilla BQ and MC: First, it encodes the prior knowledge that the integrand is positive everywhere. Second, for metrics learned from data, the volume element typically grows fast and takes on large values away from the data. This makes modeling g directly by a GP impractical, especially when the kernel encourages smoothness. The square-root transform alleviates this problem by reducing the dynamic range of f compared to that of g. Finally, WSABI yields an active learning scheme that adapts to the integrand in that the utility function explicitly depends on previous function values. Gunter et al. [2014] proposed *uncertainty sampling* as to which the next location is evaluated at the location of largest uncertainty under the integration measure. The WSABI objective is the posterior variance of the unwarped GP scaled with the squared integration measure

$$u(\mathbf{v}) = \tilde{k}_{\mathcal{D}}(\mathbf{v}, \mathbf{v})\pi(\mathbf{v})^2, \tag{4.4}$$

which we optimize to sequentially select the next tangent vector for evaluation of the exponential map and thus g.

4.4 DCV

In our manifold setting, the acquisition function is defined on the tangent space. Exponential maps yield intermediate steps on straight lines in the tangent space, as moving along a geodesic does not change the direction of the initial velocity, only the distance traveled. As a result, after solving $\mathrm{Exp}_{\boldsymbol{\mu}}(\alpha \cdot \mathbf{r})$ for a unit vector $\mathbf{r}$, we get evaluations of the integrand at other locations $\beta\mathbf{r},\ 0 < \beta < \alpha$, essentially for free. Because the IVP is solved already, only g has to be evaluated at the respective location, which is cheap once the exponential map is computed.

This observation motivates rethinking the scheme for sequential design to select good initial directions instead of fixed velocities for the exponential map. We propose to select these initial directions such that the cumulative variance along the direction on the tangent space is maximized, and multiple points along this line with large marginal variance are then selected to reduce the overall variance. The modified acquisition function, i.e., the cumulative variance along a straight line, can be written as

$$\bar{u}(\mathbf{r}) = \int_0^\infty u(\beta\mathbf{r})\ \ \mathrm{d}\beta = \int_0^\infty \tilde{k}_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r})\pi(\beta\mathbf{r})^2\ \mathrm{d}\beta, \tag{4.5}$$

The new acquisition policy arises from optimizing $\bar{u}$ for unit tangent vectors $\mathbf{r}$. We call the acquisition function from Eq. (4.5) *directional cumulative variance* (DCV). While it does have a closed-form solution, that solution costs $\mathcal{O}(M^4)$ to evaluate in the number M of evaluations chosen for BQ. We resort to numerical integration to compute the objective and its gradient. This is feasible because these are multiple univariate integrals that can efficiently be estimated from the same evaluations. Once an exponential map is computed, we use the standard WSABI objective to sample multiple informative points along the straight line $\alpha \cdot \mathbf{r}$. For simplicity, we use DCV only in conjunction with WSABI-L.

The derivative of the DCV acquisition function is

$$\frac{\partial}{\partial \mathbf{r}}\bar{u}(\mathbf{r}) = \int_0^\infty \beta\pi(\beta\mathbf{r}) \left[2\tilde{k}_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r})\frac{\partial}{\partial\beta\mathbf{r}}\pi(\beta\mathbf{r}) + \pi(\beta\mathbf{r})\frac{\partial}{\partial\beta\mathbf{r}}\tilde{k}_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r})\right] \mathrm{d}\beta. \tag{4.6}$$

Since the integration measure is Gaussian, i.e., $\pi(\beta\mathbf{r}) = \mathcal{N}(\beta\mathbf{r}; 0, \boldsymbol{\Sigma})$, its derivative is

$$\frac{\partial}{\partial\beta\mathbf{r}}\pi(\beta\mathbf{r}) = -\pi(\beta\mathbf{r})\boldsymbol{\Sigma}^{-1}\beta\mathbf{r}. \tag{4.7}$$

The derivative of the variance of the warped GP is

$$\begin{aligned}\frac{\partial}{\partial\beta\mathbf{r}}\tilde{k}_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r}) &= \frac{\partial}{\partial\beta\mathbf{r}}\left[m_{\mathcal{D}}(\beta\mathbf{r})^2 k_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r})\right] \\ &= 2m_{\mathcal{D}}(\beta\mathbf{r})k_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r})\frac{\partial}{\partial\beta\mathbf{r}}m_{\mathcal{D}}(\beta\mathbf{r}) + \frac{\partial}{\partial\beta\mathbf{r}}k_{\mathcal{D}}(\beta\mathbf{r}, \beta\mathbf{r})m_{\mathcal{D}}(\beta\mathbf{r})^2.\end{aligned} \tag{4.8}$$

The derivative of the DCV acquisition function is significantly more costly to evaluate than the objective, because it requires predictive gradients of the underlying GP. Instead of using a quadrature routine like `scipy.quad`, which would evaluate the integral for every dimension sequentially, we use Simpson's rule on 50 evenly spaced points between 0 and $\alpha_{\max}$ (defined below). Since these are multiple univariate integrals of a smooth function, the errors are practically negligible.

The scalar $\alpha_{\max}$ simultaneously constitutes an upper bound for the integration and the length of the exponential map. A bound is reasonable since longer exponential maps are slower to compute and the integration measure concentrates the mass near the center, so very far-away locations become irrelevant. For a sensible bound, we use the chi-square distribution:

$$\langle \alpha \cdot \mathbf{r}, \boldsymbol{\Sigma}^{-1} \alpha \cdot \mathbf{r} \rangle = \chi^2_p \tag{4.9}$$

by choosing a high value $p = 99.5\%$, we make sure that there is no significant amount of mass outside of this isoprobability contour. Note that this limit applies only to the computation of exponential maps and the collection of observations, not to the main quadrature itself.

Since $\mathbf{r}$ is constrained to lie on the unit hypersphere, we employ manifold gradient descent with a linesearch subroutine. Conveniently, the linesearch only evaluates the objective and not its gradient, which saves a significant amount of time. Overall, optimizing this acquisition function is costly, however. For completeness, we briefly describe the geometry of the unit (hyper)sphere. If the tangent space of our data manifold is $\mathcal{T}_{\boldsymbol{\mu}}\mathcal{M} = \mathbb{R}^D$, then a direction in this tangent space is a point on $\mathbb{S}^{D-1}$, which we represent as a unit norm vector in $\mathbb{R}^D$. For a point $\mathbf{x}$ on the sphere and a tangent vector $\boldsymbol{\xi}$, which lies in the plane touching the sphere tangentially, the exponential map is $\mathrm{Exp}_{\mathbf{x}}(\boldsymbol{\xi}) = \cos(\|\boldsymbol{\xi}\|_2)\mathbf{x} + \sin(\|\boldsymbol{\xi}\|_2)\frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|_2}$. However, the optimizer uses a cheaper *retraction map* $\mathrm{Retr}_{\mathbf{x}}(\boldsymbol{\xi}) = \frac{\mathbf{x}+\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|_2}$ instead of the exponential map to take a descent step. To obtain the gradient on the manifold, the Euclidean gradient is orthogonally projected onto the tangent plane.

Optimizing the DCV acquisition function is costly as it requires posterior mean predictions and predictive gradients of the GP inside the integration. Furthermore, confining observations to lie along straight lines implies that BQ may cover less space given a fixed number of function evaluations. Therefore, DCV will be useful in settings where exponential maps come at a high computational cost. Fig. 4.1 compares posteriors arising from standard WSABI-L using uncertainty sampling (simply referred to as WSABI-L) and DCV.

4.5 BQ for LAND

The known smoothness of the metric tensor makes the square exponential kernel (RBF) a suitable choice in most cases. However, for high-curvature manifolds, in particular in two dimensions, we found the Matérn-5/2 kernel to be slightly more stable, so we use it throughout instead of the RBF. Due to the strong smoothness prior, the RBF sometimes favored unreasonably small lengthscales.

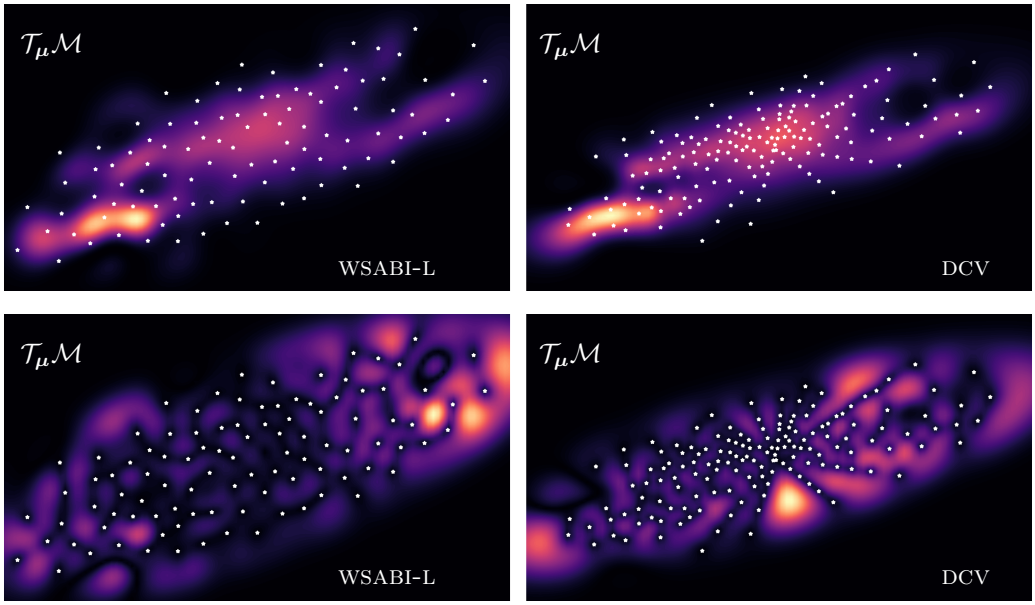


Figure 4.1: Posterior mean (top row) and variance (bottom row) over $g_{\boldsymbol{\mu}}(\mathbf{v})\mathcal{N}(\mathbf{v}; \mathbf{0}, \boldsymbol{\Sigma})$. We compare WSABI-L using uncertainty sampling (left) and DCV (right). Observed locations are scattered in white. The linear color scale represents the posterior magnitude. Visibly, DCV collects observations along straight lines emerging from the origin and thus has reduced exploration capability.

Depending on the employed Riemannian metric, we can set the constant prior mean of g to the known measure far from the data (Sec. 2.10). This amounts to the prior assumption that wherever there are no observations yet, the distance to the data is likely high.

The LAND optimization process requires the sequential computation of one integral per iteration as the parameters of the model get updated in an alternating manner. In general, elaborate schemes as in Xi et al. [2018] and Gessner et al. [2019] are available to estimate correlated integrals jointly. However, our integrand $g_{\boldsymbol{\mu}}$ does not depend on the covariance $\boldsymbol{\Sigma}$ of the integration measure π. Therefore, exponential maps remain unaltered when only the covariance $\boldsymbol{\Sigma}$ changes from one iteration to the next while the mean $\boldsymbol{\mu}$ remains fixed. BQ can thus reuse the observations from the previous iteration and only needs to collect a reduced number of new samples to account for the changed integration measure. This node reuse enables tremendous runtime savings.

Since we use the Matérn-5/2 kernel and we require further integrals for the LAND objective gradients, we use the GP as an emulator of the function we wish to model; that is, we do not calculate integrals analytically, but use extensive Monte Carlo (MC) sampling on top of the GP, which implies evaluating the posterior mean at the locations randomly drawn from the integration measure. To compute the integral without loss of precision, we use $S = 30{,}000$ samples to estimate the integrals. The time overhead and the approximation error of this procedure are negligible in practice.

We optimize the marginal likelihood of the GP with respect to the hyperparameters and use their final values to initialize the next iteration, since during the optimization the function changes smoothly from each step to the next. This information is not shared across the K components, but kept separately.

Our implementation of BQ builds upon the `bayesquad` python library [Wagstaff et al., 2018], which is available at `https://github.com/OxfordML/bayesquad`.

4.6 Further Considerations

4.6.1 Logarithmic Maps Initialization

When a mean $\boldsymbol{\mu}_k$ changes in an iteration, the logarithmic maps $\mathrm{Log}_{\boldsymbol{\mu}_k}(\mathbf{x}_n)$ are computed for all data points. This suggests the possibility of using these tangent vectors to initialize the BQ routine. Implementing the possibility to do so requires some care, for example due to possibly failing geodesic computations. Furthermore, it is more economical to use only a subset of inducing points of the datasets, as otherwise the $\mathcal{O}(N^3)$ GP computations will be too expensive. The inducing points are chosen by running K-means (e.g., $K = 20$) and then selecting the nearest datapoints. The metric at the inducing points can then be evaluated and stored when initializing the model. Overall, we found no significant improvement from this procedure, as many of the data points are not of real value to BQ when fitting a mixture model. The procedure could be adapted to take only points with significant responsibility w.r.t. the considered mean into account, but this would further complicate things for perhaps little gain. Consequently, we decided not to use this procedure for the experiments.

4.6.2 Log-Transform

An alternative to the square-root transform is the logarithmic transform, as suggested by Chai and Garnett [2019]. We added this possibility, along with WSABI-M to the `bayesquad` library, without analytic expressions for the complicated approximate integrals, however. We found the transform to be too extreme as it introduced artefacts, therefore ended up not using it further.

Chapter 5

Further LAND Improvements

5.1 Code

A significant part of this thesis in terms of time investment was a reimplementation of the LAND model in a modular fashion. The main class is the `LandMixture`, which has methods to initialize and fit the model, keeping track of most of the information. An object of this class further requires an instance of a `LandOptimizer`, where both a manifold optimizer and vanilla gradient descent are available for the covariance update. Secondly, an object of the `LandQuadrature` class is required, which can be either `MCQuadrature` or `BQuadrature`. These classes are responsible for storing reuse information and for the integral computation. `BQuadrature` uses the `BQWrapper` class as a mediator with the `bayesquad` library. The `BQWrapper` constructs the GP with the specified parameters (kernel etc.) and contains the acquisition loop. Since the LAND model is a complex computational pipeline with multiple error sources, we use extensive logging, so that each integration problem from a LAND fit can be reconstructed and visualized. Overall, debugging (playing with the solvers, looking at plots of the iterations, etc.) constitutes at least half of the total time investment of this thesis project.

5.2 Solver Chaining

To solve the geodesic equations, we combine two solvers, which have different strengths and weaknesses. By chaining them together, we obtain a more robust computational pipeline.

First, we make use of the fast and robust fixed-point solver (FP) introduced by Arvanitidis et al. [2019a]. This solver pursues a GP-based approach that avoids the often ill-behaved Jacobians of the geodesic ODE system. However, the resulting logarithmic maps are subject to significant approximation error, depending on the curvature of the manifold. The parameters of this solver are as follows:

Parameter	Value	Description
$\texttt{iter}_{\texttt{max}}$	1000	maximum number of iterations
`N`	10	number of mesh nodes.
`tol`	0.1	tolerance used to evaluate solution correctness.
σ	10^{-4}	noise of the GP.

For MNIST, we set $\texttt{iter}_{\texttt{max}} = 500$, and $\texttt{tol} = 0.2$, since this high-curvature manifold easily leads to failing geodesics.

The second solver we employ is a precise, albeit less robust one. This is the BVP solver available in the module `scipy.integrate.solve_bvp`. On high-curvature manifolds, this solver often fails (especially for long curves) and takes a significant amount of time to run. When it succeeds, however, the logarithmic maps are reliable. For this solver, we set the maximum number of mesh nodes to 100 and the tolerance to 0.1. We empirically found that choosing a high maximum number of mesh nodes (e.g., 500) can lead to high runtimes for failing geodesic computations.

To obtain fast and robust geodesics, these solvers may be chained together, i.e., we initialize the BVP solver with the FP solution, which is often worth the extra effort for speedup and improved robustness. For initialization, we use 20 mesh nodes, evenly spaced on the FP solution. If the FP solver already failed, it is very unlikely for the BVP solver to succeed, so we abort the computation.

Furthermore, we exploit previously computed BVP solutions: assume we want to compute $\text{Log}_{\boldsymbol{\mu}_t}(\mathbf{x})$. We search for past results $\text{Log}_{\boldsymbol{\mu}_t^*}(\mathbf{x})$, with $t^* < t$, $t^* = \arg\min \|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t*}\|$ and $\|\boldsymbol{\mu}_t - \boldsymbol{\mu}_{t*}\| < \epsilon_d$, where we choose $\epsilon_d = 0.5$. Since we compute logarithmic maps for data points $\mathbf{x}_{1:N}$, which do not change during LAND optimization, we can use them as hash keys in a dictionary, where we store the solutions. Looking up the solution is then linear in the number of previous LAND iterations. If such a solution is found, the FP is skipped and the solution is used to directly initialize the BVP solver.

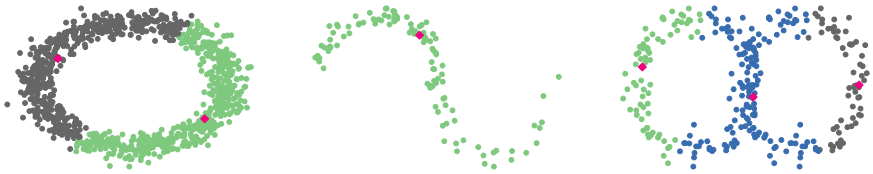
For the exponential maps, we use `scipy.integrate.solve_ivp` with a tolerance of 10^{-3}.

5.3 An Initialization Scheme

As the LAND optimization, especially with respect to the means, is computationally demanding, it is worth investing some effort in a good mean initialization.

A first, simple approach is to use a Euclidean Gaussian mixture model to initialize the means. However, this can easily place means far from the data, which makes both geodesic computations and the integration susceptible to failure.

We here propose a more complex, yet still relatively inexpensive initialization scheme. First, we build a Euclidean k-NN graph of the data and then reweigh the edges by their Riemannian lengths (by integrating the local length along the Euclidean straight line). We use this graph to approximate geodesic distances. Assume $K = 1$, so we need only one initial mean. A reasonable way is to choose the datum which minimizes the total (squared) distance to all other data. In the



(a) CIRCLE ($K = 2$) (b) CURLY ($K = 1$) (c) 2-CIRCLES ($K = 3$)

Figure 5.1: Visualization of the initialization scheme, showing the cluster means (◆). Data colored qualitatively to show the most responsible component.

case of squared distance, this is an empirical estimate for the *Karcher mean*, the Riemannian center of mass.

For the case $K > 1$, we first partition the graph by running *spectral clustering*. For this, we compute an affinity matrix with a Gaussian kernel (with fixed lengthscale $\rho = 1$) on the Riemannian graph distances. Then we select the means by considering the resulting clusters separately.

Empirically I found that this heuristic scheme leads to good initial locations of the means (see Figure 5.1), so that only few iterations of the LAND optimization are required. However, it requires the graph to be fully connected. If it is not, k may be increased or "dummy" edges added.

5.4 Manifold Linesearch

Arvanitidis et al. [2016] decomposed the precision $\mathbf{\Sigma}_k^{-1} = \mathbf{A}^\intercal\mathbf{A}$ for unconstrained optimization using gradient descent. We opt for a more principled approach by exploiting geometric structure of the symmetric positive definite (SPD) manifold, to which the covariance is confined. More specifically, we use the *bi-invariant* metric [Bhatia, 2009], one of several possible choices. Under this metric, geodesics from $\mathbf{A}$ to $\mathbf{B}$ may be parameterized as $\gamma(t) = \mathbf{A}^{\frac{1}{2}}\left(\mathbf{A}^{-\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\mathbf{A}^{-\frac{1}{2}}\right)^t\mathbf{A}^{\frac{1}{2}}, \quad 0 \leq t < 1$, and the distance from $\mathbf{A}$ to $\mathbf{B}$ is $d(\mathbf{A},\mathbf{B}) = \left\|\log \mathbf{A}^{-\frac{1}{2}}\mathbf{B}^{\frac{1}{2}}\mathbf{A}^{-\frac{1}{2}}\right\|_2$. The name stems from the fact that this distance is invariant under multiplication with any invertible square matrix $\mathbf{\Xi}$, i.e., $d(\mathbf{A},\mathbf{B}) = d(\mathbf{\Xi}\cdot\mathbf{A},\mathbf{\Xi}\cdot\mathbf{B})$. For manifold gradient descent, we calculate the Euclidean gradient $\nabla_{\mathbf{\Sigma}_k}$ and then project it onto the manifold by calculating $\frac{1}{2}\mathbf{\Sigma}_k\left(\nabla_{\mathbf{\Sigma}_k} + \nabla_{\mathbf{\Sigma}_k}^\intercal\right)\mathbf{\Sigma}_k$. We optimize with gradient descent and a deterministic manifold linesearch as a subroutine, which adaptively chooses its step lengths. This procedure as well as the SPD manifold are conveniently available in the `Pymanopt` [Townsend et al., 2016] library. We provide pseudocode for the covariance update in Alg. 2.

While we still need an initial step size, this hyperparameter is not important, as the linesearch then quickly adapts to the scale of the search landscape. In practice, we observe a faster and more robust optimization when using the linesearch, as opposed to vanilla gradient descent on the decomposed precision matrix.

As we have now assembled the whole optimization process, it is pertinent to give

the optimization hyperparameters of the optimizer, which are as follows:

Parameter	Value	Description
t_{max}	-	update each component $\texttt{t}_{\texttt{max}}$ times.
$\alpha^1_{\boldsymbol{\mu}}$	-	initial stepsize for mean updates.
$\epsilon_{\nabla_{\boldsymbol{\mu}}}$	-	tolerance for mean gradients
$\epsilon_{\mathcal{L}}$	2	likelihood tolerance
$t_{max,\boldsymbol{\Sigma}}$	4	max. $\boldsymbol{\Sigma}$ linesearch steps.
α_1	1.0	initial step size ($\boldsymbol{\Sigma}$ linesearch).
c_0	0.5	sufficient decrease factor ($\boldsymbol{\Sigma}$ linesearch).
c_1	0.5	contraction factor ($\boldsymbol{\Sigma}$ linesearch)

Cells with unspecified values (-) imply that the value of the respective parameter is not equal across all experiments and problems. Experiment-specific parameter details are in Table 6.3.

Algorithm 2 LAND update$_{\mathbf{\Sigma}_k}$ on the symmetric positive definite manifold $\mathcal{S}_+$

Input: Covariance $\mathbf{\Sigma}_k^t$, mean $\boldsymbol{\mu}_k$, max. linesearch iterations $t_{max,\mathbf{\Sigma}}$, last stepsize α_k, initial stepsize $\alpha_1 = 1.0$,
sufficient decrease factor $c_0 = 0.5$, contraction factor $c_1 = 0.5$
Output: $\mathbf{\Sigma}_k^{t+1}$, α_k (for reuse)

{define the exp. map on the $\mathcal{S}_+$ manifold, where $\mathbf{X}$ is an SPD matrix and $\mathbf{\Xi}$ is a tangent vector, i.e., a symmetric matrix}
Function $\mathrm{Exp}^+_{\mathbf{X}}(\mathbf{\Xi})$:
  return $\mathbf{X}^{\frac{1}{2}} \exp\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{\Xi}\mathbf{X}^{-\frac{1}{2}}\right)\mathbf{X}^{\frac{1}{2}}$, where exp denotes the matrix exponential.
EndFunction
{define the norm of a vector $\mathbf{\Xi}$ in the tangent space of $\mathbf{X} \in \mathcal{S}_+$}
Function $\left(\|\cdot\|^+_{\mathbf{X}}\right)(\mathbf{\Xi})$:
  $\mathbf{X} \leftarrow \mathbf{L}\mathbf{L}^\intercal$ {cholesky decomposition}
  return $\|\mathbf{L}^{-1}\mathbf{\Xi}\mathbf{L}^{-\intercal}\|_2$
EndFunction

for $i = 1$ **to** 2 {outer gradient descent loop} **do**
  Compute (or retrieve from cache) $\mathcal{L}(\mathbf{\Sigma}_k^t)$
  Compute (or retrieve from cache) Euclidean gradient $\nabla_{\mathbf{\Sigma}_k^t}\mathcal{L}(\Sigma_k^t)$ using Eq. (3.5)
  Obtain manifold gradient: $g \coloneqq \nabla_{\mathbf{\Sigma}_k^t;\mathcal{S}_+} = \frac{1}{2}\mathbf{\Sigma}_k^t\left(\nabla_{\mathbf{\Sigma}_k^t} + \nabla^\intercal_{\mathbf{\Sigma}_k^t}\right)\mathbf{\Sigma}_k^t$
  if α_k **is None or** $\alpha_k = 0$ **then**
    $\alpha_k \leftarrow \frac{\alpha_0}{\|g\|}$
  end if
  $\mathbf{\Sigma}_k^{t+1} \leftarrow \mathrm{Exp}^+_{\mathbf{\Sigma}_k^t}(-\alpha_k \cdot g)$
  Compute $\mathcal{C}_k(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k^{t+1})$
  Evaluate LAND objective $\mathcal{L}(\mathbf{\Sigma}_k^{t+1})$
  {Linesearch subroutine}
  $j \leftarrow 1$
  while $\mathcal{L}(\mathbf{\Sigma}_k^{t+1}) > \mathcal{L}(\mathbf{\Sigma}_k^t) - c_0 \cdot \alpha_k \cdot \|g\|^2$ and $j \leq t_{max,\mathbf{\Sigma}}$ **do**
    {while no sufficient decrease, contract}
    $\alpha_k \leftarrow \alpha_k \cdot c_1$
    $\mathbf{\Sigma}_k^{t+1} \leftarrow \mathrm{Exp}^+_{\mathbf{\Sigma}_k^t}(-\alpha_k \cdot g)$
    Compute $\mathcal{C}_k(\boldsymbol{\mu}_k, \mathbf{\Sigma}_k^{t+1})$
    Evaluate LAND objective $\mathcal{L}(\mathbf{\Sigma}_k^{t+1})$
    $j \leftarrow j + 1$
  end while
  if $\mathcal{L}(\mathbf{\Sigma}_k^{t+1}) > \mathcal{L}^t(\mathbf{\Sigma}_k^t)$ **then**
    $\alpha_k \leftarrow 0$
  end if
  if $j \neq 2$ **then**
    $\alpha_k = 1.3 \cdot \alpha_k$ {optimism}
  end if
  $t \leftarrow t + 1$
end for

Chapter 6

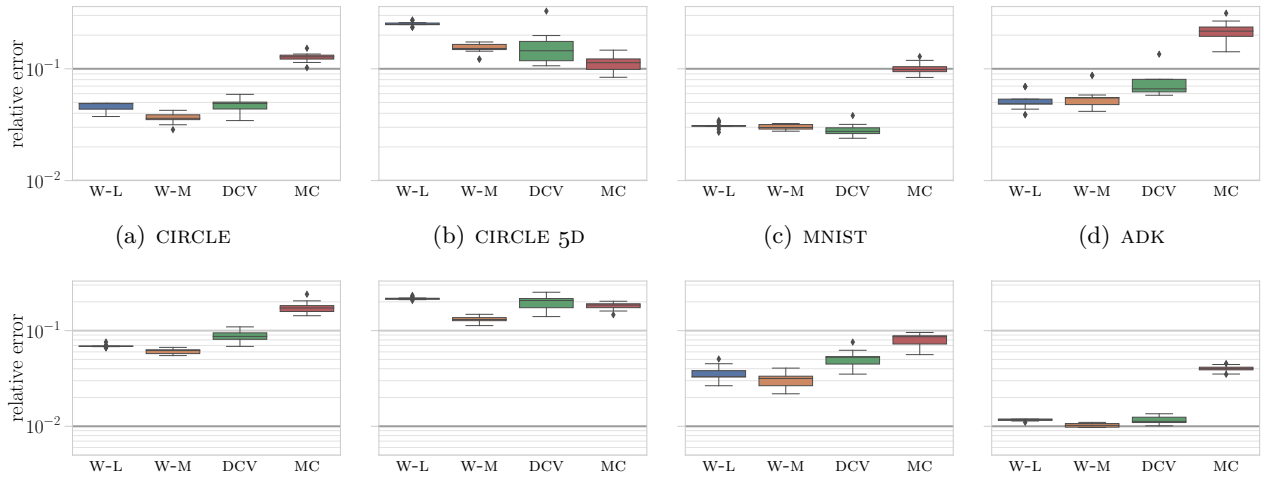
Experiments

We test the methods (WSABI-L, WSABI-M, DCV) on both synthetic and real-world data manifolds. Our aim is to show that Bayesian quadrature is faster compared to the Monte Carlo baseline, yet retains high accuracy. The experiments focus on the LAND model to illustrate practical use cases of Riemannian statistics. Furthermore, the iterative optimization process yields a wide range of integration problems of varying difficulty. In total, our experiments comprise 43,920 BQ integrations.

For different manifolds, we conduct two kinds of experiments: First, we fit the LAND model and record all integration problems arising during the optimization procedure. This allows us to compare the competitors on the whole problem, where BQ can benefit from node reuse. We fix the number of acquired samples for BQ and generate boxplots from the mean errors on the whole LAND fit for 16 independent runs (Fig. 6.1). Due to the alternating update of LAND parameters during optimization, *either* the integrand *or* the integration measure changes over consecutive iterations. We let WSABI-L and WSABI-M actively collect 80 in the former and 10 samples additionally to the reused ones in the latter case; for DCV, we fix 18 and 2 exponential maps, respectively, and acquire 6 points on each straight line. Integration cost for BQ is thus highly variable over iterations. Allocating a fixed runtime would not be sensible as BQ benefits from collecting more information after updates to the mean, a time investment that is over-compensated in the more abundant and—due to node reuse—cheap covariance updates. We choose sample numbers so as to allow for sufficient exploration of the space with practical runtime. For MC, we allocate the runtime budget of the mean slowest BQ method on that particular problem in order to compare accuracy over runtime. Mean runtimes for single integrations, averaged on entire LAND fits, are shown in Fig. 6.2 and mean exponential map runtimes, as computed by MC, are reported in 6.1.

Secondly, we focus on the first integration problem of each LAND fit in detail and compare the convergence behavior of the different BQ methods and MC over wall clock runtime (Fig. 6.3). We use the kernel metric (Sec. 2.10) when not otherwise mentioned. In the plot legends, we abbreviate WSABI-L/WSABI-M with W-L/W-M, respectively.

As the ground truth, we obtained $S = 40{,}000$ MC samples on each integration problem. Since obtaining a large number of exponential maps is computationally



(a) CIRCLE (b) CIRCLE 5D (c) MNIST (d) ADK

(e) CIRCLE 3D (f) CIRCLE 4D (g) CURLY (h) 2-CIRCLES

Figure 6.1: Boxplot error comparison (log scale, shared y-axis) of BQ and MC on whole LAND fit for different manifolds. For MC, we allocate the runtime of the mean slowest BQ method. Each box contains 16 independent runs.

Table 6.1: Mean exponential map runtimes in milliseconds, obtained by averaging over MC runtimes on the entire LAND fit.

CIRCLE	CIRCLE 5D	MNIST	ADK	CIRCLE 3D	CIRCLE 4D	CURLY	2-CIRCLES
60	50	238	68	32	45	62	36

extremely expensive, we subsampled from this pool of ground truth samples when MC samples were required in the experiments, instead of running MC again. For example, in the "error vs. runtime" experiment, we calculated the mean MC runtime per sample from the ground truth pool of this particular problem and then subsampled as many samples as the given runtime limit affords. For the boxplot experiments, we averaged the MC runtimes over the whole LAND fit and always obtained the same number of samples per integration. The MC runtime practically corresponds to the runtime of the exponential maps, since the overhead is minimal.

6.1 Toy Data

We generated three toy data sets and fitted the LAND model with pre-determined component numbers. Fig. 6.4 compares the resulting LAND fit to the Euclidean Gaussian mixture model (GMM) on a circle manifold with 1000 data points. Fig. 6.5 show the other synthetic data sets, one with a curly shape (CURLY) and a superposition of two circles (2-CIRCLES).

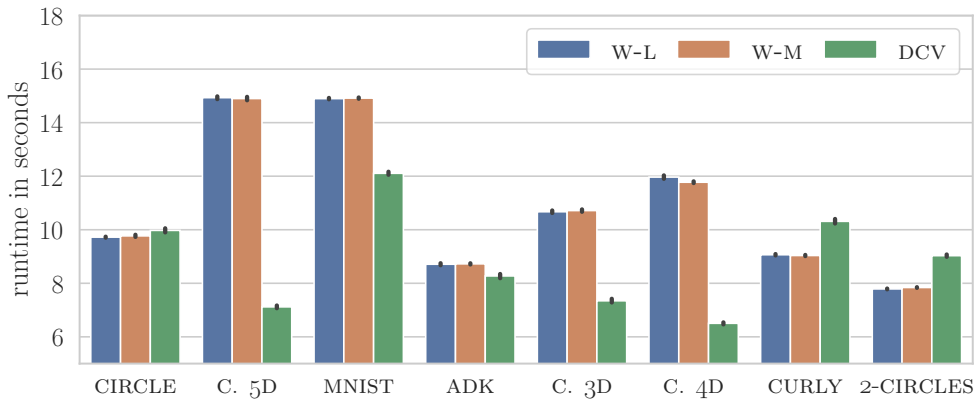


Figure 6.2: Mean runtime comparison (for a single integration) of the BQ methods. Errorbars indicate 95% confidence intervals w.r.t the 16 runs on each LAND fit. The reported runtimes belong to the boxplots in Fig. 6.1. The higher-dimensional CIRCLE datasets have been abbreviated as C.

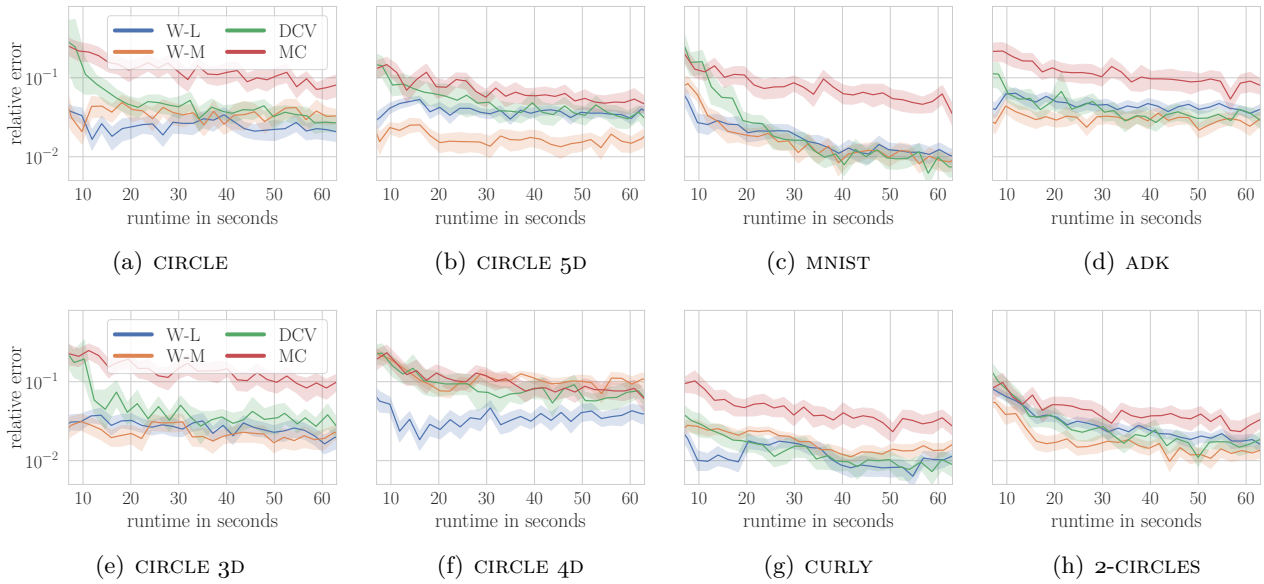
6.2 Higher-Dimensional Toy Data

With increasing number of dimensions, new challenges for metric learning and geodesic solvers appear. With the simple kernel metric, almost all of the volume will be far from the data as the dimension increases, a phenomenon which we observe already in relatively low dimensions. Such metric behavior can lead to pathological integration problems, as the integrand may then become almost constant. In this experiment, we embed the circle toy data in higher dimensions by sampling random orthonormal matrices. After projecting the data, we add Gaussian noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2 = 0.01)$ and standardize.

6.3 MNIST

We trained a Variational Auto-Encoder on the first three digits of MNIST using the surrogate metric (Sec. 2.10) and found that the LAND is able to distinguish the three clusters more clearly than a Euclidean Gaussian mixture model, see Fig. 6.6. The LAND favors regions of higher density, where the VAE has more training data. In this experiment, the gain in speed of BQ is even more pronounced, since exponential maps are slow due to high curvature. MC with 1000 samples achieves 2.78% mean error on the whole LAND fit with a total runtime of 6 hours and 56 minutes, whereas DCV (18/2 exponential maps) achieves 2.84% error within 21 minutes; a speedup by a factor of ≈ 20.

Technical Details We sampled 5,504 random data points from the first three digits of MNIST [LeCun et al., 1998], which were preprocessed by normalizing them feature-wise to $[-1, +1]$ using `sklearn.preprocessing.MinMaxScaler`. We trained a simple Variational-Autoencoder (VAE) to embed the 784 dimensional input in a latent space of dimension 2. The architecture uses separate encoders $\boldsymbol{\mu}_\phi$, $\boldsymbol{\sigma}_\phi$ and

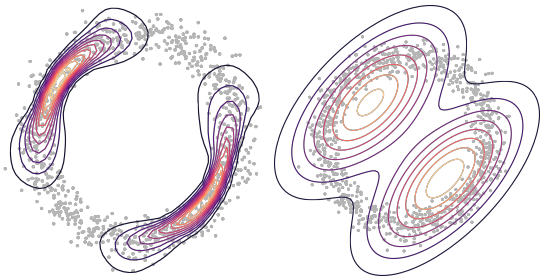


(a) CIRCLE (b) CIRCLE 5D (c) MNIST (d) ADK

(e) CIRCLE 3D (f) CIRCLE 4D (g) CURLY (h) 2-CIRCLES

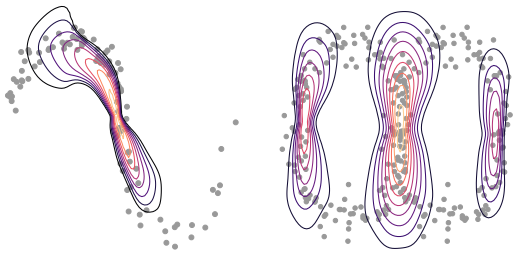
Figure 6.3: Comparison of BQ and MC errors against runtime (vertical log scale, shared legend and axes) for different manifolds, on the first integration problem of the respective LAND fit. Shaded regions indicate 95% confidence intervals w.r.t. the 30 independent runs.

decoders $\boldsymbol{\mu}_\theta$, $\boldsymbol{\sigma}_\theta$. The architecture is summarized in Table 6.2 We trained the network for 200 epochs using ADAM with a learning rate of 10^{-3}. The resulting latent codes were used to construct the surrogate metric, with $\rho = 0.001$, such that the measure far from the data is 1000. The small variances cause high curvature, which makes the integration tasks challenging and geodesic computations slow. To fit the LAND, we used 250 subsampled points to lower the amount of time spent on BVPS. In contrast, the GMM was fitted on the whole 5,504 points. Note that Fig. 6.6 shows this training data.



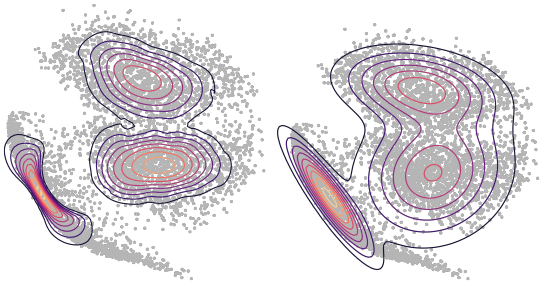
(a) LAND (b) GMM

Figure 6.4: Model comparison on CIRCLE toy data.



(a) CURLY (b) 2-CIRCLES

Figure 6.5: Further toy data LAND fits



(a) LAND (b) GMM

Figure 6.6: Model comparison on three-digit MNIST.

Encoder/Decoder	Layer 1	Layer 2	Layer 3
$\boldsymbol{\mu}_\phi$	128 (tanh)	64 (tanh)	2 (linear)
$\boldsymbol{\sigma}_\phi$	128 (tanh)	64 (tanh)	2 (softplus)
$\boldsymbol{\mu}_\theta$	64 (linear)	128 (linear)	784 (linear)
$\boldsymbol{\sigma}_\theta$	64 (linear)	128 (linear)	784 (softplus)

Table 6.2: Architecture of the VAE on MNIST. The columns contain the neuron count of the fully connected layers and the activation functions in parantheses.

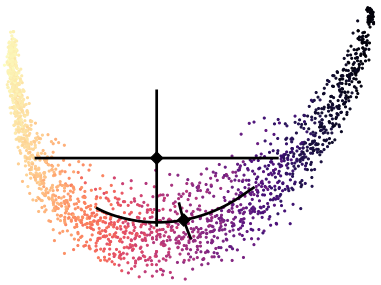


Figure 6.7: Comparison of the Euclidean Gaussian vs. LAND mean and eigenvectors on ADK data. Data is colored according to the *radius of gyration*, a measure indicating how "open" the protein is, providing a visual argument for the manifold hypothesis.

6.4 ADK

The dataset, as well as the preprocessing, are described in Section 1.1. The LAND is a suitable model on this data, as it can be used to visualize the conformational landscape and generate realistic samples. Plausible interpolations (trajectories) between conformations may be conceived of as geodesics under the Riemannian metric.

We model the ADK manifold with high curvature and large measure far from the data to account for the knowledge that realistic trajectories lie closely together ($\sigma = 0.035$ and $\rho = 10^{-5}$). This makes for a challenging integration problem, since most mass is near the data boundary due to extreme metric values.

A single-component LAND yields a representative state for the transition between the closed and open conformation. Whereas the Euclidean mean falls outside the data manifold, the LAND mean is reasonably situated. Plotting the eigenvectors of the covariance matrix makes it clear that the LAND captures the intrinsic dimensions of the data manifold (Fig. 6.7) and that the mean interpolates between the closed and open state (Fig. 1.1). Our aim here is to demonstrate that molecular dynamics is an exciting application area for Riemannian statistics and sketch potential experiments, which are then for domain experts to design.

6.5 Interpretation

We find that BQ consistently outperforms MC in terms of speed. Even on high-curvature manifolds with volume elements spanning multiple orders of magnitudes, such as MNIST and ADK, the GP succeeds to approximate the integrand well. Among the different BQ candidates, we cannot discern a clear winner, since their performance depends on the specific problem geometry and exponential map runtimes. DCV performs especially well when geodesic computations are costly, such as for MNIST. We note that geodesic solvers and metric learning are subject to new challenges in higher dimensions, which merit further research effort.

Parameter	CIRCLE	CIRCLE 3D	CIRCLE 4D	CIRCLE 5D	MNIST	ADK	CURLY	2-CIRCLES
σ	0.1	0.25	0.25	0.25	-	0.035	0.2	0.15
ρ	0.001	0.01	0.0316	0.063	0.001	0.00001	0.01	0.01
K	2	2	2	2	3	1	1	3
t_{max}	7	4	4	4	7	7	7	7
$\alpha^1_{\boldsymbol{\mu}}$	0.3	0.3	0.3	0.3	0.3	0.2	0.3	0.3
$\epsilon_{\nabla_{\boldsymbol{\mu}}}$	0.01	0.01	0.01	0.01	0.015	0.01	0.01	0.01
integrations	67	39	40	34	105	36	33	111

Table 6.3: Manifold and LAND optimization hyperparameters and resulting number of integration problems.

6.6 Technical Details

All experiments were run in a cloud setting on 8 virtual CPUs. We restricted the core usage of BLAS linear algebra subroutines to a single core, so as not to create interference between multiple processes.

Optimization Hyperparameters In Table 6.3, we report the relevant hyperparameters for the metrics (σ, ρ), which were used to construct the manifolds, and those optimization parameters which are not equal across all problems.

DCV Parameters The gradient descent is allowed a maximum of 15 steps in the "error vs. runtime experiment", whereas in the boxplot experiment we decrease this number to 5, as this experiment focuses more on speed given a fixed number of samples. The linesearch may use up to 5 steps. We set the optimism of the linesearch to 2.0 and the initial stepsize to 1.0. If a descent step has norm less than 10^{-10}, the optimization is aborted.

After an exponential map is computed according to DCV, we discretize the resulting straight line in the tangent space into 30 evenly spaced points and sequentially select 6 points using the standard WSABI objective, updating the GP after each observation.

Boxplot Experiments (Fig. 6.1) These experiment were conducted on whole LAND fits, with 16 independent runs for each of the 3 BQ methods. From Table 6.3, we can easily calculate the total number of runs as $48 \cdot (67 + 39 + 40 + 34 + 105 + 36 + 33 + 111) = 22{,}320$.

Error vs. Runtime Experiments (Fig. 6.3) We evenly space 30 runtime limits between 5 and 65 seconds using `np.linspace(5., 65., 30)`. For each of these runtime limits, we let each BQ method run 30 times. BQ will stop collecting more samples as soon as the runtime limit is reached. After this, however, it will take some more time to finalize, as an ongoing computation is not interrupted. We then record the actually resulting runtimes and average over the 30 runs. These averages are then used for the x-axes of the plots, whereas the mean relative error is on the

y-axes. In total, each BQ method thus has 900 runs on each problem. The 8 plots, 4 in the main paper and 4 in the supplementary, contain $3 \cdot 900 \cdot 8 = 21{,}600$ runs. Together with the boxplot experiments, we obtain $21{,}600 + 22, = 43{,}920$ BQ runs, that is, 14,640 for each of the 3 methods.

In Fig. 6.3(c), we removed 4 extreme DCV outliers, where seemingly the GP "broke". This amounts to $\frac{4}{21{,}600} = 0.01852\%$ of the BQ runs in the 8 plots.

Chapter 7

Discussion and Outlook

Riemannian statistics is the appropriate framework to model real data with nonlinear geometry. Yet, its wide adoption is hampered by the prohibitive cost of numerical computations required to learn geometry from data and operate on manifolds. In this work, we have demonstrated on the example of numerical integration the great potential of probabilistic numerical methods (PNM) to reduce this computational burden. PNM adaptively select actions in a decision-theoretic manner and thus handle information with greater care than classic methods, e.g., Monte Carlo. Consequently, the deliberate choice of informative computations saves unnecessary operations on the manifold.

We have extended Bayesian quadrature to Riemannian manifolds, where it outperforms Monte Carlo over a large number of integration problems owing to its increased sample efficiency. Beyond known active learning schemes for BQ, we have introduced a version of uncertainty sampling adapted to the manifold setting that allows to further reduce the number of expensive geodesic evaluations needed to estimate the integral.

Numerical integration is just one of multiple numerical tasks in the context of statistics on Riemannian manifolds where PNM suggest promising improvements. The key operations on data manifolds are geodesic computations, i.e., solutions of ordinary differential equations. Geodesics have been viewed through the PN lens, e.g., by Hennig and Hauberg [2014], but still offer a margin for increasing the performance of statistical models such as the considered LAND.

Once multiple PNM are established for Riemannian statistics, the future avenue directs towards having them operate in a concerted fashion. As data-driven Riemannian models rely on complex computation pipelines with multiple sources of epistemic and aleatory uncertainty, their robustness and efficiency can benefit from modeling and propagating uncertainty through the computations.

All in all, we believe the coalition of *geometry*- and *uncertainty*-aware methods to be a fruitful endeavor, as these approaches are united by their common intention to respect structure in data and computation that is otherwise often neglected.

Bibliography

Andrea Amadei, Antonius BM Linssen, and Herman JC Berendsen. Essential dynamics of proteins. *Proteins: Structure, Function, and Bioinformatics*, 17(4): 412–425, 1993.

Georgios Arvanitidis. *Geometrical Aspects of Manifold Learning*. PhD thesis, DTU Compute, Technical University of Denmark (DTU), 2019.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. A Locally Adaptive Normal Distribution. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4251–4259, 2016.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Georgios Arvanitidis, Søren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1506–1515. PMLR, 2019a. URL http://proceedings.mlr.press/v89/arvanitidis19a.html.

Georgios Arvanitidis, Søren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 1506–1515. PMLR, 2019b. URL http://proceedings.mlr.press/v89/arvanitidis19a.html.

Georgios Arvanitidis, Søren Hauberg, and Bernhard Schölkopf. Geometrically enriched latent spaces. In *arXiv preprint*, 2020.

Rajendra Bhatia. *Positive Definite Matrices*, volume 24. Princeton University Press, 2009.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

François-Xavier Briol, Chris J Oates, Mark Girolami, Michael A Osborne, Dino Sejdinovic, et al. Probabilistic integration: A role in statistical computation? *Statistical Science*, 34(1):1–22, 2019.

Henry R. Chai and Roman Garnett. Improving quadrature for constrained integrands. In Kamalika Chaudhuri and Masashi Sugiyama, editors, *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, volume 89 of *Proceedings of Machine Learning Research*, pages 2751–2759. PMLR, 2019.

WK Clifford. On the hypotheses which lie at the bases of geometry. *General Theory of Relativity: The Commonwealth and International Library: Selected Readings in Physics*, page 107, 2013.

Jon Cockayne, Chris Oates, T. J. Sullivan, and Mark Girolami. Bayesian probabilistic numerical methods. *SIAM Review*, 61(4):756 – 789, 2019. doi: 10.1137/17M1139357.

Manfredo Perdigao do Carmo. *Riemannian Geometry*. Birkhäuser, 1992.

Christian Fröhlich, Alexandra Gessner, Philipp Hennig, Bernhard Schölkopf, and Georgios Arvanitidis. Bayesian Quadrature on Riemannian data manifolds. *arXiv preprint arXiv:2102.06645*, 2021.

Alexandra Gessner, Javier Gonzalez, and Maren Mahsereci. Active multi-information source Bayesian quadrature. In Amir Globerson and Ricardo Silva, editors, *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, volume 115 of *Proceedings of Machine Learning Research*, pages 712–721. AUAI Press, 2019.

Jeremy Gray. Epistemology of Geometry. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2019 edition, 2019.

Tom Gunter, Michael A. Osborne, Roman Garnett, Philipp Hennig, and Stephen J. Roberts. Sampling for inference in probabilistic models with fast Bayesian quadrature. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2789–2797, 2014.

Søren Hauberg, Oren Freifeld, and Michael J. Black. A geometric take on metric learning. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2033–2041, 2012.

P. Hennig, M. A. Osborne, and M. Girolami. Probabilistic numerics and uncertainty in computations. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 2015.

Philipp Hennig and Søren Hauberg. Probabilistic solutions to differential equations and their application to Riemannian statistics. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, AISTATS 2014, Reykjavik, Iceland, April 22-25, 2014*, volume 33 of *JMLR Workshop and Conference Proceedings*, pages 347–355. JMLR.org, 2014.

Andrew Janiak. Kant's Views on Space and Time. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2020 edition, 2020.

Jürgen Jost. *Bernhard Riemann: Über die Hypothesen, welche der Geometrie zu Grunde liegen*. Springer, Berlin, 2013.

Motonobu Kanagawa and Philipp Hennig. Convergence guarantees for adaptive Bayesian quadrature methods. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6234–6245, 2019.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Guy Lebanon. Learning Riemannian metrics. *arXiv preprint arXiv:1212.2474*, 2012.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

John Lee. *Introduction to Riemannian Manifolds*. Springer, 2018.

Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of Statistical Planning and Inference*, 29(3):245–260, 1991.

Elena Papaleo, Paolo Mereghetti, Piercarlo Fantucci, Rita Grandori, and Luca De Gioia. Free-energy landscape, principal component analysis, and structural clustering to identify representative conformations from molecular dynamics simulations: the Myoglobin case. *Journal of Molecular Graphics and Modelling*, 27(8): 889–899, 2009.

Xavier Pennec. Intrinsic statistics on Riemannian manifolds: Basic tools for geometric measurements. *Journal of Mathematical Imaging and Vision*, 25(1):127, 2006.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1278–1286. JMLR.org, 2014.

Bernhard Riemann. Über die Hypothesen, welche der Geometrie zu Grunde liegen. *Königliche Gesellschaft der Wissenschaften und der Georg-Augustus-Universität Göttingen*, 13(133):1867, 1854.

Salem Said, Lionel Bombrun, Yannick Berthoumieu, and Jonathan H Manton. Riemannian Gaussian distributions on the space of symmetric positive definite matrices. *IEEE Transactions on Information Theory*, 63(4):2153–2170, 2017.

Frederic Schuller. A thorough introduction to the theory of general relativity, 2015. The WE-Heraeus International Winter School on Gravity and Light.

Sean L. Seyler, Avishek Kumar, M. F. Thorpe, and Oliver Beckstein. Path similarity analysis: A method for quantifying macromolecular pathways. *PLOS Computational Biology*, 11(10):1–37, 10 2015. doi: 10.1371/journal.pcbi.1004568. URL https://doi.org/10.1371/journal.pcbi.1004568.

Florian Sittel, Abhinav Jain, and Gerhard Stock. Principal component analysis of molecular dynamics: On the use of Cartesian vs. internal coordinates. *The Journal of chemical physics*, 141(1):07B605_1, 2014.

Stefan Sommer, Tom Fletcher, and Xavier Pennec. Introduction to differential and Riemannian geometry. In *Riemannian Geometric Statistics in Medical Image Analysis*, pages 3–37. Elsevier, 2020.

Nicholas Spellmon, Xiaonan Sun, Nualpun Sirinupong, Brian Edwards, Chunying Li, and Zhe Yang. Molecular dynamics simulation reveals correlated inter-lobe motion in protein Lysine Methyltransferase SMYD2. *PLOS ONE*, 10(12):e0145758, 2015.

Nicholas F. Stang. Kant's Transcendental Idealism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2021 edition, 2021.

Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Alessandra Tosi, Søren Hauberg, Alfredo Vellido, and Neil D. Lawrence. Metrics for probabilistic geometries. In Nevin L. Zhang and Jin Tian, editors, *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pages 800–808. AUAI Press, 2014.

James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *The Journal of Machine Learning Research*, 17(1):4755–4759, 2016.

Gareth A. Tribello and Piero Gasparotto. Using dimensionality reduction to analyze protein trajectories. *Frontiers in Molecular Biosciences*, 6:46, 2019. ISSN 2296-889X. doi: 10.3389/fmolb.2019.00046.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

Ed Wagstaff, Saad Hamid, and Michael Osborne. Batch selection for parallelisation of Bayesian quadrature. *arXiv preprint arXiv:1812.01553*, 2018.

Wanja Wiese and Thomas K. Metzinger. Vanilla PP for philosophers: A primer on predictive processing. In Thomas K. Metzinger and Wanja Wiese, editors, *Philosophy and Predictive Processing*, chapter 1. MIND Group, Frankfurt am Main, 2017. ISBN 9783958573024. doi: 10.15502/9783958573024.

Antje Wolf and Karl N Kirschner. Principal component and clustering analysis on molecular dynamics data of the ribosomal l11 · 23s subdomain. *Journal of Molecular Modeling*, 19(2):539–549, 2013.

Xiaoyue Xi, François-Xavier Briol, and Mark A. Girolami. Bayesian quadrature for multiple related integrals. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5369–5378. PMLR, 2018.

Selbständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Masterarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Ort, Datum Unterschrift