Gaussian Processes and Statistical Decision-making in Non-Euclidean Spaces

Alexander Terenin

Department of Mathematics

Imperial College London

a dissertation submitted for the degree of

Doctor of Philosophy

February 2022

Declaration

This dissertation and all research contained in it are a product of my original work, except where indicated otherwise by explicit statement or reference. All ideas, quotations, and data originating from the works of others, published or otherwise, are fully acknowledged according to standard practices of the academic discipline.

Copyright

Acknowledgments

This thesis is dedicated to the several hundred Twitter users who thought what I was working on was interesting enough to give it a read, or a comment. Thanks to you, my experiment in writing a thesis in an open-source manner visible to the public was a resounding success. Getting the opportunity to write this thesis has been a dream come true, and I am delighted to share the experience with you.

The original acknowledgments I had planned to write during most of my doctoral work would have started on a far less positive note. The sheer volume of kind feedback I've received in the weeks during which I was writing this thesis, almost all of it from people I've never met, has convinced me to leave the past behind, and not write about certain people, actions, and events that do not deserve to be remembered. Instead, I choose to thank those due to whom I had the chance.

Firstly, I am profoundly grateful to Marc Deisenroth for taking me as a student, and giving me a *third* chance to complete a Ph.D., and to Seth Flaxman for taking me as a co-supervisee in order to make this possible. Most people who do not succeed the first time, whatever the reason may be, don't even get a second chance, much less a third one. You believed in me in the darkest moments of my academic career, and anything that I ever accomplish in my career will only have been possible because of that belief.

Secondly, I am also profoundly grateful to David Draper for believing in me, and taking me as a student when nobody was willing to talk to me or take my ideas seriously. It is thanks to you that I have a master's degree, and though I was not able to finish a doctorate at the time under your guidance, I hope this thesis shows that you did the right thing in supporting me, even when it became clear I had no academic future at the place I was at, and later in helping me return to scientific work.

Thirdly, I am grateful to Eric Xing for hiring me when my career was otherwise in the midst of full-blown chaos. The time I spent at Petuum convinced me that the machine learning community was the right intellectual home for me, and prompted me to look beyond the narrow scope of ideas I had previously been around. Though I was not truly ready to be an independent researcher at that time, and therefore accomplished little, I am grateful for the opportunity and all that I learned from it.

I thank my thesis committee members, Michael Bronstein and Stefano Ermon, for examining me and for their comments during the viva. Your careful reading helped significantly improve the quality of this thesis. I am particularly grateful for Michael's comment that "[t]he introductory chapters, with enough effort, could probably be transformed into a book." These are among the kindest words my work has ever received.

I also thank my large and ever-growing set of scientific collaborators. You all give me confidence that, together, we can solve all of the hardest and most important problems! I am constantly learning from each and every one of you, and none of the work in this thesis would have been possible without your support, ideas, and opportunity to discuss.

I am particularly grateful to Viacheslav Borovitskiy for both the opportunity to collaborate on all of the work presented in this thesis, and to reconnect with my heritage by constantly practicing my Russian. Together, we were able to write substantially stronger work than any of us, or at least certainly I, would have alone—the awards we've won together speak for themselves. A large part of why I have gotten better at practicing mathematics over the years—a skill of fundamental importance—is thanks to working with you.

I thank James T. Wilson, Steindór Sæmundsson, Samuel Cohen, and So Takao, for the opportunity to do research and write together. I am grateful to Yicheng Luo, Sanket Kamthe, Hugh Salimbeni, Yasemin Bekiroğlu, Sicelukwanda Zwane, K. S. Sesh Kumar, and others in the group who have spoken to me at length about ideas during my time. I feel fortunate to to have been your labmate, and have learned an immense amount from you over the years.

I thank Andreas Hochlehnert, Kai Biegun, and Lucas Cosier for trusting in me as a researcher enough to allow me to supervise your work. Watching your ideas develop and grow has been a beautiful experience, and I look forward to seeing you shine in the years to come.

I am grateful to Michael Hutchinson, James-Michael Leahy, Peter Mostowsky, Iskander Azangulov, and others who I have collaborated with on papers, or spoken to at length about ideas. I am particularly grateful to Brandon Amos both for the opportunity to collaborate, and for suggesting I speak to Marc Deisenroth in the weeks leading up to me becoming his student—without this, my path would have been very different. I thank Nick Sharp for teaching me how to make the three-dimensional figures in this thesis.

I am grateful to Imperial College, and the Department of Mathematics in particular, for both the opportunity in funding my studies, and for support when things did not go as planned. I am particularly thankful to Henrik Jensen: thank you for treating me with kindness in difficult times. I am grateful to Pierre Degond, Darryl Holm, and Chris Hallsworth for your ideas in teaching me mathematics. I joined Imperial because I wanted to get better at mathematics, and it is thanks largely to you that I feel I succeeded at this.

I am thankful to Måns Magnusson, Leif Jonsson, and Shawfeng Dong for collaborating with me in my early days as a researcher, and helping me learn and get my bearings together. In particular, I thank Peter Drake and Raya Feldman for first setting me on this path. I am grateful to Matthew Johnson and Chris De Sa for the counterexamples sent to me in the early times, which convinced me that proper mathematics was of utmost importance and that I had to improve at it in order trust myself to say things that are true.

I am grateful to my friends at Carnegie Mellon University, including Dominic Chen, Aurick Qiao, Willie Neiswanger, Kumail Jaffer, Ziv Scully, Sarah Allen Scully, Sol Boucher, Guillaume Didier, Stefan Muller, Priya Donti, Gabriele Farina, Noam Brown, Ben Blum, Evan Cavallo, and others who convinced me to apply to a PhD program by showing me that many that of the people spending their life studying ideas were just like me. I am grateful to Yuanran Zhu for showing me the same, but in a different place and time.

I am grateful to friends who have supported me over the years, including to Jon Frost for almost a decade of friendship, as well as Paula Siauchó Unriza, Victor Espinosa, Sam Aragon, and others who know me well and have been there for me over the years. Even as time had gone on, whether we last spoke days or years ago, I have always found nothing about our friendship to have changed whenever we have had the chance to speak again.

Finally, I want to thank my family, including my mom, Irina Terenina. Without your sacrifices, I would not have spent my youth in the United States, or had almost any of the opportunities I have had. I am grateful to my grandmother, Olga Novikova, for your wisdom and for keeping our family together. I wish you could have seen me graduate and write this thesis.

I am grateful to my dad, Vadim Terenin: your authoritative presentation style has at times blinded me to your ideas. I am grateful to my stepmom, Rheesa Eddings, for both your deeply insightful way of understanding the world, and for your humanity, diplomacy, kindness, helpful ideas, and support over the years.

My gratitude to family includes those who are family by virtue of their friendship and support for my entire adult life, including Jerry Hirsch, John Halper, Lois Morera, Christian Morera, and Peter Morera. Thank you for continuing to stay in touch, for being the extended family I never otherwise had, and for being with me as my journey has unfolded.

Though this list of acknowledgments has ballooned beyond what might otherwise be expected in a dissertation, I nonetheless feel compelled to write it in full by virtue of my path being what it was. It is also without a doubt incomplete, so I am grateful to anyone whose name should have been here but has been accidentally omitted, including whoever suggested I limit all paragraphs written to seven lines or less. To conclude, I offer you, the reader, my exceeding gratitude for taking the time to read my work and ideas.

Abstract

Bayesian learning using Gaussian processes provides a foundational framework for making decisions in a manner that balances what is known with what could be learned by gathering data. In this dissertation, we develop techniques for broadening the applicability of Gaussian processes. This is done in two ways.

Firstly, we develop pathwise conditioning techniques for Gaussian processes, which allow one to express posterior random functions as prior random functions plus a dependent update term. We introduce a wide class of efficient approximations built from this viewpoint, which can be randomly sampled once in advance, and evaluated at arbitrary locations without any subsequent stochasticity. This key property improves efficiency and makes it simpler to deploy Gaussian process models in decision-making settings.

Secondly, we develop a collection of Gaussian process models over non-Euclidean spaces, including Riemannian manifolds and graphs. We derive fully constructive expressions for the covariance kernels of scalar-valued Gaussian processes on Riemannian manifolds and graphs. Building on these ideas, we describe a formalism for defining vector-valued Gaussian processes on Riemannian manifolds. The introduced techniques allow all of these models to be trained using standard computational methods.

In total, these contributions make Gaussian processes easier to work with and allow them to be used within a wider class of domains in an effective and principled manner. This, in turn, makes it possible to potentially apply Gaussian processes to novel decision-making settings.

Table of Contents

1 Introduction. 21
1.1 Bayesian learning 23
1.1.1 Review of probability theory 24
1.1.2 Bayes' Rule for probability measures 26
1.1.3 Technical lemmas 32
1.2 Statistical decision-making 34
1.2.1 Markov decision processes 34
1.2.2 Multi-armed bandits. 38
1.2.3 Bayesian optimization 45
1.3 Gaussian processes 47
1.3.1 Review of vector spaces 49
1.3.2 Gaussian random variables 51
1.3.3 Gaussian random vectors 52
1.3.4 Gaussian random functions 55
1.3.5 Gaussian processes in the sense of duality 59
1.4 Discussion. 64

2 Pathwise Conditioning 67
2.1 Conditioning multivariate Gaussians 68
2.1.1 Distributional conditioning 68
2.1.2 Pathwise conditioning 69
2.2 Conditioning Gaussian processes 73
2.2.1 Distributional conditioning 73
2.2.2 Pathwise conditioning 75
2.3 Sampling from prior Gaussian processes 79
2.3.1 Random feature methods 79
2.3.2 Karhunen–Loève expansions. 82

2.3.3 Finite element methods 84
2.4 Approximating pathwise data-dependent terms 86
2.4.1 Inducing points 87
2.4.2 Approximate priors 90
2.5 Error analysis 92
2.6 Parallel Thompson sampling 96
2.7 Conclusion 100

3 Non-Euclidean Matérn Gaussian Processes 103
3.1 Riemannian Matérn Gaussian processes 104
3.1.1 Review of differential geometry 104
3.1.2 The Laplace–Beltrami operator 107
3.1.3 A no-go theorem for kernels on manifolds 109
3.1.4 Stochastic partial differential equations 111
3.1.5 The Riemannian Matérn kernel 113
3.1.6 Illustrated examples 120
3.2 Graph Matérn Gaussian processes 121
3.2.1 Review of graph theory 121
3.2.2 The graph Laplacian. 122
3.2.3 The graph Matérn kernel 124
3.2.4 Illustrated examples 127
3.3 Gaussian vector fields on Riemannian manifolds 129
3.3.1 Vector fields on manifolds. 129
3.3.2 Coordinate representations 134
3.3.3 Projected kernels 135
3.3.4 Illustrated examples 137
3.4 Geometry-aware Bayesian optimization 139
3.5 Conclusion 142

4 Discussion 145
4.1 Future work 146
4.2 Conclusion 150

References 151

List of Figures

1.1 Bayesian models 28
1.2 Markov decision processes 35
1.3 Multi-armed bandits. 39
1.4 Upper confidence bound algorithm 42
1.5 Bayesian optimization using Thompson sampling 46
1.6 Gaussian distributions 52
1.7 Multivariate Gaussian densities 53
1.8 Multivariate Gaussian quantile ellipsoids. 54
1.9 Gaussian processes 56

2.1 Distributional conditioning of multivariate Gaussians. . . . 69
2.2 Pathwise conditioning of multivariate Gaussians. 70
2.3 Distributional conditioning of Gaussian processes 74
2.4 Pathwise conditioning of Gaussian processes 76
2.5 Approximate pathwise conditioning. 77
2.6 Random Fourier feature prior approximations. 81
2.7 Karhunen–Loève prior approximations 84
2.8 Finite element prior approximations 86
2.9 Variational approximations of Gaussian processes 88
2.10 Canonical basis functions. 89
2.11 Variance starvation in Fourier feature models. 91
2.12 Parallel Thompson sampling benchmark results 98

3.1 Common manifolds 105
3.2 Laplace–Beltrami eigenfunctions: circle and torus 108
3.3 Laplace–Beltrami eigenfunctions: sphere and dragon manifold 109
3.4 Matérn kernel: circle and torus 114
3.5 Matérn kernel: sphere and dragon manifold 115
3.6 Posterior Gaussian process: dragon manifold 120

3.7 Graph Matérn kernels 125
3.8 Limits of graph Matérn kernels 126
3.9 Posterior Gaussian process: traffic data, global view 128
3.10 Posterior Gaussian process: traffic data, local view 128
3.11 Gaussian vector fields 130
3.12 Representing vector fields in different frames 134
3.13 Projected kernels 137
3.14 Posterior Gaussian vector field: climate data 138
3.15 Global optimization benchmark functions 140
3.16 Geometry-aware Bayesian optimization benchmark results. . 141

Principal Notation

General

$(\Omega, \mathcal{F}, \mathbb{P})$	Probability space.	24
$\mathbb{E}(\cdot)$	Expectation of a real-valued random variable	25
$\mathrm{Cov}(\cdot, \cdot)$	Covariance between two real-valued random variables	25
$y \sim \pi$	Random variable y with distribution π	24
$\mathcal{M}_1(X)$	Space of probability measures over X.	30
$D_{\mathrm{KL}}(\cdot \Vert \cdot)$	Kullback–Leibler divergence between probability measures	30
$W_{p,d}(\cdot, \cdot)$	Wasserstein distance of pth order over distance d	92
$\mathbb{N}$	Natural numbers, not including zero	
$\mathbb{Z}$	Integers	
$\mathbb{R}$	Real numbers	
$\mathbb{C}$	Complex numbers	
$\mathbb{R}^d$	Euclidean space of dimension d	
$\mathbb{R}^X$	Space of functions from X to $\mathbb{R}$	
$\boldsymbol{a}$	Euclidean vector (bold italic letters)	
$\mathbf{A}$	Matrix (bold upface letters)	
$\oplus$	Direct sum of vector spaces.	49
V^*	Dual space of a topological vector space.	50
$\mathcal{A}^*$	Adjoint of a bounded linear operator	50
$\Vert \cdot \Vert$	Norm of a vector in a Banach space	49
$\langle \cdot, \cdot \rangle$	Inner product between vectors in a Hilbert space	49
$\langle \cdot \mid \cdot \rangle$	Duality pairing between topological vector spaces	59
$L(V; W)$	Banach space of bounded linear operators	50
$C^0(X; \mathbb{R})$	Banach space of real-valued continuous functions	50
$L^p(X; \mathbb{R})$	Lebesgue space	50

Bayesian learning

$(\Theta, \mathit{\Theta})$	Measurable space representing quantity of interest	24
$(Y, \mathcal{Y})$	Measurable space representing the data	24
$A_{(\cdot)}$	Measurable set	24
θ, y	Random variables	24
$\pi_{(\cdot)}$	Probability measure	24
$f_{(\cdot)}$	Probability density	29
$(\cdot \mid \cdot)$	Jointly measurable stochastic process	25
$\pi_{(\cdot\|\cdot)}$	Probability kernel	25
$f_{(\cdot\|\cdot)}$	Conditional probability density	29

Markov decision processes

S	State space	34
A	Action space	34
$r(\cdot \mid \cdot, \cdot)$	Reward kernel	34
$p(\cdot \mid \cdot, \cdot)$	Transition kernel	34
π	Policy	35
$V^{(\pi)}$	Value function with respect to policy π	35
$R^{(\pi)}$	Regret of policy π	36

Multi-armed bandits

X	Space of arms	38
K	Number of arms	39
T	Maximum time	38
y	Observed stochastic rewards	38
$f(\cdot)$	Expected rewards	38
ε	Observation noise	38
R	Cumulative stochastic regret of a bandit algorithm	38
$\Delta(\cdot)$	Regret of a specific arm	40
n_t	Cumulative number of arm pulls up to time t	40
$f_t^+(\cdot)$	Upper confidence bound at time t	41
$\mu_t(\cdot)$	Empirical mean rewards at time t	40
$\sigma_t(\cdot)$	Width of confidence bounds at time t	41
c_t	Scaling constant for confidence bound width at time t	41

Gaussian processes

$\mathrm{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2	51
$\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$	53
$k(x, x')$	Positive semi-definite kernel	57
$\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}'}$	Kernel matrix with entries $k(x_i, x'_j)$	57
$\mathrm{GP}(\mu, k)$	Gaussian process with mean μ and covariance kernel k.	58
f	Prior Gaussian process	73
$\boldsymbol{\varepsilon}$	Likelihood noise	73
$\boldsymbol{\Sigma}$	Covariance of likelihood noise.	73
$(\boldsymbol{x}, \boldsymbol{\gamma})$	Data used to condition the model $y_i = f(x_i) + \varepsilon_i$	73
$f \mid \boldsymbol{y}$	Posterior Gaussian process conditioned on $\boldsymbol{y} = \boldsymbol{\gamma}$	73
n	Number of training data points	75
n_*	Number of evaluation locations	75
$\tilde{f}$	Approximate prior defined via a finite basis expansion	76
$\phi_i(\cdot)$	Basis function used by approximate prior	76
w_i	Random weight of basis function ϕ_i	76
ℓ	Number of basis functions used in approximate prior	76
$k(x_j, \cdot)$	Canonical basis function.	77
v_j	Random weight of canonical basis function $k(x_j, \cdot)$	77
H_k	Reproducing kernel Hilbert space with kernel k.	79
$\boldsymbol{\phi}$	Finite-dimensional approximate feature map.	80
m	Number of inducing points in sparse approximation	87

Manifolds and graphs

X	Manifold	104
TX	Tangent bundle	105
T^*X	Cotangent bundle	106
$C^\infty(X)$	Space of smooth real-valued functions	105
$\Gamma(TX)$	Space of smooth sections.	106
d	Dimension of X	104
f_*	Pushforward of a function f	105
$g(\cdot, \cdot)$	Metric tensor of a Riemannian manifold	106
Δ_g	Laplace–Beltrami operator induced by g	107
(λ_n, f_n)	Laplace–Beltrami eigenpair.	108
$\Phi(\Delta_g)$	Operator defined via functional calculus	108
$\mathcal{W}_g$	Gaussian white noise induced by g.	115

$\boldsymbol{\Delta}$	Graph Laplacian	122
$\boldsymbol{\mathcal{W}}$	Standard Gaussian	124
$\Phi(\boldsymbol{\Delta})$	Matrix defined via functional calculus	123
F	Frame over X	134
$\mathbf{P}_F(\cdot)$	Projection map induced by F	136

Chapter 1

Introduction

Learning from experience in order to change behavior is one of the defining abilities of biological systems, which differentiates them from other kinds of systems found in the world. Replicating the processes biological systems use to learn and adapt is a fundamental goal of science and technology. To this end, the development of mathematical formalisms rich enough to capture the notion of learning is one of the crowning achievements of statistics, machine learning, and artificial intelligence.

One such formalism is the *Bayesian* view of learning. The idea behind Bayesian learning is to represent the information known about the quantity of interest using probability. To do so, the relationship between the quantity of interest and the data is formalized as a joint probability distribution. This gives rise to a conditional probability distribution describing what was learned about the quantity of interest by observing the data.

Bayesian learning fits naturally within a theory of *decision*, which describes how an abstract decision system should select actions in pursuit of a goal. This is done by learning how different actions affect pursuit of the goal, and selecting optimal actions consistent with what was learned. By virtue of being probabilistic, such decision systems assess and propagate uncertainty, enabling them to balance what is already known with what could be learned by taking actions—a concept known as the *explore-exploit tradeoff*.

The performance of a decision system can be evaluated by examining how quickly its decisions improve and become optimal. A decision system's *regret* is the reduction in its quality of decisions by virtue of not knowing the quantity of interest in advance. In most non-trivial settings, one can show that some regret is inevitable: a decision-making system must make some degree of mistakes in order to learn. A decision system is considered *optimal* if its regret is within a constant factor of the best possible regret.

Decision systems with optimal or close-to-optimal regret require less data in order to solve their respective tasks, and are called *data-efficient*. Data-efficiency is a key concern in practical settings, where data-collection takes time and can be expensive. By virtue of resolving explore-exploit tradeoffs in a manner amenable to regret analysis, the Bayesian formalism gives broad tools for constructing data-efficient decision systems.

The key limitation of the Bayesian approach is that it often leads to computational problems which are intractable. Conditional distributions generally contain more information than actually needed to make optimal decisions, yet calculating them is largely unavoidable. Probabilistic decision systems are thus most attractive in settings where their strengths—including data-efficiency, solid technical foundations, and amenability to analysis—can shine, while computational costs are kept under control.

In my view, *Gaussian processes* are one such setting: they are powerful enough to model wide classes of unknown quantities of interest, yet their computational costs are generally polynomial. Better yet, Gaussian-process-based decision systems have demonstrated excellent performance in real-world scientific applications. Studying Gaussian processes is therefore a promising avenue towards improved understanding of Bayesian learning and Bayesian decision-making in pursuit of artificial intelligence.

The goals of this dissertation are twofold: (i) to make Gaussian processes easier to work with when used within larger decision systems, and (ii) to expand the set of settings where Gaussian processes can used, enabling construction of decision systems for applications not previously considered. Contributions toward (i) include path-wise conditioning techniques studied in Chapter 2, and contributions toward (ii) include non-Euclidean Gaussian processes studied in Chapter 3. Following these, Chapter 4 concludes.

To pursue these goals, it is critically important that all of the concepts described in the preceding paragraphs be made into rigorous mathematics, so that the ideas described in the sequel ultimately reduce to definitions and implications, and not metaphor or opinion. Together, we therefore begin by defining the key mathematical notions needed.

Contributions. The work presented in this thesis is published as a series of papers. My contributions to the individual works are primarily on the theoretical and methodological side, and are described below.

In Chapter 2, we present pathwise conditioning techniques: this work is published as Wilson et al. [122, 123]. My contributions include (i) development of the random-function-based formalism for describing pathwise conditioning of Gaussian processes, (ii) error analysis of basis-function-approximation-based pathwise sampling, (iii) re-interpretation of inducing point methods, and (iv) review of prior sampling methods. All of these were developed jointly with the other authors.

In Chapter 3, we present Matérn Gaussian processes in non-Euclidean settings: this work is published as Borovitskiy et al. [13, 12], Jaquier et al. [54], and Hutchinson et al. [53]. My contributions include (i) developing the differential-geometric and stochastic-partial-differential-equation-based formalisms for defining these processes, and (ii) describing computational techniques for working with these models in practice. All of these ideas were developed jointly with the other authors.

1.1. Bayesian learning

The first concept we develop in depth along our path towards a mathematically precise understanding of statistical decision-making is *Bayesian learning*—a mathematical formalism for reasoning about unknown quantities of interest on the basis of data. Bayesian learning is a *probabilistic* theory: a model specifies how the quantity of interest and the data depend on one another within a probability distribution. Learning entails calculating how the distribution of the quantity of interest changes upon observing the data.

One of the key strengths of Bayesian theory is that it applies in wide generality, owing to the substantial scope of probability theory. In particular, one can study learning of quantities of interest that are function-valued using observed data consisting of pointwise function evaluations. This, however, requires a non-elementary treatment, as one cannot rely solely on probability densities to define what conditional distributions are. We therefore begin by recalling the necessary mathematical formalism.

For an overview of Bayesian methods from a model-building perspective, see Gelman et al. [41], and for an overview from a mathematical perspective see Ghosal and van der Vaart [43] and Giné and Nickl [44].

1.1.1. Review of probability theory. We adopt the language of measure-theoretic probability, which we now describe. To ease presentation, we state the definitions together with useful ways of thinking about them. We use Kallenberg [59] as our standard reference, along with certain results from Bogachev [10, 11].

Measurable space

We say that *measurable space* is a pair $(Y, \mathcal{Y})$ consisting of a set Y and a σ-algebra $\mathcal{Y}$ over Y. A *σ-algebra* is a set of subsets of Y containing the space itself which is closed under countable unions, intersections, complements, and therefore set-theoretic monotone limits. These can be reinterpreted as Boolean logical operations, so $\mathcal{Y}$ can be thought of as the set of all true-false questions one can ask about elements of the set Y. These questions, then, are closed under *and/or/not* operations and monotone limits thereof.

Product of measurable spaces

Measurable subspace

Given two measurable spaces $(Y, \mathcal{Y})$ and $(\Theta, \varTheta)$, if we form the Cartesian product $Y \times \Theta$, then we can define the *product σ-algebra* $\mathcal{Y} \otimes \varTheta$ as the smallest σ-algebra containing all sets of the form $A_y \times A_\theta$ with $A_y \in \mathcal{Y}$ and $A_\theta \in \varTheta$. For a measurable subset $Y' \subseteq Y$, we can define $\mathcal{Y}' = \{A_y \cap Y' : A_y \in \mathcal{Y}\}$, which is also a σ-algebra, thereby making $(Y', \mathcal{Y}')$ into a measurable space. We call $\mathcal{Y}'$ the *subset σ-algebra.*

Borel σ-algebra

If Y is a topological space, the *Borel σ-algebra* $\mathcal{B}(Y)$ is defined as the smallest σ-algebra containing the open sets in the topology. Topological spaces, in turn, are sets equipped with additional structure enabling them to admit notions such as locality and convergence—we review these spaces in additional detail in Chapter 3.

Measurable function

A map $f : Y \to Y'$ between measurable spaces $(Y, \mathcal{Y})$ and $(Y', \mathcal{Y}')$ is said to be *measurable* if its preimage defines a map $f^{-1} : \mathcal{Y}' \to \mathcal{Y}$ between the respective σ-algebras. This intuitively means that true-false questions for the space Y' can be asked and answered relative to those in Y. On product spaces, a map $f : Y \to \Theta \times \Theta'$ is measurable if its components are measurable, but a map $f : Y \times Y' \to \Theta$ *need not* automatically be measurable if $f(\cdot, y') : Y \to \Theta$ and $f(y, \cdot) : Y' \to \Theta$ are measurable for all y and y'.

Probability measure

Probability space

A *probability measure* is a non-negative countably additive map $\pi_y : \mathcal{Y} \to \mathbb{R}$ satisfying $\pi_Y(Y) = 1$. This can be thought of as a map that takes a true/false question, and assigns a number indicating how close to true or false its answer is according to the measure—in this view, probability measures describe uncertainty. A *probability space* $(\Omega, \mathcal{F}, \mathbb{P})$ consists of a measurable space and probability measure. The set Ω can be thought of as a space of abstract random numbers, with a random number generator described by $\mathbb{P}$.

Random variable

Distribution

Given a probability space, we say that a *random variable* is a measurable map $y : \Omega \to Y$. A random variable, then, maps random numbers $\omega \in \Omega$ into the space Y. The *distribution* of a random variable is defined as the *pushforward measure* $\pi_y = y_* \mathbb{P}$ where y_* is defined as $(y_* \mathbb{P})(A_y) = \mathbb{P}(y^{-1}(A_y))$ for all $A_y \in \mathcal{Y}$. The probability of an event A_y, then, is determined by measuring the probability of random numbers under which A_y occurs—measurability guarantees this is possible.

Notation for random variables

In this work, we will generally *not* adopt the standard convention of suppressing ω-arguments of random variables from our notation, and will write such arguments explicitly in cases that other function arguments are also used. Though this makes expressions slightly denser, it also avoids ambiguity and presents the mathematics more precisely.

Expectation

Covariance

The *expectation* $\mathbb{E}(y)$ of a real-valued random variable y, if it exists, is defined as its integral with respect to $\mathbb{P}$. This can be thought of as the average value of y under the random numbers generated from $\mathbb{P}$, with averaging performed via the algebraic structure of the reals. This notion extends to vector-valued random variables. The *covariance* of two real-valued random variables, if it exists, is defined as $\operatorname{Cov}(y, y') = \mathbb{E}(yy') - \mathbb{E}(y)\,\mathbb{E}(y')$. From this, we define the covariance of a finite-dimensional random vector componentwise.

Equality of random variables

There are multiple senses in which we can say random variables are equal. We say that $y = y'$ *surely* if they are equal as mathematical functions. This notion is essentially never used, because it is exceedingly strong. We say that $y = y'$ *almost surely* if $\mathbb{P}(y \neq y') = 0$, in which case y and y' are equal as functions, except possibly on a set of probability zero. We say that $y = y'$ *in distribution* if $y_* \mathbb{P} = y'_* \mathbb{P}$, meaning their distributions are equal.

Existence of a random variable with given distribution

For a given probability space $(\Omega, \mathcal{F}, \mathbb{P})$, a random variable $y : \Omega \to Y$ with distribution π_y need not exist. However, if it does not, there always exists another probability space $(\Omega', \mathcal{F}', \mathbb{P}')$ and a measurable map $i : \Omega' \to \Omega$ such that $\mathbb{P} = i_* \mathbb{P}'$, and for which a π_y-distributed random variable $y : \Omega' \to Y$ *does* exist. We thus implicitly assume all probability spaces are large enough to ensure all random variables with prescribed distributions exist.

Probability kernel

We will be interested in probability measures extended to allow them to be parameterized by other quantities. A *probability kernel* is defined to be a map $\pi_{y|\theta} : \mathcal{Y} \times \Theta \to \mathbb{R}$ satisfying two conditions: (i) the map $\pi_{y|\theta}(\cdot, \vartheta) : \mathcal{Y} \to \mathbb{R}$ is a probability measure for all $\vartheta \in \Theta$, and (ii) the map $\pi_{y|\theta}(A_y, \cdot) : \Theta \to \mathbb{R}$ is measurable for all $A_y \in \mathcal{Y}$. In particular, this means that a probability kernel can be integrated against other measures.

Conditional distribution

Random variables can also be extended to parameterize them by other quantities. A $\mathcal{F} \otimes \mathit{\Theta}$-measurable map $y \mid \theta : \Omega \times \Theta \to Y$ is called a *jointly measurable stochastic process*—we largely eschew this terminology to emphasize the Bayesian formalism. In particular, unlike most presentations of stochastic processes, we *never* think of Θ as representing time. Define the *conditional distribution* of $y \mid \theta$ to be $(y \mid \theta)_* \mathbb{P}$, with the pushforward taken in the first argument—by Lemma 1.7, this is a probability kernel.

1.1.2. Bayes' Rule for probability measures. The key idea behind Bayesian learning is to formalize *learning* using the concept of *conditional probability.* This entails (i) building a model quantifying the relationship between the *quantity of interest* θ, and the *data* y, and (ii) quantifying what is learned using Bayes' Rule. We now begin to make these concepts precise in the language of measure-theoretic probability.

Bayesian model

Definition 1.1. *Let Y and Θ be sets. A* BAYESIAN MODEL *is a probability measure defined on $\Theta \times Y$.*

A model can be constructed in a number of different ways. The most common technique is to specify two components: (i) a probability distribution describing what is known about the quantity of interest external to the data, and (ii) how the data relates to the quantity of interest. These are called the *prior distribution* and *likelihood*, respectively. For example, a simple Gaussian model is given by

$$y_i \mid \theta \sim \mathrm{N}(\theta, 1) \qquad\qquad \theta \sim \mathrm{N}(0, 1). \tag{1.1}$$

In this model, θ is an unknown scalar assigned a standard normal prior, while $y_i \mid \theta$ is assigned a Gaussian likelihood centered at θ. Our first goal is to make this notation into proper mathematics.

Proposition 1.2. *A Bayesian model can be constructed in the following ways.*

1. *Using measures: integrate the following two components.*
   (a) PRIOR: *a probability measure $\pi_\theta : \mathit{\Theta} \to \mathbb{R}$.*
   (b) LIKELIHOOD: *a probability kernel $\pi_{y|\theta} : \mathcal{Y} \times \Theta \to \mathbb{R}$.*
2. *Using random variables: compose the following two components.*

(a) Prior: a random variable $\theta : \Omega \to \Theta$.

(b) Likelihood: a jointly measurable stochastic process $y \mid \theta : \Omega \times \Theta \to Y$.

Moreover, if one takes the pushforward of the composition of the obtained random variables with respect to $\mathbb{P} \times \mathbb{P}$, then the two constructions coincide.

Proof. For the former, define

$$\pi_{\theta,y}(A_\theta \times A_y) = \int_{A_\theta} \pi_{y|\theta}(A_y \mid \theta) \, \mathrm{d}\pi_\theta(\theta) \tag{1.2}$$

which extends to the full product σ-algebra by Lemma 1.9, giving the desired probability measure. For the latter, define the random variable

$$\gamma : \Omega \times \Omega \to \Theta \times Y \qquad \gamma : (\omega, \omega') \mapsto (\theta(\omega), (y \mid \theta)(\omega', \theta(\omega))) \tag{1.3}$$

and take $\pi_{\theta,y} = \gamma_*(\mathbb{P} \times \mathbb{P})$. We now prove these expressions coincide. Write

$$\begin{aligned}
\pi_{\theta,y}(A_\theta \times A_y) &= \int_{A_\theta} \pi_{y|\theta}(A_y \mid \theta) \, \mathrm{d}\pi_\theta(\theta) && (1.4)\\
&= \int_\Omega \mathbb{1}_{\theta(\omega) \in A_\theta} \pi_{y|\theta}(A_y \mid \theta(\omega)) \, \mathrm{d}\mathbb{P}(\omega) && (1.5)\\
&= \int_\Omega \mathbb{1}_{\theta(\omega) \in A_\theta} \int_\Omega \mathbb{1}_{(y|\theta)(\omega',\theta(\omega)) \in A_y} \, \mathrm{d}\mathbb{P}(\omega') \, \mathrm{d}\mathbb{P}(\omega) && (1.6)\\
&= \int_\Omega \int_\Omega \mathbb{1}_{(\theta(\omega),(y|\theta)(\omega',\theta(\omega))) \in A_\theta \times A_y} \, \mathrm{d}\mathbb{P}(\omega') \, \mathrm{d}\mathbb{P}(\omega) && (1.7)\\
&= (\mathbb{P} \times \mathbb{P})(\gamma \in A_\theta \times A_y) && (1.8)
\end{aligned}$$

which follows using Tonelli's Theorem and Lemma 1.9. ■

These two components should be interpreted as follows: θ describes what is known about the quantity of interest external to the data, and $y \mid \theta$ describes how the data y relates to the quantity of interest. More specifically, the likelihood describes how y would be distributed if θ was known and equal to the conditioned value.

The condition that the pushforward of the composition of the prior and likelihood needs to be taken with respect to the product measure $\mathbb{P} \times \mathbb{P}$ should be interpreted as a kind of *non-duplicate dependence* condition which ensures that $y \mid \theta$ depends on θ only through its second argument, and not through the abstract random numbers ω.

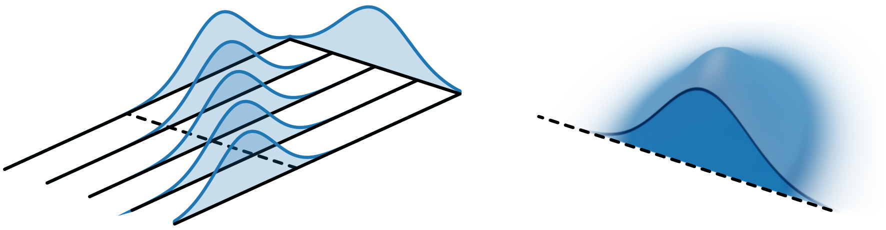


Figure 1.1. Here, we illustrate a Bayesian model. For all possible values of the quantity of interest, the likelihood describes the respective distribution of the data, shown on the left. This is combined with the prior, also shown on the left, to form a joint distribution, shown on the right. From this, the posterior distribution, also shown on the right, is obtained by conditioning. Note that the two plots are shown not to scale, and that the joint and posterior live on different spaces, hence are normalized differently.

Given a Bayesian model, we can formalize the notion of what is *learned* about θ from observing y as its conditional probability distribution—note that this is meant in a *distributional* sense, not a random-variable-based sense. The formulation we use requires topological assumptions: we say that a topological space is *Polish* if it is homeomorphic to a complete separable metric space. Note that any two uncountable Polish spaces are Borel-isomorphic: see Villani [117], Chapter 1, for discussion. The key result is given as follows.

Bayes' Rule

Result 1.3. *Suppose that Θ is a Polish topological space, and let $\pi_y = \pi_{\theta,y}(\Theta \times \cdot)$. Then for every Bayesian model $\pi_{\theta,y}$ there is a π_y-a.e. unique probability kernel $\pi_{\theta|y}$ satisfying*

$$\pi_{\theta,y}(A_\theta \times A_y) = \int_{A_y} \pi_{\theta|y}(A_\theta \mid y) \, \mathrm{d}\pi_y(y) \tag{1.9}$$

which we call the POSTERIOR DISTRIBUTION.

Proof. The claim follows directly from Bogachev [11], Corollary 10.4.15. See also Kallenberg [59], Theorem 5.3 and Theorem 5.4, Ambrosio et al. [3], Theorem 5.3.1, and Chang and Pollard [20] for variations of this result. ■

This result is illustrated in Figure 1.1, and shows that given a Bayesian model, the posterior distribution describing what was learned from the data exists. In its full abstract formulation, however, Bayes' Rule is *non-constructive*, and it is not at all clear how to calculate any kind of useful formula from it. Moreover, the null sets can be problematic: to ensure that $\pi_{\theta|y}$ is defined pointwise, one needs further properties such as continuity. Fortunately, in many settings these issues are resolved by virtue of additional structure.

The simplest such structure occurs when $\pi_{\theta,y}$ admits a density with respect to a product measure λ. In this case, it follows from Lemma 1.10 that π_θ admits the density f_θ with respect to $\lambda(\cdot \times Y)$, and similarly for π_y and f_y. Define a *conditional density* as the ratio of joint and marginal densities, for instance $f_{\theta|y} = \frac{f_{\theta,y}}{f_y}$, and let $\propto$ denote equality up to a multiplicative proportionality constant. Then, we have the following.

Bayes' Rule for densities

Proposition 1.4. *Suppose that $\pi_{\theta,y}$ admits the density $f_{\theta,y}$ with respect to $\lambda_{\theta,y}$. Then we have*

$$f_{\theta|y} \propto f_{y|\theta} f_\theta. \tag{1.10}$$

Proof. We have

$$f_{\theta,y} = \frac{f_{\theta,y}}{f_\theta} f_\theta = f_{y|\theta} f_\theta \qquad\qquad f_{\theta,y} = \frac{f_{\theta,y}}{f_y} f_y = f_{\theta|y} f_y \tag{1.11}$$

which, when combined, give the result. ■

Sometimes, the knowledge of $f_{\theta|y}$ up to proportionality is enough to fully deduce its form. For example, if the likelihood is Gaussian with unknown mean and known variance, and the prior is Gaussian, then posterior is also Gaussian. In cases such as this, where the prior and posterior land in the same parameterized class of distributions, the pair $(\pi_{y|\theta}, \pi_\theta)$ are called *conjugate*.

Conjugacy

Bayes' Rule for densities is remarkable in its generality yet restrictiveness. On the one hand, we require no direct assumptions about the spaces Θ and Y, and in particular allow for real spaces, discrete spaces, and Riemannian manifolds. On the other hand, other than in the aforementioned settings, where we can employ the Lebesgue, counting, and Riemannian volume measures, respectively, finding a suitable reference measure can be difficult.

We will work with posterior distributions in infinite-dimensional vector spaces in the sequel—there, densities are either not available or not convenient. In those cases, it is enough to calculate the posterior on arbitrary finite-dimensional marginal projections to uniquely determine its value on the full infinite-dimensional space.

Conditioning and marginalization

Proposition 1.5. *Conditioning and marginalization commute: given a probability measure $\pi_{\theta,\theta',y}$, if Bayes' Rule holds, then we have π_y-a.e. that*

$$\pi_{\theta|y} = \pi_{\theta,\theta'|y}(\cdot \times \Theta'). \tag{1.12}$$

Proof. Let $\pi_{\theta,\theta'|y}$ be the π_y-a.e. unique probability kernel given by the Disintegration Theorem satisfying

$$\pi_{\theta,\theta',y}(A_{\theta,\theta'} \times A_y) = \int_{A_y} \pi_{\theta,\theta'|y}(A_{\theta,\theta'} \mid y) \,\mathrm{d}\pi_y(y). \tag{1.13}$$

Plugging in $A_\theta \times \Theta'$ for $A_{\theta,\theta'}$ gives

$$\pi_{\theta,y}(A_\theta \times A_y) = \int_{A_y} \pi_{\theta,\theta'|y}(A_\theta \times \Theta' \mid y) \,\mathrm{d}\pi_y(y). \tag{1.14}$$

On the other hand, applying the Disintegration Theorem to $\pi_{\theta,y}$ gives a π_y-a.e. unique probability kernel satisfying

$$\pi_{\theta,y}(A_\theta \times A_y) = \int_{A_y} \pi_{\theta|y}(A_\theta \mid y) \,\mathrm{d}\pi_y(y). \tag{1.15}$$

Since both $\pi_{\theta,\theta'|y}$ and $\pi_{\theta,y}$ are defined with respect to the same null sets, we conclude by uniqueness that they coincide, and the claim follows. ■

This result makes densities into a substantially more powerful tool than they would be otherwise, since it enables us to use them for calculating posterior distributions even where they are not directly available. In particular, we can map an infinite-dimensional function space into finite-dimensional vector spaces induced by pointwise function evaluations at arbitrary points, enabling us to calculate posterior distributions in such settings, in spite of no suitable densities existing directly in the space of interest.

We now introduce the *variational formulation* of Bayes' Rule, which expresses the posterior as the solution to an infinite-dimensional optimization problem. Let $\mathcal{M}_1(\Theta)$ be the space of all probability measures over Θ.

Bayes' Rule in variational form

Proposition 1.6. *Let $\pi_{y,\theta}$ be a Bayesian model, and assume it admits the density $f_{\theta,y}$ with respect to a product measure λ. Then for every $\gamma \in Y$, the posterior distribution satisfies*

$$\pi_{\theta|y}(\cdot \mid \gamma) = \operatorname*{arg\,min}_{\mathbb{q}_\theta \in \mathcal{M}_1(\Theta)} D_{\mathrm{KL}}(\mathbb{q}_\theta \,||\, \pi_\theta) - \mathop{\mathbb{E}}_{\vartheta \sim \mathbb{q}_\theta} \ln f_{y|\theta}(\gamma \mid \vartheta) \tag{1.16}$$

where D_{KL} is the Kullback–Leibler divergence between probability measures, and the minima does not depend on the choice of λ.

Proof. Write the quantity being minimized as

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathbb{q}_\theta \mid\mid \pi_\theta) - \mathop{\mathbb{E}}_{\vartheta \sim \mathbb{q}_\theta} \ln f_{y\mid\theta}(\gamma \mid \vartheta) &= \mathop{\mathbb{E}}_{\vartheta \sim \mathbb{q}_\theta} \ln \frac{f_{\mathbb{q}}(\vartheta)}{f_\theta(\vartheta) f_{y\mid\theta}(\gamma \mid \vartheta)} && (1.17) \\
&= \mathop{\mathbb{E}}_{\vartheta \sim \mathbb{q}_\theta} \ln \frac{f_{\mathbb{q}}(\vartheta)}{f_{\theta\mid y}(\vartheta \mid \gamma) f_y(\gamma)} && (1.18) \\
&= D_{\mathrm{KL}}(\mathbb{q}_\theta \mid\mid \pi_{\theta\mid y}(\cdot \mid \gamma)) - \ln f_y(\gamma) && (1.19)
\end{aligned}
$$

Since $\ln f_y(\gamma)$ does not depend on $\mathbb{q}_\theta$, and since $D_{\mathrm{KL}}(\mu \mid\mid \nu) = 0$ implies $\mu = \nu$, we conclude that the objective is minimized by taking $\mathbb{q}_\theta = \pi_{\theta\mid y}(\cdot \mid \gamma)$. Since the minima does not depend on the choice of the measure λ with respect to which the densities are defined, the claim follows. ■

For any given dataset, this result shows that Bayes' Rule can be viewed in *information-theoretic* manner: among all probability measures, the posterior maximizes predictive power, while retaining as many bits as possible from the prior, in the sense given by the Kullback–Leibler divergence.

It's also possible to prove the result in a different way using the calculus of variations, with variations taken in the Banach space of signed measures under the total variation norm. That argument also employs Bayes' Rule for densities for its key step, making it largely similar in spirit.

The result suggests a way to approximately compute posterior distributions: restrict the optimization problem to a suitably chosen subspace of the space of all probability measures $\mathcal{M}_1(\Theta)$. This strategy will be particularly fruitful in the later-described setting of Gaussian processes, where techniques for constructing such approximations with well-understood and favorable accuracy will be considered.

We *always* view variational approximations as minimization of Kullback–Leibler divergences, as this is mathematically sound. We will eschew standard presentations involving *evidence lower bounds*: the only widely-known mathematical explanations for why maximizing these bounds should improve model performance appeal to Kullback–Leibler divergences—so, then, why talk about evidence lower bounds in the first place?

Once a posterior is calculated, the next step is to extract the relevant quantities from it. Traditionally, this is often done by displaying summary statistics to be evaluated and interpreted by a person with statistical training, often with a focus on assessing uncertainty. We will instead focus on settings where the posterior is given as input to an upstream decision-making algorithm: we explore these next.

1.1.3. Technical lemmas. We now prove a number of technical lemmas used in the preceding text, which for completeness are presented here in order to avoid disrupting the reader's flow.

Lemma 1.7. *Let $b : \Omega \times A \to B$ be a jointly measurable stochastic process. Then the map $b_* \mathbb{P} : \mathcal{B} \times A \to \mathbb{R}$, where the pushforward is taken in the first argument, is a probability kernel.*

Proof. It is clear that $b_* \mathbb{P}$ is a probability measure for all $a' \in A$, so we only need to prove that the map $a \mapsto (b(\cdot, a)_* \mathbb{P})(A_b)$ is measurable for all $A_b \in \mathcal{B}$. First, write

$$(b_* \mathbb{P})(A_b) = \mathbb{P}(b(\cdot, a)^{-1}(A_b)) = \int_\Omega \mathbb{1}_{b(\omega,a)\in A_b} \, \mathrm{d}\mathbb{P}(\omega). \tag{1.20}$$

Now, note that the map $\Omega \times A \to \mathbb{R}$ given by $(\omega, a) \mapsto \mathbb{1}_{b(\omega,a)\in A_b}$ is bounded measurable, since b is measurable in both arguments, and indicators of measurable functions on measurable sets are bounded measurable. Finally, since for any bounded measurable $f : \Omega \times A \to \mathbb{R}$, the map $a \mapsto \int_\Omega f(\omega, a) \, \mathrm{d}\mathbb{P}(\omega)$ is measurable by Fubini's Theorem, the claim follows. ■

Lemma 1.8. *Let $\mathcal{S}$ be a* SEMI-ALGEBRA OF SETS, *which is defined as a family of sets satisfying the following conditions.*

1. *$\mathcal{S}$ contains the empty set.*
2. *$\mathcal{S}$ is closed under finite intersections.*
3. *The complement of any set in $\mathcal{S}$ can be written as a finite union of disjoint sets in $\mathcal{S}$.*

Then any bounded countably additive non-negative function $\mu : \mathcal{S} \to \mathbb{R}$ satisfying $\mu(\varnothing) = 0$ extends uniquely to a measure defined on the smallest σ-algebra containing $\mathcal{S}$.

Proof. Define the *algebra of sets* generated by $\mathcal{S}$ to be

$$a(\mathcal{S}) = \left\{ \bigcup_{i=1}^{n} S_i : S_i \in \mathcal{S} \text{ disjoint} \right\} \tag{1.21}$$

and note that the smallest σ-algebra generated by $\mathcal{S}$ obviously coincides with the smallest σ-algebra generated by $a(\mathcal{S})$. We claim every function $\mu : \mathcal{S} \to \mathbb{R}$ satisfying the given assumptions extends uniquely to a function on $a(\mathcal{S})$. Every element of $a(\mathcal{S})$ can be written as a finite union of disjoint sets—using this, define

$$\mu^{(a)} : a(\mathcal{S}) \to \mathbb{R} \qquad\qquad \mu^{(a)}\left(\bigcup_{i=1}^{n} S_i\right) = \sum_{i=1}^{n} \mu(S_i). \tag{1.22}$$

To ensure this is well-defined, we check that the definition is independent of the choice of which disjoint subsets to take the union of—suppose that $\bigcup_{i=1}^n S_i = \bigcup_{j=1}^m T_j$. Then we have $S_i = \bigcup_{j=1}^m S_i \cap T_j$ and similarly $T_j = \bigcup_{i=1}^n S_i \cap T_j$: plugging these in to μ yields a double sum of disjoint sets and affirms well-definedness. Next, note that $\mu^{(a)}(\varnothing) = \mu(\varnothing) = 0$. To see that μ inherits countable additivity, take a sequence $S_1, S_2, .. \in a(\mathcal{S})$ and note that

$$\bigcup_{n=1}^{\infty} S_n = \bigcup_{n=1}^{\infty} \bigcup_{m=1}^{p_n} T_{nm} \tag{1.23}$$

where $T_{nm} \in \mathcal{S}$ and $p_n \in \mathbb{N}$. Plugging this into $\mu^{(a)}$ and applying countable additivity of μ gives the desired property. We have thus obtained a uniquely defined countably additive function $\mu^{(a)} : a(\mathcal{S}) \to \mathbb{R}$ defined on an algebra of sets $a(\mathcal{S})$ which extends μ. This function satisfies the assumptions of Carathéodory's Extension Theorem—see Kallenberg [59], Theorem A1.1 or Billingsley [8], Theorem 3.1—applying this result gives the claim, where we note that uniqueness follows since the range of $\mu^{(a)}$ is bounded. ■

Lemma 1.9. *Let $\pi : \mathcal{A} \times \mathcal{B} \to \mathbb{R}$ be a countably additive function with $\pi(\varnothing) = 0$. Then π extends uniquely to a measure on the product σ-algebra.*

Proof. By Lemma 1.8, it suffices to show that $\mathcal{A} \times \mathcal{B}$ is a semi-algebra of sets. The first required property is immediate, the second and third properties follow by $(A \times B) \cap (A' \times B') = (A \cap A') \times (B \cap B')$ and $(A \times B)^c = (A^c \times B) \cup (A \times B^c) \cup (A^c \times B^c)$. The claim follows. ■

Lemma 1.10. *Let $\pi_{a,b}$ be a measure which admits the density $f_{a,b}$ with respect to a product measure $\lambda_{a,b}$. Then $\pi_a = \pi_{a,b}(\cdot \times B)$ admits the density f_a with respect to $\lambda_a = \lambda_{a,b}(\cdot \times B)$, and similarly in the other argument.*

Proof. By assumption and Tonelli's Theorem, we have

$$\begin{aligned}
\pi_a(A_a) &= \pi_{a,b}(A_a \times B) && (1.24)\\
&= \int_{A_a \times B} f_{a,b}(\alpha, \beta) \,\mathrm{d}\lambda_{a,b}(\alpha, \beta) && (1.25)\\
&= \int_{A_a} \int_B f_{a,b}(\alpha, \beta) \,\mathrm{d}\lambda_b(\beta) \,\mathrm{d}\lambda_a(\alpha) && (1.26)\\
&= \int_{A_a} f_a(\alpha) \,\mathrm{d}\lambda_a(\alpha) && (1.27)
\end{aligned}$$

where the desired probability density is $f_a(\alpha) = \int_B f_{a,b}(\alpha, \beta) \,\mathrm{d}\lambda_b(\beta)$. ■

1.2. Statistical decision-making

We now use the Bayesian formalism to construct a probabilistic theory of decision-making. To begin, we formalize the very general concept of an abstract agent making decisions in an environment in pursuit of some goal. We use Sutton and Barto [110] and Bertsekas [7] as our references.

1.2.1. Markov decision processes. A Markov decision process is a stochastic system consisting of a set of states, actions, and transitions between states, together with a rewards that vary depending on states and actions. This is defined as follows.

Discrete-time Markov decision process

Definition 1.11. *A* DISCRETE-TIME MARKOV DECISION PROCESS *is a* 4*-tuple consisting of the following.*

1. *State space: a measurable space* S.
2. *Action space: a measurable space* A.
3. *Reward: a probability kernel* $r : \mathcal{B}(\mathbb{R}) \times S \times A \to \mathbb{R}$.
4. *Transition kernel: a probability kernel* $p : \mathcal{S} \times S \times A \to \mathbb{R}$.

This is a very broad notion—however, a number of variations are also possible. For instance, one can consider continuous-time, purely deterministic, and partially observed analogs—each of these involve their own subtleties and deserve study in their own right, but we do not pursue them here.

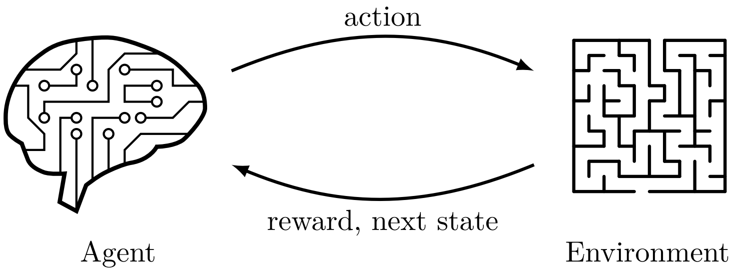


Figure 1.2. Illustration of the feedback loop induced by a Markov decision process. Here, the agent chooses an action, which results in the environment changing to a new state. The agent observes the new state, as well as the reward given by the previous state and action. The agent's goal is to select actions to maximize long-term rewards.

The idea behind this definition is that, at every point in time, the agent observes the current state, chooses an action $a \in A$, and obtains another state $s' \sim p(s \mid a)$. Figure 1.2 illustrates this. Note that *time* can, and often will, be part of the state s. The agent's goal is to choose actions so as to control the full trajectory of states in order to obtain maximum rewards. The choice of action in every state is called a *policy*, and is formalized as follows.

Definition 1.12. *Define the following.* Policy

1. *A measurable function $\pi : S \to A$ is called a* DETERMINISTIC POLICY.
2. *A probability kernel $\pi : \mathcal{A} \times S \to \mathbb{R}$ is called a* MARKOV POLICY.

Markov policies include deterministic policies as a special case, by taking the conditional distributions to be Dirac and re-interpreting the given expressions appropriately. As with Markov decision processes, one can also consider more general classes of policies, but we do not do so here.

Different policies yield different state trajectories, and therefore different rewards. Of these, some obtain more rewards than others. We therefore introduce notions for distinguishing between policies according to the rewards they obtain.

Definition 1.13. *Let $T \in \mathbb{N}$ be the* TIME HORIZON. *A policy is called* OPTIMAL Optimal policy *if it maximizes the* VALUE FUNCTION

$$V^{(\pi)}(s_0) = \mathbb{E} \sum_{t=0}^{T} r_t \tag{1.28}$$

where $r_t \mid s_t, a_t \sim r(s_t, a_t)$, $a_t \mid s_t \sim \pi(s_t)$, *and* $s_{t+1} \mid s_t, a_t \sim p(s_t, a_t)$.

Finding an optimal policy therefore amounts to selecting the best possible actions to maximize expected total rewards. Note that closely-related alternative notions of optimality, such as minimizing infinite discounted sums, or limits of averages, are also possible. The most important distinction between different decision problems for finding optimal policies is given by what is assumed known.

1. If p and r are known, we say we have an *optimal control* problem.
2. Otherwise, we say we have a *reinforcement learning* problem.

These classes differ fundamentally from one another. Optimal control problems can be viewed as a class of structured optimization problems, where the goal is to compute π by evaluating r and p as necessary. Here, one generally proceeds by proving that $V^{(\pi)}$ and π satisfy certain recursive equations, and developing schemes for solving them directly.

Reinforcement learning problems are more complex. Due to the rewards or dynamics being *unknown*, they cannot simply be maximized and their expectation must be *learned*. This forces one to consider whether to take advantage of actions known to be good, or to try others in case they might be better—this is known as the *explore-exploit tradeoff*. For such settings, we need an appropriate solution concept—to obtain one, define the following.

Regret

Definition 1.14. *The* REGRET *of a policy* π *is defined as*

$$R^{(\pi)}(s_0) = V^*(s_0) - V^{(\pi)}(s_0) \tag{1.29}$$

provided that the value function V^* *with respect to an optimal policy exists.*

Minimizing regret is equivalent to maximizing value, but when p and r are unknown doing so directly is impossible. Instead, the goal is to find an *algorithm*—that is, a way of updating the policy based on observed data—so as to limit growth of regret.

In most settings, one can prove that every algorithm which does not know p and r necessarily incurs some level of regret. This is done by exhibiting a randomized set of problems and rewards over which regret is lower-bounded in expectation for any algorithm. In such a class, actions that perform well on one problem will necessarily perform badly on another problem. Such arguments show that some degree of regret is inevitable.

On the other hand, some algorithms incur more regret than others. The obviously-bad algorithm which learns nothing and chooses the exact same action over and over again incurs at most linear regret. An algorithm is said to *solve* a decision problem if its asymptotic regret rate with respect to T matches the respective regret lower bound in the given problem class. Finding such algorithms is of key interest.

One way to construct algorithms for solving decision problems is to employ a *model-based* approach, which loosely speaking works as follows.

1. Learn the unknown transitions and/or rewards from observed data using a supervised learning approach.
2. Use the learned model(s) to find a policy satisfying some criteria.

The details of such approaches depend on the setting. We distinguish between two key kinds of reinforcement learning problems.

1. If $|S| = 1$, we say we have a *multi-armed bandit* problem. Multi-armed bandit
2. Otherwise, we say we have a general reinforcement learning problem.

Multi-armed bandits possess no variable state S, and thus only require one to learn the rewards to determine optimal actions. In general reinforcement learning, actions can influence transitions between states—these problems require long-term planning, making them much more general, difficult, and important. One can argue that as a mathematical theory, reinforcement learning is powerful enough to describe many aspects of human and animal intelligence, making it fundamentally interesting to study and develop.

We now restrict ourselves to the bandit setting, which is substantially easier to study and so can be understood more deeply. Here, even when A is a finite set and the rewards are Gaussian, the model-based approach consisting of (i) estimating rewards using empirical risk minimization and (ii) choosing the policy which maximizes rewards is known to be non-optimal. This approach fails to explore, and can get stuck chasing inferior rewards.

One way to fix this problem is to replace empirical risk minimization with

Bayesian learning, and adopt an appropriate decision rule for selecting actions. Doing this yields approaches which can be shown optimal in many settings. We therefore proceed to study multi-armed bandits in more detail.

1.2.2. Multi-armed bandits. The multi-armed bandit problem takes its name from a casino analogy. In the 1950s, slot machines often had levers one could pull instead of buttons one could press, and were called *one-armed bandits* for their ability to empty gamblers' wallets. A *multi-armed bandit* is a slot machine, which, for a fixed cost, allows one to pull an arm $x \in X$ and receive a random reward whose distribution depends on x. The goal is to minimize expected loss, or, equivalently, maximize total expected rewards.

Multi-armed bandits can be viewed as discrete-time Markov decision processes with a one-element state space, but this is not necessarily the most fruitful way to think about them. We thus begin by introducing formalism and notation better suited to the given setting, which can be viewed as a special case of the notions considered previously. We use Slivkins [98] and Lattimore and Szepesvári [64] as references.

Multi-armed bandit

Definition 1.15. *Let $f : X \to \mathbb{R}$ be a bounded above measurable function, let $\varepsilon : \Omega \times X \to \mathbb{R}$ be a jointly measurable stochastic process such that $\mathbb{E}(\varepsilon(x)) = 0$ for all x, and let $y(\omega, x) = f(x) + \varepsilon(\omega, x)$. Define the following.*

1. *We say that the Markov decision process $(\{1\}, X, y_* \mathbb{P}, \delta_1)$, fully defined below, is a* MULTI-ARMED BANDIT.
2. *We say that a probability kernel $\pi : \mathcal{X} \times \bigoplus_{n=1}^{\infty}(X \times \mathbb{R})^n \to \mathbb{R}$ is a* MULTI-ARMED BANDIT ALGORITHM.

Here, δ_1 is the Dirac measure centered at 1 for all actions $x \in X$, which is the only possible conditional probability distribution over a one-element set, and $\bigoplus$ denotes the disjoint union of measurable spaces.

An algorithm, then, assigns every dataset of arbitrary size to a probability measure describing what arms should be picked with what probability. Each dataset consists of (x, y) pairs where x are the locations chosen by the algorithm, and y are the noisy observed values—recall ω is the randomness used by the noise. Some algorithms maximize f faster than others—we consider this next.

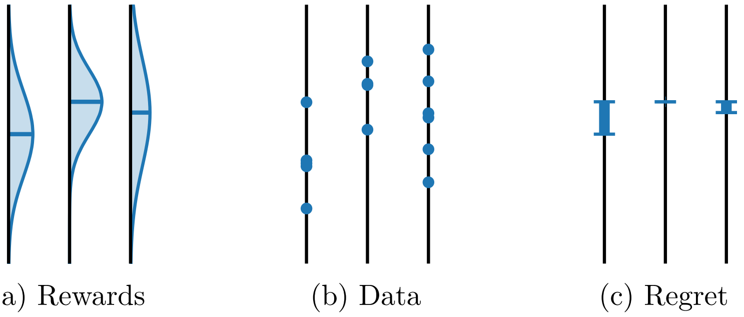

(a) Rewards (b) Data (c) Regret

Figure 1.3. Here, we illustrate a simple three-armed bandit. Each of the three arms has its own reward distribution, shown on the left. These are unknown to the agent, which only sees the rewards obtained by actions it takes, shown in the center, and must use the obtained information to decide which arm to pick. Each arm's regret is given by its expected decrease in reward compared to the optimal arm, shown on the right.

Definition 1.16. *For a given multi-armed bandit, let $f(x^*)$ be the global maxima of f, which we assume to exist. Define the* REGRET *of an algorithm π at time T to be* Regret

$$R(\omega, T) = \sum_{t=1}^{T} f(x^*) - f(x_t(\omega)) \tag{1.30}$$

where $x_t \sim \pi(x_1, y_1, .., x_{t-1}, y_{t-1})$, $y_t(\omega) = f(x_t) + \varepsilon_t(\omega)$, and $\varepsilon_t \sim \varepsilon(\cdot, x_t)$.

Regret thus counts the total reward lost by virtue of not knowing the optimal arm in advance—this is illustrated in Figure 1.3. Algorithms which achieve small regret therefore learn the optimal rewards effectively, and avoid getting stuck pulling bad arms.

Regret behavior in multi-armed bandit problems is strongly dependent on properties of the underlying function f, and in particular its domain X. It is clear by considering for instance $X = \mathbb{R}$ that if X is too large or too unstructured, no algorithm achieves better than linear regret. We therefore begin studying the simplest non-trivial domain class.

Definition 1.17. *We say that a multi-armed bandit defined over a finite set with cardinality $K = |X|$ is a K-*ARMED BANDIT*. Moreover, if $y(\cdot, x)$ is a Bernoulli random variable for all x, we say it is a* BERNOULLI BANDIT. K-armed bandit

For such bandits, regret can be decomposed on a per-arm basis in the manner given as follows.

Lemma 1.18. *Regret satisfies*

$$R(\omega, T) = \sum_{x=1}^{K} \Delta(x) n_T(\omega, x) \tag{1.31}$$

where $n_T(\omega, x) = \sum_{t=1}^{T} \mathbb{1}_{x_t(\omega)=x}$ *and* $\Delta(x) = f(x^*) - f(x)$.

Proof. The claim follows directly from the definitions of R and n_T by grouping terms in the sum. ■

What kind of performance is possible on such a problem? One can ask and answer this question as follows.

Theorem 1.19. *For any algorithm defined over a class of K-armed bandits there is an f such that*

$$\mathbb{E}(R(\cdot, T)) \geq \Omega(\sqrt{KT}). \tag{1.32}$$

Proof. Slivkins [98], Theorem 2.11. ■

A similar lower bound, but where the right-hand-side depends on an extra term controlling the difficult of the problem, was originally proved by Lai and Robbins [62]. Such bounds are called *instance-dependent*: in that case, the rate obtained is logarithmic, owing to the presence of the extra term. In our analysis, we focus on *instance-independent* bounds such as the one presented above.

This result tells us that regret is necessarily incurred as consequence of not knowing the expected rewards of each arm given by f. The next step, then, is to ask: is there an algorithm which achieves this rate? We first consider simply evaluating each arm once, and then making choices according to the empirical averages.

Proposition 1.20. *For a Bernoulli bandit, the algorithm which chooses actions by first trying each arm out once, and then selecting arms according to the* *maximum empirical average*

$$x_{t+1} = \operatorname*{arg\,max}_{x \in X} \mu_t(x) \qquad \mu_t(x) = \frac{\sum_{t=1}^{T} \mathbb{1}_{y_t=1} \mathbb{1}_{x_t=x}}{\sum_{t=1}^{T} \mathbb{1}_{x_t=x}} \tag{1.33}$$

of the random observed data achieves linear regret.

Proof. We first show that it suffices to exhibit a bandit along with an event S, which we call the *stuck event*, which occurs with constant probability and induces linear regret. To see this, for an event S, write

$$\begin{aligned} \mathbb{E}(R(\cdot,T)) &= \mathbb{E}(R(\cdot,T) \mid S)\,\mathbb{P}(S) + \mathbb{E}(R(\cdot,T) \mid S^c)\,\mathbb{P}(S^c) && (1.34) \\ &\geq \mathbb{E}(R(\cdot,T) \mid S)\,\mathbb{P}(S). && (1.35) \end{aligned}$$

Now, take S to be the event that during the initial first arm pulls, the optimal arm yields zero reward, while some other arm yields a reward: clearly $\mathbb{P}(S) > 0$. On the other hand, $\mathbb{E}(R(\cdot,T) \mid S) = \mathcal{O}(T)$, because the optimal arm conditional on S has empirical mean zero, which is always smaller than some alternative, and will never be selected again. The claim follows. ■

This algorithm fails to resolve the explore-exploit tradeoff, and gets stuck with suboptimal choices. While the setup is obviously contrived—for instance, giving each arm some non-zero selection probability even if it previously gave bad rewards would seemingly break the lower bound—it also illustrates an important principle: the algorithm must explore. The question, then, is how should it explore? As an alternative, consider a simple uncertainty-based decision rule for selecting arms.

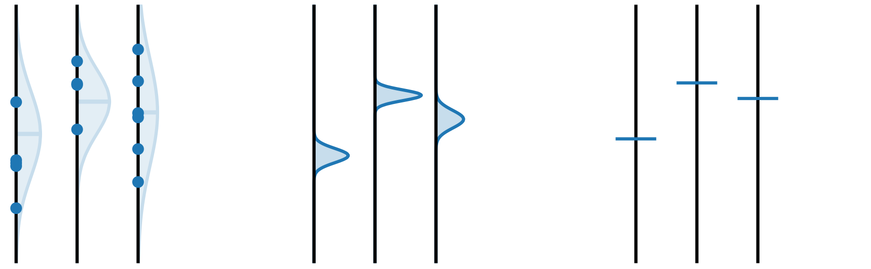
Definition 1.21. *Define the* Hoeffding upper confidence bound *algorithm for Bernoulli bandits, which tries each arm once, and then, for a total of T rounds, selects actions by*

Hoeffding upper confidence bound algorithm

$$x_{t+1} = \operatorname*{arg\,max}_{x \in X} f_t^+(x) \qquad f_t^+(x) = \mu_t(x) + c_t \sigma_t(x) \tag{1.36}$$

where μ_t are the empirical means, $\sigma_t(x) = \sqrt{\frac{1}{n_t(x)}}$, and $c_t = \sqrt{2\ln(T)}$.

Many algorithms within the upper confidence bound class have been proposed [62, 2, 4]: we focus on the variant analyzed by Auer et al. [4], but illustrate an alternative in Figure 1.4. We study the Hoeffding-based algorithm's regret,



(a) Data (b) Posterior (c) Upper confidence bound

Figure 1.4. The idea behind the upper confidence bound algorithm is to use the observed data to construct a set of error bars that reflect what has been learned about the mean rewards. Here, we construct these using the posterior distribution under a Gaussian prior and likelihood. The next arm is chosen as the maximum of the quantile-based error bars. This process is repeated iteratively as additional data is obtained, with the quantile becoming more strict over time, ensuring the bounds hold increasingly often.

and present a simplified but less sharp analysis compared to the one given in that work. It turns out the simple modification of adding a set of error bars, with a threshold growing in time and decreasing with data size, is enough to result in favorable regret behavior.

Theorem 1.22. *The Hoeffding upper confidence bound algorithm achieves an expected regret of*

$$\mathbb{E}(R(\cdot,T)) \leq \widetilde{\mathcal{O}}(\sqrt{KT}) \tag{1.37}$$

uniformly for all f, where $\widetilde{\mathcal{O}}$ denotes asymptotics up to logarithmic factors.

Proof. We adapt the argument presented by Slivkins [98]. First, let B denote the event that the *bound holds*, namely $f_t^-(\cdot,x) \leq f(x) \leq f_t^+(\cdot,x)$ for all $x \in X$ and all $t \leq T$, where $f_t^-(\cdot,x) = \mu_t(\cdot,x) - c_t\sigma_t(\cdot,x)$, where all stochasticity is defined with respect to the data. Under this event, the upper confidence bound is a true bound on the rewards. Thus, we have

$$\begin{aligned}\mathbb{E}(R(\cdot,T)) &= \mathbb{E}(R(\cdot,T) \mid B)\,\mathbb{P}(B) + \mathbb{E}(R(\cdot,T) \mid B^c)\,\mathbb{P}(B^c) && (1.38)\\ &\leq \mathbb{E}(R(\cdot,T) \mid B) + T\,\mathbb{P}(B^c) && (1.39)\end{aligned}$$

since $\mathbb{P}(B) \leq 1$, and a regret of T is the maximum possible value as there are T rounds and rewards are in $\{0,1\}$. Our strategy will be to bound $\mathbb{E}(R(\cdot,T) \mid B)$, while choosing the scaling factor c_t in the width of the upper confidence bound to ensure that $\mathbb{P}(B^c)$ decays fast enough that the latter term is negligible. We begin with the latter. By the union bound, we have

$$\mathbb{P}(B^c) \leq \sum_{t=1}^{T} \sum_{x \in X} \mathbb{P}(p_x \leq f_t^-(\cdot, x)) + \mathbb{P}(p_x \geq f_t^+(\cdot, x)) \tag{1.40}$$

$$\leq 2KT \max_{\substack{x \in X \\ 1 \leq t \leq T}} \mathbb{P}(p_x \geq f_t^+(\cdot, x)) \tag{1.41}$$

where p_x is the Bernoulli parameter for arm x, and we have used symmetry of the error bars to combine terms. For each $n > 0$, if y is a sum of independent Bernoulli random variables, the probability we want to bound is

$$\mathbb{P}\left(p_x \geq \frac{y(\cdot)}{n} + \sqrt{\frac{2\ln(T)}{n}}\right) \leq \exp\left(-2\left(\sqrt{\frac{2\ln(T)}{n}}\right)^2 n\right) = \frac{1}{T^4} \tag{1.42}$$

using Hoeffding's inequality. Since holds uniformly in n, it follows that

$$\mathbb{P}(p_x \geq f_t^+(\cdot, x)) = \mathbb{P}\left(p_x \geq \frac{\sum_{x_t = x} y_t(\cdot)}{n_t(\cdot, x)} + \sqrt{\frac{2\ln(T)}{n_t(\cdot, x)}}\right) \leq \frac{1}{T^4}. \tag{1.43}$$

We therefore conclude that

$$T\,\mathbb{P}(B^c) \leq \frac{2KT}{T^4} = \mathcal{O}(1) \tag{1.44}$$

which shows that the upper confidence bounds are exceeded sufficiently rarely that this possibility incurs no regret in the asymptotic limit, and completes this part of the argument. We now proceed to analyze the $\mathbb{E}(R(\cdot, T) \mid B)$ term—assume henceforth that B holds. We have that

$$\Delta(x_t(\omega)) = f(x^*) - f(x_t(\omega)) \tag{1.45}$$

$$\leq f_t^+(\omega, x^*) - f_t^-(\omega, x_t(\omega)) \tag{1.46}$$

$$\leq f_t^+(\omega, x_t(\omega)) - f_t^-(\omega, x_t(\omega)) \tag{1.47}$$

$$= 2\sigma_t(\omega, x_t(\omega)) \tag{1.48}$$

using the definition of the event B. Now, if we consider the rescaled error bar width $c_t\sigma_t(\omega, x_t(\omega))$ of the arm chosen at time t, this is

$$\Delta(x_t(\omega)) \leq \mathcal{O}(c_t\sigma_t(\omega, x_t(\omega))) = \widetilde{\mathcal{O}}\left(\frac{1}{\sqrt{n_t}}\right). \tag{1.49}$$

By Lemma 1.18—which we note still holds conditional on B via the exact same argument—we have

$$
\begin{aligned}
\mathbb{E}(R(\cdot,T) \mid B) &= \sum_{x=1}^{K} \Delta(x) n_T(\cdot,x) && (1.50)\\
&\leq \sum_{x=1}^{K} \widetilde{\mathcal{O}}\left(\sqrt{n_T(x)}\right) && (1.51)\\
&\leq \widetilde{\mathcal{O}}\left(\sqrt{K}\sqrt{\sum_{x\in X} n_T(x)}\right) && (1.52)\\
&= \widetilde{\mathcal{O}}(\sqrt{KT}) && (1.53)
\end{aligned}
$$

using that by definition, every x for which $n_T(\omega,x) > 0$ is equal to $x_t(\omega)$ for some t, enabling us to apply the preceding inequality. The claim follows. ■

This shows that the proposed algorithm, which uncertainty built via the error bars, effectively balances exploration and exploitation in this setting. The above behavior is not unique to the given form of the upper confidence bound, nor to the upper confidence bound rule itself. For example, we can consider a variation of the above algorithm, where the width of the error bars is constructed via a Bayesian model.

Beta–Bernoulli upper confidence bound algorithm

Definition 1.23. *Define a Bayesian model via the likelihood $\gamma(x) \sim \operatorname{Ber}(\mu(x))$ and prior $\mu(x) \sim \operatorname{Beta}(a,b)$. Define the* BETA–BERNOULLI UPPER CONFIDENCE BOUND *algorithm which selects actions by maximizing the function*

$$
x_{t+1} = \operatorname*{arg\,max}_{x\in X} f_t^+(x) \qquad f_t^+(x) = \mu_t(x) + c_t\sigma_t(x) \tag{1.54}
$$

where (μ_t,σ_t) are the mean and standard deviation of the posterior distribution $\mu \mid \gamma(x_1) = y_1, .., \gamma(x_t) = y_t$, and c_t is defined as previously.

One can show this algorithm, illustrated in Figure 1.4, achieves the same regret as its Hoeffding-based analogue: the proof is similar, but instead uses the empirical Bernstein inequality, and is more messy owing to the complicated form of the upper confidence bound. Even more generally, one can select arms by optimizing a function $\alpha : X \to \mathbb{R}$, called an *acquisition function*, built from a posterior distribution, confidence set, or other appropriate construction. Many different acquisition functions have been proposed.

We have made no attempt to optimize the bound, and significant improvements are possible, particularly when considering variations of the upper confidence bound algorithm—see Lattimore and Szepesvári [64] for state-of-the-art results. Of these, algorithms built using *confidence sets* rather than posterior distributions, such as the variant presented previously, are particularly important in the K-armed bandit setting. We focus on the Bayesian view because it generalizes well to more complex settings.

We now proceed to explore a more general setting which will enable us to employ the ideas developed so far to develop efficient black-box optimization algorithms. This will enable us to use Bayesian methods to solve a broad class of decision problems of practical interest.

1.2.3. Bayesian optimization. We now describe the formalism of *Bayesian optimization* for using ideas built on Bayesian learning and multi-armed bandits to design global optimization algorithms. We use Frazier [39] as our main reference. Our goal now is to minimize a black-box function

$$f : X \to \mathbb{R} \tag{1.55}$$

which is assumed continuous, and defined on a compact set $X \subseteq \mathbb{R}^d$. Such a function is automatically bounded. Our goal is to minimize f with as few evaluations as possible. To measure performance, we again introduce a notion of regret. One choice is to define

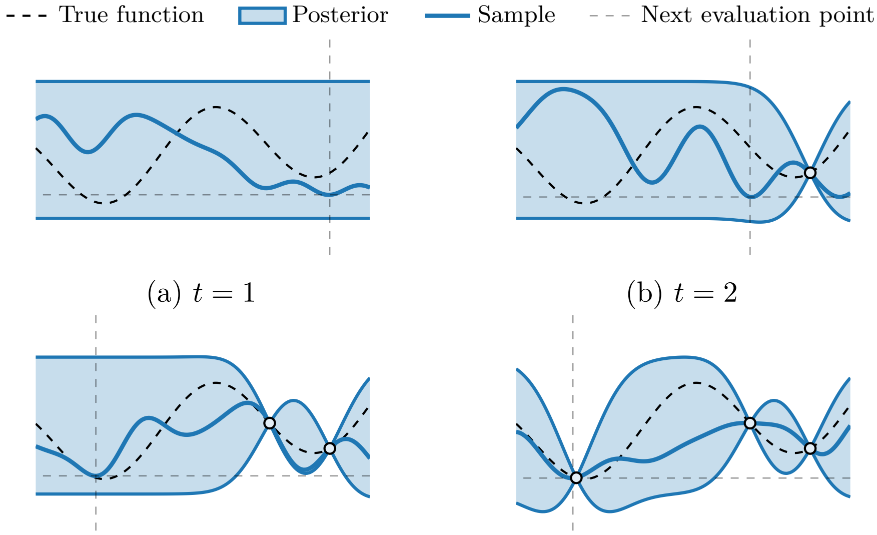
$$R(T) = \sum_{t=1}^{T} f(x_t) - f(x^*) \tag{1.56}$$

where we have used the opposite sign convention compared to bandits and reinforcement learning, because our goal is to minimize f rather than maximizing rewards. Note that unlike before, for a deterministic algorithm this is now a purely deterministic quantity, and not a random variable.

As before, we can approach this problem by building a Bayesian model. For an arbitrary sequence of points $x_1, .., x_t$, define the likelihood

$$y_t(\omega) = f(x_t) + \varepsilon_t(\omega) \qquad\qquad \varepsilon_t \sim \mathrm{N}(0, \sigma^2). \tag{1.57}$$

Here, the space of observed values is $\mathbb{R}$, and our quantity of interest is the actual *function* f, which we at least implicitly view as an element of a space of functions, such as for instance the Banach space of continuous functions $C^0(X; \mathbb{R})$.



(a) $t = 1$

(b) $t = 2$

(c) $t = 3$

(d) $t = 4$

Figure 1.5. Bayesian optimization using Thompson sampling. Here, we sample a random function from the posterior distribution over possible functions, and choose the next evaluation point to be the minima of the sampled random function. As additional data is obtained, the posterior distribution concentrates around the true function. Unlike the upper confidence bound acquisition function considered previously, Thompson sampling makes use of a full probability distribution, rather than just a set of error bars.

It is now clear why we bothered with setting up a dense, abstract formalism for Bayesian learning in general measure spaces: this formalism is rich and powerful enough to enable us to properly define priors on spaces like this—we explore these in the sequel. Suppose for the moment that this is possible. Then, we can calculate the posterior distribution

$$f \mid y_1, .., y_t \tag{1.58}$$

which is now a probability measure supported on $C^0(X; \mathbb{R})$. To determine the next point to query, we introduce and optimize an acquisition function. Typical acquisition functions include the *upper confidence bound* [4] acquisition function considered previously, as well as *Thompson sampling* [113, 93], first proposed half a century before Bayesian optimization was otherwise popularized. This is defined as a random acquisition function given by

$$x_{t+1}(\omega) = \operatorname*{arg\,min}_{x \in X} \alpha_t(\omega, x) \qquad\qquad \alpha_t \sim f \mid y_1, .., y_t. \tag{1.59}$$

We show an example of Bayesian optimization using Thompson sampling in Figure 1.5. Another typical choice is given by *expected improvement* [78, 57, 101], which is defined as

$$x_{t+1} = \operatorname*{arg\,max}_{x \in X} \mathbb{E} \max(0, f_t(\cdot, x) - f(x_t^*)) \tag{1.60}$$

where $f_t = f \mid y_1, .., y_t$ is the posterior, and $x_t^* = \arg\min_{t \in \{1,..,t\}} f(x_t)$ is location with the smallest value observed so far. Many other effective acquisition functions, including *probability of improvement* [61], *predictive entropy search* [51] and *information directed sampling* [92], are also possible.

Regret analysis for all of these choices can be performed, and will in general depend on the detailed properties of the Bayesian model, acquisition function, and unknown function f [106]. In particular, regularity and smoothness properties may play a role, as well as structure present in the domain X. In settings where the function f is unknown, it's also possible to analyze

$$BR(T) = \mathbb{E} \sum_{t=1}^{T} f(\omega, x_t) - f(\omega, x^*(\omega)) \tag{1.61}$$

where regret is now considered in expectation with respect to the prior—this is called the *Bayesian regret.* Some acquisition functions, such as Thompson sampling, admit particularly simple analyses in this setting [64].

This concludes our development and showcasing of decision-making algorithms. We now proceed to develop the final piece of the puzzle not yet studied in detail: how to place priors on function spaces, in order to build Bayesian models for settings such as Bayesian optimization.

1.3. Gaussian processes

In the preceding section, considerations arising from Bayesian decision-making algorithms motivated us to find a way to place priors on spaces of functions $f : X \to \mathbb{R}$. Gaussian processes are a broad class of random functions capable of this. In Figure 1.5, to perform Bayesian optimization using Thompson sampling, we used a Gaussian process model to define the posterior distribution over random functions, as needed by the decision rule.

To develop Gaussian processes, we will need to start with simpler notions and gradually work towards increasing levels of generality. Gaussian processes are defined by the key property that no matter what angle one views them from, they yield Gaussian marginal distributions. We therefore begin by studying properties of Gaussian random variables and Gaussian random vectors, which will be used as the basic building blocks for the general case.

The notion of a Gaussian process as a random function will turn out to be too restrictive for all of our settings of interest: it is not possible to view certain random variables, which do deserve to be called Gaussian processes, in this way. The obstructions can be geometric or analytic in nature. For example, a vector field on a manifold is not a vector-valued continuous function, it is a section of a vector bundle: what, then, should the term *Gaussian* actually mean? Difficulties also occur when considering white noise processes.

To handle these issues, we work with the notion of a Gaussian process $f : \Omega \to V$ where V is a real vector space equipped with additional structure arising from the setting at hand. Gaussian process are fundamentally *linear* objects which reflect this structure. The simplest possible settings one can study arise when V is smallest.

1. Choosing $V = \{0\}$ to be the trivial vector space, there is exactly one V-valued random variable, whose distribution is the Dirac measure centered at 0, which can trivially be called Gaussian. This random variable is not very interesting, so we do not consider it further.

2. Choosing $V = \mathbb{R}$ to coincide with the underlying scalar field yields the setting of *Gaussian random variables*. This is the next-simplest setting and the first one we explore in detail.

3. Choosing $V = \mathbb{R}^d$ yields the setting of *Gaussian random vectors*, whose components are multivariate Gaussian—or, equivalently, whose *dot products* are all scalar-valued Gaussian.

4. Choosing V to be a space of functions $f : X \to \mathbb{R}$ yields the setting of Gaussian processes, whose finite-dimensional marginals are multivariate Gaussian. This setting is well-studied when X is itself a Euclidean space: in Chapter 3, we will examine cases where X instead possesses various kinds of geometric structure.

5. Finally, choosing V to be a possibly infinite-dimensional topological vector space yields the most general setting we examine. This level of generality will be important for two reasons: (i) to provide a formalism for studying stochastic partial differential equations, and (ii) to develop a coordinate-free notion of Gaussian random vectors that will be useful in differential-geometric settings.

In what follows, our goal will be to lay the groundwork for subsequent development. Thus, we will not discuss Bayesian learning with Gaussian processes, which will instead be presented in Chapter 2. We use Lifshits [71] as our standard reference. We first study the scalar case, before generalizing to other settings.

1.3.1. Review of vector spaces. Since Gaussian processes are closely connected to vector spaces and related mathematical structures, here we review ideas from functional analysis that are useful in understanding them. We use Lang [63] as our reference on these topics.

Topological vector space

A set equipped with a notion of addition and multiplication by scalars taking values in an underlying field is called a *vector space*. We work with vector spaces where the underlying field is the real numbers. A *topological vector space* is a vector space equipped with a *topology* under which the vector operations are continuous—the topology gives rise to notions of *convergence*, *continuity*, *compactness*, and many others. Many interesting spaces of functions and generalizations thereof are topological vector spaces.

Banach space

Hilbert space

A *Banach space* is a topological vector space equipped with a *norm*, which assigns a notion of size to all vectors, and induces a notion of distance given by a topologically complete metric, which in turn induces the space's topology. A *Hilbert space*, similarly, is a topological vector space equipped with an *inner product* which induces a complete norm. Many, but not all, useful spaces of functions are Banach or Hilbert spaces. Similarly, many, but not all, useful notions of convergence come from norms or inner products.

Basis

Dimension

Every vector space admits a *Hamel basis*, which allows arbitrary vectors to be decomposed into linear combinations of linearly independent vectors in the basis. Bases are generally highly non-unique. The *dimension* of a vector space is the cardinality of such a basis, which does not depend on which basis is chosen. Hamel bases of infinite-dimensional vector spaces are usually poorly-behaved, and alternative notions, such as that of a *Schauder basis* of a Banach space, are often used instead.

Direct sum

Given two vector spaces V and W, we can form the *direct sum* vector space $V \oplus W$ by taking the Cartesian product of both spaces, and defining vector operations for $V \oplus W$ componentwise using the vector operations on V and W. This yields a vector space whose dimension is the sum of the dimensions of V and W. If V and W are topological vector spaces, equipping $V \oplus W$ with the product topology makes it into a topological vector space. In this work, we only consider finite direct sums.

Linear operator

Maps between vector spaces are called *linear operators* if they preserve the vector operations. By linearity, continuous maps between Banach spaces are automatically bounded with respect to their inputs, and vice versa, so in this context we refer to boundedness and continuity interchangeably. Given two respective bases, linear maps between finite-dimensional vector spaces can be viewed as matrices via their action on a vector's basis coefficients. In doing so, composition of linear maps becomes matrix multiplication.

Dual space

Functional

For a topological vector space V, the space V^* consisting of continuous linear maps $\phi : V \to \mathbb{R}$ into the underlying scalar field is called its *dual space*. The dual space can be made into a vector space: for many topological vector spaces, including Banach and Hilbert spaces, it can canonically be assigned a topology making it into a topological vector space. We generally call elements of V^* *linear functionals*.

Space of bounded linear operators

Operator norm

Adjoint

The spaces of linear operators between two vector spaces can also be given the structure of a vector space. Using this, if V and W are Banach spaces, one can define the space of $L(V;W)$ of bounded linear operators. This space can be made into a Banach space via the *operator norm*, defined as the largest possible size of a unit-norm vector in V after it is mapped into W by the given operator. The *adjoint* $\mathcal{A}^* : H \to G$ of a bounded linear operator $\mathcal{A} : G \to H$ between Hilbert spaces is defined by $\langle \mathcal{A}g, h \rangle_H = \langle g, \mathcal{A}^* h \rangle_G$.

Space of continuous functions

We now describe some particularly important function spaces. The space of *continuous functions* is denoted $C^0(X;\mathbb{R})$: if X is a compact set, this space can be equipped with the supremum norm to obtain a Banach space. This is not the only useful topology on this space: in many settings, for instance, the topology induced by convergence of all bounded linear functionals is also used. The space of infinitely differentiable functions is denoted by $C^\infty(X;\mathbb{R})$: this space, in contrast, is generally not a Banach space.

Lebesgue space

For a measure space X, the *Lebesgue space* $L^p(X;\mathbb{R})$, defined for $1 \leq p < \infty$, is the Banach space of equivalence classes of measurable functions $f : X \to \mathbb{R}$, identified up to almost sure equality, whose norm, given by integration of absolute pth powers, is finite. Of these spaces, only $L^2(X;\mathbb{R})$ is a Hilbert space. The space $L^\infty(X;\mathbb{R})$ is the Banach space of essentially bounded equivalence classes of functions equipped with the essential supremum norm.

Space of distributions

Many useful topological vector spaces are not spaces of functions. The *space of distributions* $D'(X)$ is defined as the dual of the space $D(X)$ of infinitely differentiable functions with compact support, equipped with the topology of uniform convergence of the function and all derivatives on compact sets.

Locally integrable functions embed into $D'(X)$, so we can view it as a space of generalized functions, containing both functions and other, less regular elements such as the Dirac delta function, which is not a classical function.

1.3.2. Gaussian random variables. A Gaussian random variable is a map which takes in an abstract random number, and returns a real scalar. The basic object from which other Gaussians will be constructed is the standard scalar Gaussian, defined as follows.

Definition 1.24. *A random variable $z : \Omega \to \mathbb{R}$ is called* STANDARD GAUSSIAN *if it admits the Lebesgue density*

Standard Gaussian random variable

$$f(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \tag{1.62}$$

From this, we define general scalar Gaussians.

Definition 1.25. *A random variable $y : \Omega \to \mathbb{R}$ is called* GAUSSIAN *if there are scalars $\mu, \sigma \in \mathbb{R}$, and a standard Gaussian z, such that*

Gaussian random variable

$$y = \sigma z + \mu. \tag{1.63}$$

Note that we do *not* require $\sigma \geq 0$: hence, every Gaussian random variable is determined uniquely in distribution by the pair (μ, σ^2), which we call its *mean* and *variance*, respectively. We write $y \sim \mathrm{N}(\mu, \sigma^2)$. True to these parameter names, we have

$$\mathbb{E}(y) = \mu \qquad\qquad \mathbb{E}\Big((y-\mu)^2\Big) = \sigma^2. \tag{1.64}$$

A Gaussian random variable is called *centered* if $\mu = 0$. For a given variance σ^2 and standard Gaussian z, the expressions σz and $-\sigma z$ define two *different* centered Gaussians with the same distribution. At this stage, pointing these distinctions may appear needlessly pedantic: they will become more pronounced and important once we consider more general objects. The density of a Gaussian random variable, if it exists, takes on a form analogous to that of a standard Gaussian, namely

$$f(y) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y-\mu)^2}{2\sigma^2}\right). \tag{1.65}$$

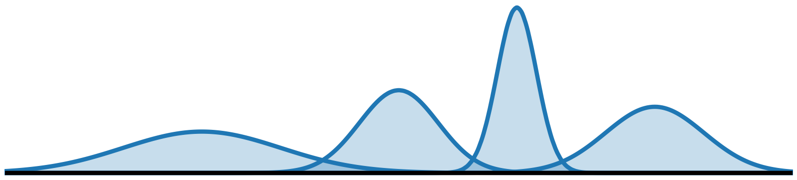


Figure 1.6. Probability density functions for a set of Gaussian random variables, each with different mean and standard deviation parameters. All of these can be obtained by shifting and rescaling any of the others via affine transformations.

This density is visualized in Figure 1.6. Note that the density will not exist if $\sigma^2 = 0$: the distributions of such Gaussians are Dirac measures centered at μ. By examining this density, one sees that Gaussian random variables respect the additive and multiplicative structures of the reals.

Affine maps between Gaussians

Proposition 1.26. *Let $y \sim \mathrm{N}(\mu, \sigma^2)$. Then for $a, b \in \mathbb{R}$ we have that*

$$ay + b \sim \mathrm{N}(a\mu + b, a^2\sigma^2). \tag{1.66}$$

Proof. Immediate by definition. ■

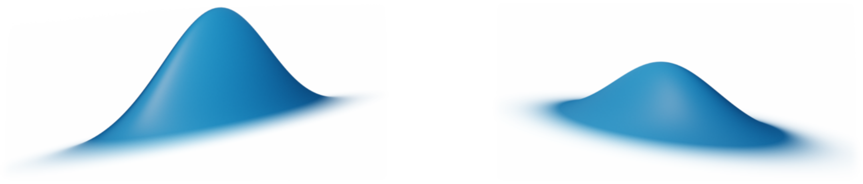
This compatibility with linear structure will be true at all levels of generality we consider. We now lift this definition to construct multivariate analogs.

1.3.3. Gaussian random vectors. A multivariate Gaussian random vector is a random variable taking values in $\mathbb{R}^d$. We write vectors defined in $\mathbb{R}^d$ in bold italics to emphasize this distinction, and similarly distinguish matrices from linear maps by writing the former in bold upface letters. As before, we begin by defining a standard Gaussian.

Standard multivariate Gaussian

Definition 1.27. *A random variable $\boldsymbol{z} : \Omega \to \mathbb{R}^d$ is called* STANDARD MULTIVARIATE GAUSSIAN *if its distribution is the product measure of the distributions of d standard Gaussians.*

Once the notion of a standard Gaussian is available, we can again define multivariate Gaussians as transformations of standard Gaussians.



(a) $\rho = 0.9$

(b) $\rho = -0.6$

Figure 1.7. Two multivariate Gaussian densities, in dimension two, with unit variances and different correlation coefficients ρ. As $|\rho| \to 1$, the individual components of the multivariate Gaussian become more and more dependent.

Multivariate Gaussian

Definition 1.28. *A random variable* $\boldsymbol{y} : \Omega \to \mathbb{R}^d$ *is called* MULTIVARIATE GAUSSIAN *if there is a vector* $\boldsymbol{\mu} \in \mathbb{R}^d$, *matrix* $\mathbf{L} \in \mathbb{R}^{d \times d}$, *and standard multivariate Gaussian* $\boldsymbol{z}$, *such that*

$$\boldsymbol{y} = \mathbf{L}\boldsymbol{z} + \boldsymbol{\mu}. \tag{1.67}$$

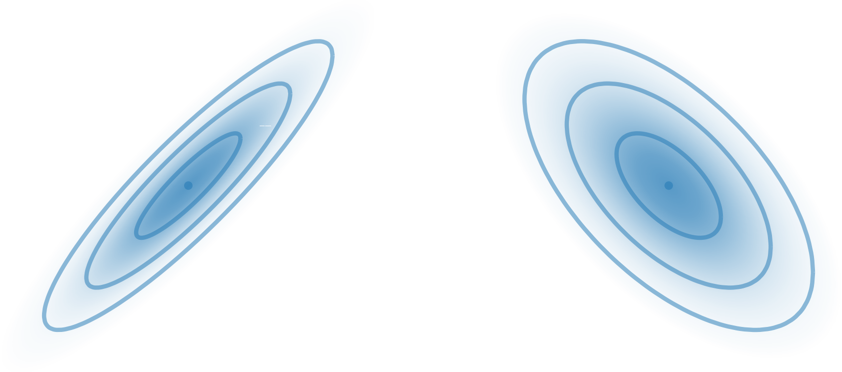
Every multivariate Gaussian is determined by its *mean vector* $\boldsymbol{\mu}$ and positive semi-definite *covariance matrix* $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^T$, and, as before, is called *centered* if $\boldsymbol{\mu} = \mathbf{0}$. We write $\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. We can obtain a centered multivariate Gaussian with a given distribution by calculating a *matrix square root* of $\boldsymbol{\Sigma}$, multiplying it with a standard Gaussian. The mean and covariance are

$$\mathbb{E}(\boldsymbol{y}) = \boldsymbol{\mu} \qquad \operatorname{Cov}(\boldsymbol{y}) = \mathbb{E}\Big((\boldsymbol{y} - \boldsymbol{\mu})(\boldsymbol{y} - \boldsymbol{\mu})^T\Big) = \boldsymbol{\Sigma} \tag{1.68}$$

which mirror the previous situation. If the determinant $|\boldsymbol{\Sigma}|$ is non-zero, the density, displayed in two different ways in Figures 1.7 and 1.8, is

$$f(\boldsymbol{y}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{y} - \boldsymbol{\mu})\right) \tag{1.69}$$

which now might not exist even if the distribution of $\boldsymbol{y}$ is not Dirac. In such cases, one can see that at least some eigenvalues of $\boldsymbol{\Sigma}$ must be zero, and so Gaussians which do not admit densities must, when viewed in an appropriate basis, be products of Dirac measures with Gaussians which do admit densities. Already in the multivariate case, then, we see the technical power of densities weakening: this will become more pronounced as we consider more general settings. Affine maps, however, behave as before.



(a) $\rho = 0.9$ (b) $\rho = -0.6$

Figure 1.8. Quantile ellipsoids for two multivariate Gaussian densities in dimension two with unit variances, different correlation coefficients ρ, and quantile levels equal to 0.8, 0.95, and 0.99.

Affine maps between multivariate Gaussians

Proposition 1.29. *Let* $\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$. *Then for* $\mathbf{A} \in \mathbb{R}^{d \times d}$ *and* $\boldsymbol{b} \in \mathbb{R}^d$ *we have*

$$\mathbf{A}\boldsymbol{y} + \boldsymbol{b} \sim \mathrm{N}(\mathbf{A}\boldsymbol{\mu} + \boldsymbol{b}, \mathbf{A}\mathbf{\Sigma}\mathbf{A}^T). \tag{1.70}$$

Proof. Immediate by definition. ■

We now pause and reflect. First, we distinguish $\mathbb{R}^d$ from a generic d-dimensional vector space: the former comes with a product structure $\mathbb{R}^d = \mathbb{R} \times .. \times \mathbb{R}$, including projection maps onto each coordinate, which in turn induce a *canonical* choice of inner product given by the Euclidean dot product. A generic finite-dimensional vector space lacks this structure: it admits many different inner products, and provides for no canonical choice.

With this in mind, we observe that we have not actually used the product structure of $\mathbb{R}^d$: suppose that V is a finite-dimensional inner product space, and define

$$y(\omega) = \sum_{i=1}^{d} z_i(\omega) e_i \qquad\qquad z_i \sim \mathrm{N}(0, 1) \tag{1.71}$$

where e_i is any orthonormal basis. By noting that the choice of basis e_i induces a Borel isomorphism $V \cong \mathbb{R}^d$, it is easy to see that this definition is basis-independent, since linear maps associated with changes of orthonormal bases are represented by orthogonal matrices. This expression therefore defines a standard Gaussian with respect to the given inner product.

Subtle difference such as the ones considered become increasingly important in the sequel. Therefore, we introduce an alternative definition to help build intuition for later.

Definition 1.30. *A random variable $\boldsymbol{y} : \Omega \to \mathbb{R}^d$ is called* MULTIVARIATE GAUSSIAN *if, for any vector $\boldsymbol{\phi} \in \mathbb{R}^d$, the Euclidean dot product*

Multivariate Gaussian (duality)

$$\boldsymbol{\phi} \cdot \boldsymbol{y} : \Omega \to \mathbb{R} \tag{1.72}$$

is univariate Gaussian.

This definition turns out to be equivalent to the original one.

Proposition 1.31. *The notions of multivariate Gaussians in the sense of transformations and in the sense of duality coincide.*

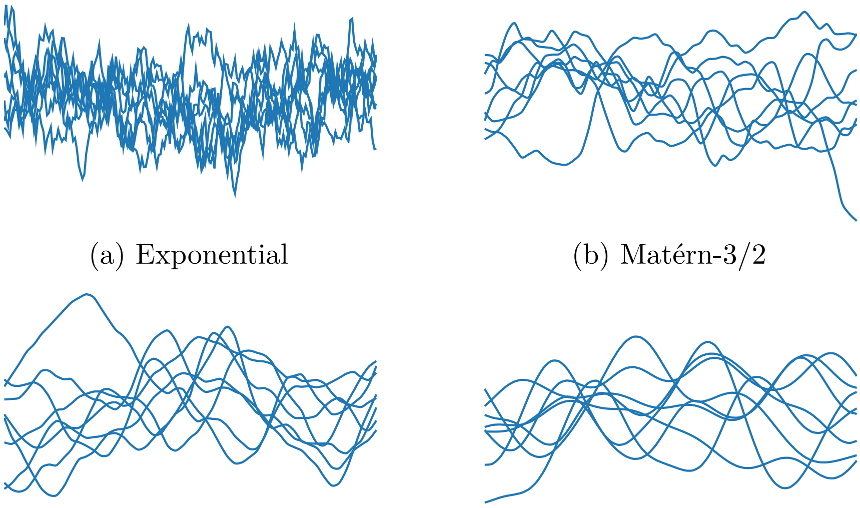
Proof. Since the dot product is a linear map, it is clear that multivariate Gaussians in the sense of transformations are also Gaussian in the sense of duality. To see the other direction, consider unit vectors $\boldsymbol{\phi}$ where all coordinates except one are zero. By using dot products with such vectors to reassemble the mean vector and covariance matrix, one sees that the claim follows from eigenvalue factorization of positive semi-definite matrices. ■

This definition allows one to begin imagining what a substantially more general notion of Gaussianity might look like: dot products become linear functionals, and covariance matrices become bilinear forms. The preceding argument even suggests that such random vectors can be studied using spectral theory. Of course, in infinite-dimensional settings, topological and analytic considerations come into play, making theory more difficult. We develop these ideas subsequently, but first consider a simpler setting.

1.3.4. Gaussian random functions. We now consider Gaussian random functions, which are the first notion of a Gaussian random variable that is generally called a *Gaussian process*. Here, we will adopt a *bottom-up* view which, from a technical perspective, departs slightly from the notions introduced so far.

Recall that for a set X, an $\mathbb{R}$-valued *stochastic process* is a map $f : \Omega \times X \to \mathbb{R}$ measurable in its first argument. If we have a set of points $x_1, .., x_n \in X$, we

Stochastic process



(a) Exponential

(b) Matérn-3/2

(c) Matérn-5/2

(d) Squared exponential

Figure 1.9. Random functions generated from Gaussian processes with four different covariance kernels, which differ in particular according to their regularity, ranging from nowhere-differentiable in the exponential case to infinitely-differentiable in the squared exponential case.

can plug them into f to obtain a map $(f(\cdot, x_1), .., f(\cdot, x_n)) : \Omega \to \mathbb{R}^n$, which, by virtue of being a product of measurable maps, is a random variable. We call its distribution a *finite-dimensional marginal distribution.* Using this notion, we define Gaussian processes.

Finite-dimensional marginal distribution

Gaussian process (stochastic process)

Definition 1.32. *Let X be a set. A random process $f : \Omega \times X \to \mathbb{R}$ is called a* GAUSSIAN PROCESS *if, for any finite set of points $x_1, .., x_n \in X$, the random variable $(f(\cdot, x_1), .., f(\cdot, x_n)) : \Omega \to \mathbb{R}^n$ is multivariate Gaussian.*

Immediately upon writing this definition, one is left to wonder: does it actually make sense? In particular, do Gaussian processes in the sense given here exist?

Under general conditions on the finite-dimensional marginals, Kolmogorov's Consistency Theorem states that there exists a unique probability measure on a suitable σ-algebra, called the cylindrical σ-algebra and defined below, whose finite-dimensional projections coincide with the distributions of the random variables written above. Thus, Gaussian processes exist so long as a family of multivariate Gaussians satisfying the conditions of Kolmogorov's Consistency Theorem can be found. We visualize such processes in Figure 1.9.

The same results also allow us to reinterpret Gaussian processes as *function-valued random variables*, which, in many ways, are a more natural point of view to take. Define $\mathbb{R}^X = \{f : X \to \mathbb{R}\}$ and equip it with the cylindrical σ-algebra, namely the smallest σ-algebra containing all sets of the form $\{f \in \mathbb{R}^X : f(x_1), .., f(x_n) \in A\}$, where A is a measurable set on $\mathbb{R}^n$, for all n, to make it into a measurable space. Let $V \subseteq \mathbb{R}^X$, and equip it with the subset σ-algebra. We can then reinterpret Gaussian processes as follows.

Gaussian process (random function)

Definition 1.33. *Let X be a set. A random variable $f : \Omega \to V \subseteq \mathbb{R}^X$ is called a* GAUSSIAN PROCESS *if, for any finite set of points $x_1, .., x_n \in X$, the random variable $(f(\cdot)(x_1), .., f(\cdot)(x_n)) : \Omega \to \mathbb{R}^n$ is multivariate Gaussian.*

It is clear that every Gaussian process in the random process sense induces a Gaussian process in the random variable sense, and vice versa. Thus, these two notions are simply two different ways of viewing the same object.

We can then ask, what properties of multivariate Gaussians are still true in this setting? Since a Gaussian process is uniquely determined by its finite-dimensional marginals, we need to find functions that generate a family of mean vectors and covariance matrices which are positive semi-definite. The former is straightforward: every function $\mu : X \to \mathbb{R}$ can be evaluated at a finite set of points $x_1, .., x_n$ to obtain a mean vector $\boldsymbol{\mu} = \mu(x_1, .., x_n)$. The latter requires only slightly more thinking.

Positive semi-definite kernel

Definition 1.34. *A symmetric function $k : X \times X \to \mathbb{R}$ is called a* POSITIVE SEMI-DEFINITE KERNEL *if, for any finite set of points $x_1, .., x_n \in X$, the kernel matrix*

$$\begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix} \tag{1.73}$$

is positive semi-definite.

We can recover such a kernel from a given Gaussian process.

Covariance kernel

Definition 1.35. *Define the* COVARIANCE KERNEL *of a Gaussian process to be*

$$k(x, x') = \operatorname{Cov}(f(\cdot, x), f(\cdot, x')). \tag{1.74}$$

It is then clear that the pair (μ, k) uniquely define a Gaussian process, and so we write $f \sim \operatorname{GP}(\mu, k)$. Defining a centered Gaussian process then amounts to defining a positive semi-definite kernel.

In the Euclidean case, this can be done straightforwardly by noting that (i) the linear kernel $k(x, x') = \langle x, x' \rangle$ is positive semi-definite by non-degeneracy of the inner product, (ii) that sums, powers, and limits of positive semi-definite kernels are positive semi-definite. By this technique, we see that the widely-used exponential and squared exponential kernels are positive semi-definite.

For many spaces of functions $V \subseteq \mathbb{R}^X$, we can define addition and scalar multiplication, thereby making V into a vector space. If we do so, then, as before, affine maps preserve Gaussianity.

Affine maps between Gaussian processes

Proposition 1.36. *Let $f \sim \operatorname{GP}(\mu, k)$, and equip V and $W \subseteq \mathbb{R}^X$ with the topology of pointwise convergence. Suppose that f can be written as an infinite sum of deterministic basis functions as $f(\omega, x) = \sum_{n=0}^{\infty} w_i(\omega) f_i(x)$ where w_i are independent Gaussian random variables. If $\mathcal{A} : V \to W$ is a continuous linear map, and $b \in W$, then $\mathcal{A}f + b$ is a Gaussian process.*

Proof. It is clear by considering a set of finite-dimensional marginal distributions that adding b preserves Gaussianity. Writing

$$(\mathcal{A}f)(x) = \mathcal{A}\left(\sum_{i=0}^{\infty} w_i(\omega) f_i(\cdot)\right)(x) = \sum_{i=0}^{\infty} w_i(\omega)(\mathcal{A}f_i)(x) \tag{1.75}$$

gives the statement for all finite-dimensional marginals, since sums and limits of Gaussian random variables are Gaussian. The claim follows. ∎

This is a substantially weaker assertion than previously: though we can conclude that affine maps of Gaussian processes are Gaussian, for a generic affine map we cannot immediately determine the form of the resulting kernel. Moreover, the conditions are unnatural: by supposing them, we have more-or-less *assumed* the resulting kernel to exist. The problem here is that the notion of a *kernel* is too rigid to permit a broad statement—an analogous property will hold if this is replaced with a different notion of covariance.

The situation for the other properties considered previously is even worse. In particular, it is clear that, as an infinite-dimensional object, f does not admit a probability density analogous to the ones considered previously, because an infinite-dimensional Lebesgue measure, in the sense of a locally finite translation invariant measure, does not exist. It is also not clear what a *standard* Gaussian process, nor what the analog of a matrix square root of a positive semi-definite kernel, might be.

The loss of these technical tools has consequences on what can be said about Gaussian processes. In Chapter 3, we will study Gaussian processes whose domain X is a Riemannian manifold. In that setting, one cannot begin by proving positive semi-definiteness of linear kernels, because there is simply no useful analog of this concept. Defining positive semi-definite kernels there turns out to be non-trivial, and the kernels we study do not admit simple expressions like they do in the Euclidean case.

We therefore proceed to adopt a function-analytic perspective which is more technical, but significantly more powerful, and will allow us to recover the previous set of tools in a much more pure and abstract form.

1.3.5. Gaussian processes in the sense of duality. In the preceding section, we established a notion of a *Gaussian process*, generalizing the notion of Gaussianity to random functions. In developing this viewpoint, some of the appeal of Gaussianity was lost: in particular, unlike in the finite-dimensional setting, no simple covariance kernel transformation rule was available. Here, we consider an alternative function-analytic view which avoids these limitations at cost of a higher degree of abstraction.

Gaussian process (duality)

Definition 1.37. *Let V and W be a pair of measurable real vector spaces equipped with a jointly measurable non-degenerate bilinear form $\langle \cdot \mid \cdot \rangle : W \times V \to \mathbb{R}$. We say that a random variable $f : \Omega \to V$ is a* GAUSSIAN PROCESS IN THE SENSE OF DUALITY *if, for any $\phi \in W$, the scalar-valued random variable $\langle \phi \mid f \rangle$ is Gaussian.*

This is an exceedingly broad definition, which is more useful as an organizing framework connecting different concepts than as a technical tool in its own right. At first glance, it is not clear what, if anything, this notion has to do with the Gaussian processes we have considered previously. To better understand this, an illustrative example is in order.

Let $f \sim \mathrm{GP}(\mu, k)$ be a Gaussian process defined on $[0, 1]$ whose sample paths are almost surely continuous, and note by compactness that they are automatically bounded. Our Gaussian process can therefore be viewed as a random variable $f : \Omega \to C^0([0, 1]; \mathbb{R})$. We endow the latter space with the supremum norm, making it into a Banach space. If we take another function $\phi \in C^\infty([0, 1], \mathbb{R})$, called the *test function*, we can define the pairing

$$\langle \phi \mid f \rangle = \int_0^1 \phi(x) f(x) \, \mathrm{d}x \tag{1.76}$$

which by boundedness is almost surely finite. Now, note that since sums of Gaussians are Gaussian, Riemann sums of Gaussian processes are Gaussian. Since limits of Gaussians are Gaussian, the quantity $\langle \phi \mid f \rangle$ will be a *Gaussian* scalar. Therefore, a Gaussian in the sense of duality can loosely be thought of as a Gaussian process whose integral against arbitrary test functions is always Gaussian—an adaptation of the dot product notion encountered previously to the infinite-dimensional setting.

Of course, the above only describes how one can intuitively think about Gaussians in the sense of duality. For general Gaussians, there is no defined notion of an *integral*—only of a dual pairing. Such a Gaussian also need not be a random real-valued function: it's possible to consider random distributions and other objects generalizing the usual notion of a function. It is therefore clear this view is very general.

This generality comes with a price: the resulting random vectors become more abstract, and it is much more difficult to understand whether or not they actually exist or say anything useful about them. This difficulty is handled by specializing to less-general settings: for example, Bogachev [9] studies the setting where V is a locally convex topological vector space and W is its dual, and Hairer [48] and Lifshits [71] study cases where V is a Banach or Hilbert space. In those cases, a number of results are available.

Provided we are considering a Gaussian f which does exist, what objects play the role of a mean and covariance, and uniquely characterize f? Since the canonical pairing $\langle \cdot \mid \cdot \rangle$ is linear and non-degenerate, we know that $f = g$ holds if and only if $\langle \phi \mid f \rangle = \langle \phi \mid g \rangle$ for all $\phi \in W$, with both equalities used in the same sense. It is therefore clear that the *mean* of a Gaussian process is simply a vector $\mu \in V$, since such a vector uniquely determines all expectations $\mathbb{E}\langle \phi \mid f \rangle$. The covariance is only slightly more subtle.

Definition 1.38. *We say that a symmetric positive semi-definite bilinear form $k : W \times W \to \mathbb{R}$ is a* COVARIANCE FORM.

Covariance form

As before, we can construct a covariance form from a given Gaussian process.

Definition 1.39. *The* COVARIANCE FORM *of a Gaussian in the sense of duality is defined by*

$$k(\phi, \psi) = \operatorname{Cov}(\langle \phi \mid f \rangle, \langle \psi \mid f \rangle). \tag{1.77}$$

Covariance form of a Gaussian process

It is clear that two Gaussian processes f and g are equal in distribution if and only if they have the same mean and covariance form. This can be seen by noting these requirements force $\langle \phi \mid f \rangle = \langle \phi \mid g \rangle$ to hold for all $\phi \in W$.

We now ask: what relationship does the covariance form have with the covariance kernel defined previously? Consider again our Gaussian process f defined on the space of continuous functions. For any pair of test functions $\phi, \psi \in C^\infty([0,1]; \mathbb{R})$, the symmetric positive semi-definite bilinear form

$$(\phi, \psi) \mapsto \int_0^1 \int_0^1 \phi(x) k(x, x') \psi(x') \, \mathrm{d}x \, \mathrm{d}x' \tag{1.78}$$

is the covariance form of f.

So far, this perspective mirrors the preceding ones, albeit with a more abstract presentation. We have only introduced definitions. These definitions, however, suffice to recover a notion of affine maps.

Proposition 1.40. *Let $f \sim \mathrm{GP}(\mu, k)$. For $\mathcal{A} : V \to V'$ and $b \in V'$ we have*

$$\mathcal{A} f + b \sim \mathrm{GP}(\mathcal{A}\mu + b, k(\mathcal{A}^*(\cdot), \mathcal{A}^*(\cdot))) \tag{1.79}$$

where V' and W' are a dual pair, and $\mathcal{A}^* : W' \to W$ is the adjoint operator defined by $\langle \mathcal{A}^* \phi \mid v \rangle = \langle \phi \mid \mathcal{A} v \rangle$.

Affine maps between Gaussian processes

Proof. Immediate by definition. ■

Suppose now that V is a locally convex topological vector space, and $W = V^*$ its topological dual. Since the covariance form is a map $k : V^* \times V^* \to \mathbb{R}$, it gives rise to an operator $\mathcal{K} : V^* \to V^{**}$, called the *covariance operator*, which in the case that V is reflexive becomes an operator $\mathcal{K} : V^* \to V$ via

Covariance operator

composition with the canonical isomorphism given by reflexivity. For our running example, this operator is given by

$$\phi \mapsto \int_0^1 \phi(x)k(x,\cdot)\,\mathrm{d}x. \tag{1.80}$$

One can easily see that the transformation rule for covariance operators under affine maps is given by

$$\mathcal{K} \mapsto \mathcal{A}\mathcal{K}\mathcal{A}^*. \tag{1.81}$$

If it is often unclear whether or not a Gaussian process with a given covariance form exists, it can be even less clear whether or not a Gaussian process with a given covariance operator exists. Still, this notion is important, as these operators can potentially be studied using spectral theory. We now move to the final concept we consider at this stage: that of a *standard Gaussian.*

Gaussian white noise

Definition 1.41. *Let $W \subseteq H$ where H is a Hilbert space. We say that $\mathcal{W} : \Omega \to V$ is a* GAUSSIAN WHITE NOISE *process if its covariance form coincides with the inner product in all cases where the former is defined.*

In cases where $\mathcal{K}$ admits a sufficiently rich spectral theory, this can potentially provide a suitable notion of an *operator square root* which relates Gaussian white noise to other Gaussian processes.

With these definitions in hand, we have recovered the two technical notions lost when transitioning from multivariate Gaussians to Gaussian processes in the sense of random functions. The introduced machinery gives a way of constructing Gaussian processes that does not rely on defining a kernel: (i) construct a Gaussian white noise process, and (ii) define the Gaussian process of interest as an affine map of white noise.

We employ a variation of this strategy in Chapter 3 to construct Gaussian processes on Riemannian manifolds, where defining a kernel directly leads to difficulties. Loosely speaking, we do so by taking V to be a space of distributions, and $\mathcal{A}$ to be the inverse of a differential operator. This is chosen to ensure that $\mathcal{A}$ smooths its inputs, so that its output is regular enough to be understood as a random function.

This line of thought leads one to the theory of stochastic partial differential equations driven by Gaussian white noise. The main point swept here under the rug is that there is a convenient way to sidestep much of the analysis needed to carry out the above calculations using the theory of reproducing kernel Hilbert spaces. Roughly, this amounts to instead working with certain vector spaces that uniquely determine the Gaussian processes of interest.

We also use the theory of Gaussians in the sense of duality for a second purpose: to construct a *coordinate-free* notion of Gaussianity as a suitable building block for constructing Gaussian vector fields on Riemannian manifolds. The issue here is that the Gaussian process *cannot* be understood as a real-valued random function due to topological obstructions—it is instead a *random section*, which we define in Chapter 3. Finite-dimensional Gaussians in the sense of duality end up being the right tool for this setting.

For this, we prove a general existence theorem on Gaussians in the sense of duality in finite-dimensional settings. Aside from giving an intrinsic reinterpretation of the multivariate Gaussians described previously, this ensures our framework is suited for its purpose in the coordinate-free setting.

Proposition 1.42. *Let V be a finite-dimensional real topological vector space, and let $W = V^*$ be its topological dual. Then for any vector $\mu \in V$ and covariance form $k : V^* \times V^* \to \mathbb{R}$, there exists a unique-in-distribution random vector $y \sim \mathrm{N}(\mu, k)$.*

Proof. To prove this, choose a basis e_i on V, and let e^i be the dual basis. Let $\mathcal{E} : V \to \mathbb{R}^d$ be the continuous linear isomorphism induced by the basis, and define $y = \mathcal{E}^{-1}\boldsymbol{y}$ with $\boldsymbol{y} \sim \mathrm{N}(\boldsymbol{\mu}, \mathbf{K})$ defined by

$$\boldsymbol{\mu} = \begin{bmatrix} \langle e^1 \mid \mu \rangle \\ \vdots \\ \langle e^d \mid \mu \rangle \end{bmatrix} \qquad \mathbf{K} = \begin{bmatrix} k(e^1, e^1) & \dots & k(e^1, e^d) \\ \vdots & \ddots & \vdots \\ k(e^d, e^1) & \dots & k(e^d, e^d) \end{bmatrix}. \tag{1.82}$$

It is clear by direct calculation using Gaussians on $\mathbb{R}^d$ that the resulting vector is Gaussian with the right mean and covariance form. The claim follows by noting that the assumed non-degeneracy of the dual pairing forces the distribution of every Gaussian in the sense of duality to be uniquely determined by its mean and covariance form. ■

Here, we see the key difference between the coordinate-free view and the matrix-vector view considered previously: the Gaussian random vector can be viewed as a real-valued multivariate Gaussian in any basis, but itself is defined on V independent of this choice. The value of considering this distinction in the first place will become clear in Chapter 3. Here, finite-dimensionality suffices to ensure existence: the story in infinite-dimensional settings is completely different and requires case-by-case analysis.

To conclude, we reflect on the introduced ideas. We began by studying Gaussian random variables and random vectors, before generalizing these to Gaussian random functions, which offered a concrete framework where certain aspects of Gaussianity were seemingly lost. Adopting a function-analytic view restored these aspects, at cost of increased abstraction. This also gave a coordinate-free way to reinterpret the preceding multivariate constructions.

While much of the technical power of the duality framework is extraneous for our purposes, studying it nonetheless helps provide a unified conceptual perspective from which to interpret our developments. By considering these notions, it becomes much clearer how one should understand the constructions encountered later, which might otherwise appear as if they arise out of thin air. This completes our study of Gaussianity for its own sake, independent of Bayesian learning and other machine-learning-related considerations.

1.4. Discussion

The preceding sections paint a rich and detailed picture of what a mathematical theory of decision-making under uncertainty looks like. We now recap the steps taken so far, and reflect on them, before proceeding to describe contributions to be presented.

We began with the concept of probability, constructed in the language of measure theory. We used this language to formalize the concept of learning via the notion of conditional probability, thereby obtaining the theory of *Bayesian learning.* By working in an abstract measure-theoretic setting, we obtained a formalism suitable for learning about very general unknown quantities of interest.

We then took a step back, examining how to formalize the notion of an agent selecting actions in an unknown environment on basis of interactions, obtaining the key concept of a *Markov decision process.* We then immediately restricted to the simpler setting of *multi-armed bandits.* We saw that model-based algorithms built atop Bayesian learning yielded decision systems that perform effectively. With these notions, we described how to efficiently solve global optimization problems using *Bayesian optimization.*

To transform the preceding ideas into a workable class of methods, we proceeded to study *Gaussian process* models in depth. We developed these models in sequence, starting from the simplest settings, and ending with the highest generality. These ideas provide us with key tools for understanding Gaussian processes, so that we can use them as building blocks of Bayesian models and high-performance decision systems atop those models.

It is worth pausing to reflect on the merits of the setting chosen, within a broader context of artificial intelligence. In choosing to work with Bayesian learning, we opted to represent uncertainty using probability—a powerful but computationally limiting choice. This choice was counterbalanced by working with simple models in bandit-like settings, and is most effective when the decisions of interest must be made in a data-efficient manner that only algorithms with near-asymptotically-optimal regret can achieve.

Not all settings fit these criteria well. In many reinforcement learning problems of interest in robotics, the complexity of the dynamics—which, for multi-armed bandits, are totally absent from the problem—is a key difficulty. Gaussian processes are largely not expressive enough to represent multi-object collision dynamics and related phenomena. We have also not addressed partial observations—another key difficulty in that setting.

On the other hand, no other currently known theoretical framework comes close to understanding decision to the degree of command one can obtain from the notions described. In the absence of a probabilistic framework, it is unclear how to assess, represent, and propagate uncertainty to resolve explore-exploit tradeoffs and minimize regret in non-trivial settings. Thus, when probabilistic methods can be considered, they absolutely should be.

As a step towards building increasingly sophisticated decision systems, it seems fruitful to expand probabilistic approaches built via Bayesian learning to more general settings. Improved understanding of these phenomena may yield lessons of broad interest to the application of decision systems. Contributions presented here include development of *pathwise conditioning* methods for making Gaussian process models easier to work with, and a variety of *non-Euclidean Gaussian processes*, both described next.

Chapter 2

Pathwise Conditioning

Gaussian processes admit analytic conditional distributions, making them a key model class for Bayesian learning. The standard way of viewing these conditional distributions mirrors the general measure-theoretic setup common to all Bayesian models, and has strongly influenced how people think about Gaussian processes.

In the early 1970s, an alternative view emerged in the geostatistics community. Miraculously, in the Gaussian case it is also possible to develop conditioning in a manner not purely based on *distributions*, but on *random variables* directly. This view turns out to lift from the multivariate to the Gaussian process setting, yielding *pathwise* representations of posterior Gaussian processes, which have largely been overlooked in machine learning until now.

The pathwise perspective turns out to be a powerful point of view with wide-ranging consequences. We will show how to use it to resolve a long-standing difficulty in Bayesian optimization: constructing a posterior approximation whose computational cost is linear both at training time and at test time, with excellent approximation properties and error control.

One of the key ingredients used within the construction will be basis function expansions of *prior* Gaussian processes. We will thus examine a number of methods for constructing such expansions for different classes of priors. We will also reinterpret sparse approximations in a function-based manner simpler than the typical viewpoint. We conclude by benchmarking Bayesian optimization using pathwise sampling. We proceed to these developments.

2.1. Conditioning multivariate Gaussians

We now describe conditioning of multivariate Gaussians. Recall that using Bayes' Rule, a prior and likelihood combine into a joint distribution, which factorizes into the marginal distribution of the data and the posterior. The posterior is the conditional distribution of the parameters given the data, which is unique almost everywhere with respect to the marginal distribution. We now study how to represent this distribution for the case of interest.

2.1.1. Distributional conditioning. The most obvious way to represent a Gaussian conditional distribution is to simply calculate its distribution as a closed-form analytic expression. This is given below.

Multivariate Gaussian conditionals

Proposition 2.1. *Let*

$$\begin{bmatrix}\boldsymbol{\theta}\\ \boldsymbol{y}\end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix}\boldsymbol{\mu}_{\boldsymbol{\theta}}\\ \boldsymbol{\mu}_{\boldsymbol{y}}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}} & \boldsymbol{\Sigma}_{\boldsymbol{\theta y}}\\ \boldsymbol{\Sigma}_{\boldsymbol{y\theta}} & \boldsymbol{\Sigma}_{\boldsymbol{yy}}\end{bmatrix}\right) \tag{2.1}$$

be non-singular. Then we have that

$$(\boldsymbol{\theta} \mid \boldsymbol{y})(\cdot, \boldsymbol{\gamma}) \sim \mathrm{N}\Big(\boldsymbol{\mu}_{\boldsymbol{\theta}} + \boldsymbol{\Sigma}_{\boldsymbol{\theta y}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}(\boldsymbol{\gamma} - \boldsymbol{\mu}_{\boldsymbol{y}}), \boldsymbol{\Sigma}_{\boldsymbol{\theta\theta}} - \boldsymbol{\Sigma}_{\boldsymbol{\theta y}}\boldsymbol{\Sigma}_{\boldsymbol{yy}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{y\theta}}\Big). \tag{2.2}$$

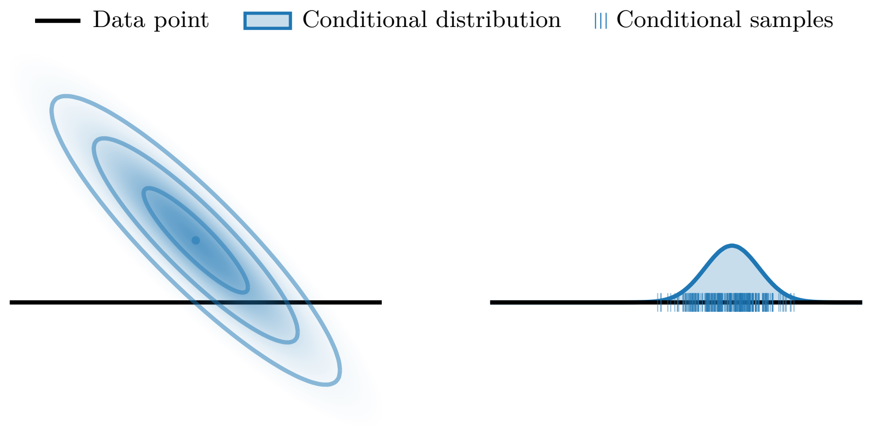
Proof. By non-singularity, $(\boldsymbol{\theta}, \boldsymbol{y})$ admits a Lebesgue density, and the claim follows by direct calculation via applying Bayes' Rule for densities. ■

See Rasmussen and Williams [89], Appendix A, for a full derivation. The non-singularity requirement here is more-or-less necessary: otherwise, the marginal distribution of $\boldsymbol{y}$ may admit too many null sets, rendering the desired conditional distribution non-unique, except in regions where one can apply a linear map to recover a suitable Lebesgue density and apply the above argument on the obtained subspace.

If one wants to work with this expression numerically, then it's possible to simply compute the desired conditional mean and covariance. Then, one can generate conditional samples via the expression

$$(\boldsymbol{\theta} \mid \boldsymbol{y})(\omega, \boldsymbol{\gamma}) = \mathbf{L}_{\boldsymbol{\theta}|\boldsymbol{y}}\boldsymbol{z}(\omega) + \boldsymbol{\mu}_{\boldsymbol{\theta}|\boldsymbol{y}} \qquad \boldsymbol{z} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}) \tag{2.3}$$

where $\boldsymbol{\mu}_{\boldsymbol{\theta}|\boldsymbol{y}}$ is the conditional mean, and $\mathbf{L}_{\boldsymbol{\theta}|\boldsymbol{y}}$ is a Cholesky factor of the conditional covariance, or any other matrix square root obtained numerically.



(a) Calculate conditional

(b) Draw samples

Figure 2.1. Illustration of distributional conditioning of a bivariate Gaussian. Here, we form the joint distribution, calculate the conditional distribution, and then draw samples from it. Note that all steps except the very last one are *distributional* in nature and do not involve the use of random variables, which only appear at the very end of the process.

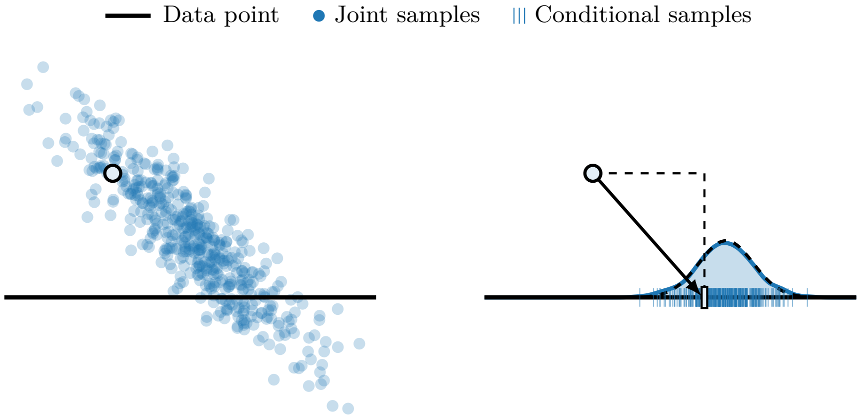
We illustrate this procedure in Figure 2.1. This enables one to calculate any quantity of interest depending on the conditional distribution numerically via the Monte Carlo method. The computational costs will in general be cubic in the dimension of both $\boldsymbol{\theta}$ and $\boldsymbol{y}$, owing to the need to compute $\mathbf{L}_{\boldsymbol{\theta}|\boldsymbol{y}}$ and invert $\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}$, respectively.

2.1.2. Pathwise conditioning. The preceding considerations gave closed-form analytic expressions for Gaussian conditionals in terms of matrix-vector expressions that can be computed numerically. From this, one might be tempted to conclude that there is nothing more to say about conditioning multivariate Gaussians—this, however, would miss an alternative view: Gaussian conditionals, which in general are a purely *distributional* notion, can also be described in a *pathwise* manner using *random variables.*

Theorem 2.2. *For $\boldsymbol{\theta}, \boldsymbol{y}$ defined in Proposition 2.1, we have that*

Matheron's update rule

$$(\boldsymbol{\theta} \mid \boldsymbol{y})(\omega,\gamma) = \boldsymbol{\theta}(\omega) + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}(\gamma - \boldsymbol{y}(\omega)). \tag{2.4}$$



(a) Sample jointly (b) Transform into conditional

Figure 2.2. Illustration of pathwise conditioning of a bivariate Gaussian. Here, we first sample a random vector from the joint distribution, then transform it into a sample from the conditional distribution. These steps are called *pathwise* because they are defined directly using random variables, rather than indirectly through probability distributions.

Proof. By direct calculation,

$$\mathbb{E}(\boldsymbol{\theta} + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}(\gamma - \boldsymbol{y})) = \boldsymbol{\mu}_{\boldsymbol{\theta}} + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}(\gamma - \boldsymbol{\mu}_{\boldsymbol{y}}) = \mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y} = \gamma) \tag{2.5}$$

and

$$\begin{aligned} \operatorname{Cov}(\boldsymbol{\theta} + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}(\gamma - \boldsymbol{y})) &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{\theta}} - 2\boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{\theta}} && (2.6) \\ &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{\theta}} - \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{\theta}} = \operatorname{Cov}(\boldsymbol{\theta} \mid \boldsymbol{y} = \gamma) && (2.7) \end{aligned}$$

where we have cancelled a factor of $\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}\boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}$ in the middle term. ■

This is illustrated in Figure 2.2. The above argument affirms the claim, but gives few hints on where this expression originates or how to obtain it from first principles. To better understand this, we now prove Theorem 2.2 in a different way. To do so, we first prove a plug-in property of conditioning.

Lemma 2.3. *Consider three random vectors $\boldsymbol{a} : \Omega \to \mathbb{R}^m$, $\boldsymbol{b} : \Omega \to \mathbb{R}^n$, and $\boldsymbol{c} : \Omega \to \mathbb{R}^m$ such that*

$$\boldsymbol{a} = f(\boldsymbol{b}) + \boldsymbol{c} \tag{2.8}$$

where $f : \mathbb{R}^n \to \mathbb{R}^m$ is a measurable function, and where the random variables $\boldsymbol{b}$ and $\boldsymbol{c}$ are independent. Then we have

$$(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) = f(\boldsymbol{\beta}) + \boldsymbol{c}. \tag{2.9}$$

Proof. This follows by direct calculation by writing

$$\begin{aligned}
\int_{A_{\boldsymbol{b}}} \pi_{\boldsymbol{a}|\boldsymbol{b}}(A_{\boldsymbol{a}} \mid \boldsymbol{\beta}) \, \mathrm{d}\pi_{\boldsymbol{b}}(\boldsymbol{\beta}) &= \mathbb{P}(\boldsymbol{a} \in A_{\boldsymbol{a}}, \boldsymbol{b} \in A_{\boldsymbol{b}}) && (2.10) \\
&= \mathbb{P}(f(\boldsymbol{b}) + \boldsymbol{c} \in A_{\boldsymbol{a}}, \boldsymbol{b} \in A_{\boldsymbol{b}}) && (2.11) \\
&= \int_{\mathbb{R}^m \times \mathbb{R}^n} \mathbb{1}_{f(\beta)+\varsigma \in A_{\boldsymbol{b}}, \beta \in A_{\boldsymbol{b}}} \, \mathrm{d}(\pi_{\boldsymbol{c}} \otimes \pi_{\boldsymbol{b}})(\boldsymbol{\varsigma}, \boldsymbol{\beta}) && (2.12) \\
&= \int_{A_{\boldsymbol{b}}} \int_{\mathbb{R}^m} \mathbb{1}_{f(\beta)+\varsigma \in A_{\boldsymbol{b}}} \, \mathrm{d}\pi_{\boldsymbol{c}}(\boldsymbol{\varsigma}) \, \mathrm{d}\pi_{\boldsymbol{b}}(\boldsymbol{\beta}) && (2.13) \\
&= \int_{A_{\boldsymbol{b}}} \mathbb{P}(f(\boldsymbol{\beta}) + \boldsymbol{c} \in A_a) \, \mathrm{d}\pi_{\boldsymbol{b}}(\boldsymbol{\beta}) && (2.14)
\end{aligned}$$

where we have used independence to represent the probability as an integral over a product measure, followed by Tonelli's Theorem. By the Disintegration Theorem, $\pi_{\boldsymbol{a}|\boldsymbol{b}}$ is $\pi_{\boldsymbol{b}}$-a.e. unique, implying that the conditional distributions of interest are equal, and the claim follows. ■

The key idea behind our argument will be to choose the *conditional expectation* for our function f, or more precisely, the map $\boldsymbol{y} \mapsto \mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y})$. We now recall this notion and some of its key properties.

Conditional expectation is defined as the orthogonal projection from the Lebesgue space $L^2(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ onto the subspace $L^2(\Omega, \sigma(\boldsymbol{y}), \mathbb{P}; \mathbb{R}^n)$ where $\sigma(\boldsymbol{y})$ is the smallest σ-algebra containing all preimages $\boldsymbol{y}^{-1}(A_{\boldsymbol{y}})$ where $A_{\boldsymbol{y}} \in \mathcal{B}(\mathbb{R}^n)$. Recall that the preimage is a map $\boldsymbol{y}^{-1} : \mathcal{B}(\mathbb{R}^n) \to \mathcal{F}$ between σ-algebras. This definition is reasonably intuitive: we can think of this as projecting onto the subspace induced by all collections of random numbers in Ω which play a role in determining what $\boldsymbol{y}$ does.

Recall that $L^2(\Omega, \mathcal{F}, \mathbb{P}; \mathbb{R}^n)$ is the Hilbert space of equivalence classes of random variables with inner product given by $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \mathbb{E}(\boldsymbol{a} \cdot \boldsymbol{b})$. Using this, it follows from the Projection Theorem for Hilbert spaces that the terms $\mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y})$ and $(\boldsymbol{\theta} - \mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y}))$ are uncorrelated—and, in the Gaussian case, that they are independent. This gives us the candidate random variables to use for $\boldsymbol{b}$ and $\boldsymbol{c}$, if we choose conditional expectation for f.

Finally, recall that for multivariate Gaussians, the conditional expectation is given by $\mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \boldsymbol{\mu}_{\boldsymbol{\theta}} + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}} \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{y}})$, where we note in our setting that the inverse always exists by non-singularity of $\boldsymbol{y}$. With these preparations, we are ready to revisit Theorem 2.2.

Matheron's update rule

Theorem 2.2. *For $\boldsymbol{\theta}, \boldsymbol{y}$ defined in Proposition 2.1, we have that*

$$(\boldsymbol{\theta} \mid \boldsymbol{y})(\omega, \boldsymbol{\gamma}) = \boldsymbol{\theta}(\omega) + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}} \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1} (\boldsymbol{\gamma} - \boldsymbol{y}(\omega)). \tag{2.4}$$

Proof. Write

$$\boldsymbol{\theta} = \mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y}) + (\boldsymbol{\theta} - \mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y})) \tag{2.15}$$

and note that since $\mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y})$ and $(\boldsymbol{\theta} - \mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y}))$ are uncorrelated and jointly Gaussian, they are independent. Applying Lemma 2.3 yields

$$\begin{aligned} (\boldsymbol{\theta} \mid \boldsymbol{y})(\omega, \boldsymbol{\gamma}) &= \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}} \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1} \boldsymbol{\gamma} + (\boldsymbol{\theta}(\omega) - \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}} \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1} \boldsymbol{y}(\omega)) && (2.16) \\ &= \boldsymbol{\theta}(\omega) + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}} \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1} (\boldsymbol{\gamma} - \boldsymbol{y}(\omega)) && (2.17) \end{aligned}$$

where we have substituted $\mathbb{E}(\boldsymbol{\theta} \mid \boldsymbol{y}) = \boldsymbol{\mu}_{\boldsymbol{\theta}} + \boldsymbol{\Sigma}_{\boldsymbol{\theta}\boldsymbol{y}} \boldsymbol{\Sigma}_{\boldsymbol{y}\boldsymbol{y}}^{-1} (\boldsymbol{y} - \boldsymbol{\mu}_{\boldsymbol{y}})$ and immediately cancelled the mean terms. The claim follows. ■

From Theorem 2.2, we obtain a second way of representing multivariate Gaussian conditionals. This entails two steps: (i) sample $\boldsymbol{\theta}, \boldsymbol{y}$ jointly, and (ii) transform $\boldsymbol{\theta}, \boldsymbol{y}$ into $\boldsymbol{\theta} \mid \boldsymbol{y} = \boldsymbol{\gamma}$ by employing the given expression. For an illustration of this procedure, see Figure 2.2.

Remarkably, this result is seemingly missing from every machine learning textbook on Gaussian processes in widespread use, and appears almost entirely unknown within the field. It's possible this is because the expression's computational costs are cubic in the combined dimension, which is more expensive than the previous costs. While this holds for general Gaussians, we show it can be avoided for many cases of practical interest.

On the other hand, Theorem 2.2 is certainly known in other communities. In a tribute to Georges Matheron, who pioneered the expression's use in geostatistics, Chilès and Lantuéjoul [22] say that:

> *[Matheron's update rule] is nowhere to be found in Matheron's entire published works, as he merely regarded it as an immediate consequence of the orthogonality of the [conditional expectation] and the [residual process].*

More recently, Doucet [32] describes the algorithm in a technical report which begins with the remark:

> *This note contains no original material and will never be submitted anywhere for publication. However, it might be of interest to people working with [Gaussian processes] so I am making it publicly available.*

Additionally, Theorem 2.2 is reasonably well-known in geostatistics [58, 30, 34, 81]. In parallel, these ideas were rediscovered in the astrophysics community, with Hoffman and Ribak [52] describing approximations similar in spirit to the ones we study in the sequel.

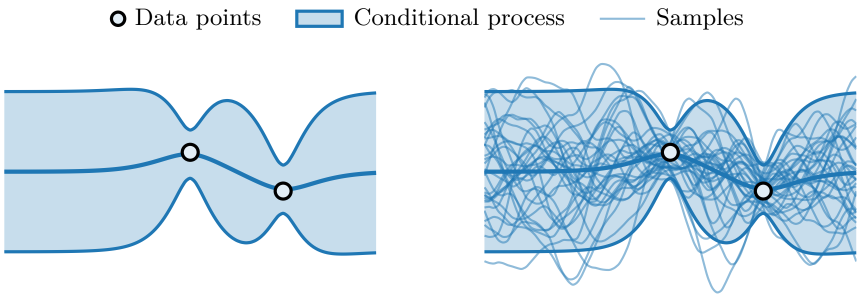
The present state of affairs therefore seems to be that a small set of technical experts are aware of Theorem 2.2 but believe it to be too trivial to write about, while practitioners working in areas such as Bayesian optimization do not know that it exists. While for multivariate Gaussians the result certainly is trivial, we subsequently show that using it in the right manner yields significant progress towards resolving certain issues in decision-making settings. To do so, we now consider Gaussian processes.

2.2. Conditioning Gaussian processes

We now study conditioning in Gaussian processes. For this, we use Rasmussen and Williams [89] as our canonical reference. In what follows, we develop and showcase the two points of view—distributional and pathwise—introduced in the Gaussian process setting. We begin with the former.

2.2.1. Distributional conditioning. The standard way of representing Gaussian process conditionals is to use the finiteness of the data to pick a suitable set of locations and work with finite-dimensional marginals. Since conditioning and marginalization commute, conditioning the Gaussian process thus reduces to conditioning multivariate Gaussians. We now describe this.

To ease notation, we now set the prior mean to zero: the general case can be recovered by adding and subtracting mean functions. Additionally, in what follows, we let $\boldsymbol{x} = (x_1, .., x_n)$ denote the data locations. For this usage specifically, we use bold italics to indicate a product structure rather than a linear structure, and in particular do not require X to be a vector space. Finally, functions applied to product spaces are understood componentwise. With this notation in place, we are ready to state the claim.



(a) Calculate conditional (b) Draw samples

Figure 2.3. Illustration of distributional conditioning of a Gaussian process. Here, we calculate the conditional distribution at a finite set of locations, and then draw samples from a multivariate Gaussian, obtaining the value of the posterior Gaussian process at a set of points. This view of conditioning is called *distributional*, since random variables appear only at the very end of the algorithmic process.

Posterior Gaussian process

Proposition 2.4. *The Bayesian model*

$$\boldsymbol{y} \mid f \sim \mathrm{N}(f(\boldsymbol{x}), \mathbf{\Sigma}) \qquad\qquad f \sim \mathrm{GP}(0, k) \tag{2.18}$$

admits the Gaussian process

$$(f \mid \boldsymbol{y})(\cdot, \boldsymbol{\gamma}) \sim \mathrm{N}(\mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}+\mathbf{\Sigma})^{-1}\gamma, \mathbf{K}_{(\cdot,\cdot)}-\mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}+\mathbf{\Sigma})^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}) \tag{2.19}$$

as its posterior.

Proof. Apply Proposition 2.1 to a set of finite-dimensional marginals. ■

This result, illustrated in Figure 2.3, gives us a way of carrying out Bayesian learning using Gaussian processes, given a finite set of data. Note the *bottom-up* nature of this perspective: we describe the posterior Gaussian process—which is an actual *process* defined everywhere—using its posterior finite-dimensional marginals.

We now consider computational costs of the above formula. Calculating posterior moments clearly entails cubic costs with respect to the data size n, owing the need to invert $n \times n$ matrices. Now, suppose we are are interested in computing quantities involving the posterior distribution. Consider for instance computing the Thompson sampling acquisition function considered in Section 1.2.3, which we recall is

$$x_{t+1}(\omega) = \operatorname*{arg\,min}_{x \in X} \alpha_t(\omega, x) \qquad\qquad \alpha_t \sim f \mid \boldsymbol{y}. \tag{2.20}$$

Given the minimization involved in this objective, there is no chance in finding an analytic expression for x_{t+1}, and we must resort to numerical methods. Just about any numerical procedure one can imagine—for instance, gradient descent—will involve drawing random samples from $f \mid \boldsymbol{y}$ at different locations, and performing the necessary algorithmic operations on them. Summarizing, the computational costs of this expression are as follows.

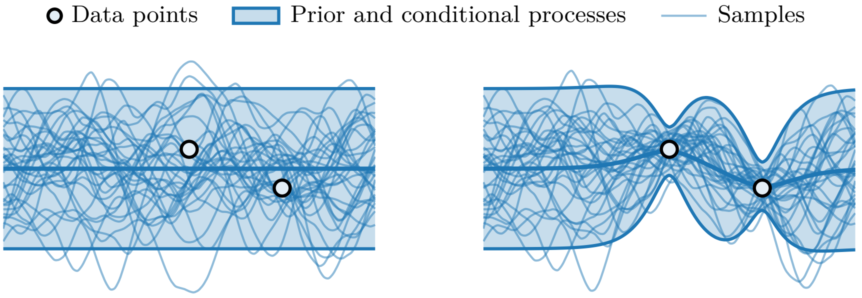
1. Data: $\mathcal{O}(n^3)$ where n is the size of the training set, due to the intermediate term $(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma})^{-1}$ in the posterior covariance.

2. Predictions: $\mathcal{O}(n_*^3)$ where n_* is the number of locations the posterior Gaussian process needs to be jointly evaluated at, due to the need to factorize the posterior covariance $\mathbf{K}_{(\cdot,\cdot)} - \mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma})^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}$.

This becomes more difficult if one considers evaluation locations that are not known in advance, and might be determined using previous points. Due to the need to iteratively re-condition on sampled process values and factorize matrices at every intermediate computation step, roundoff errors accumulate as computations proceed. Thus, even if we are willing to pay cubic costs, we then face the secondary issue of numerical instability. The situation if one needs to differentiate through objectives such as α_t is even worse.

Without additional considerations, these costs are disastrous, and illustrate typical difficulties in building practical decision-making systems powered by Gaussian processes. Fortunately, a wide variety of techniques to deal with them are available: in particular, *inducing point* methods provide a broad set of approximations for reducing the $\mathcal{O}(n^3)$ costs. We will complement these ideas by introducing techniques to tackle the $\mathcal{O}(n_*^3)$ costs. For this, we proceed to develop a pathwise view of conditioning.

2.2.2. Pathwise conditioning. Given the pathwise view of conditioning multivariate Gaussians given by Matheron's update rule, one can ask: is there an analogous statement for Gaussian processes? Does the purely distribution notion of a Gaussian conditional have an analogous description in terms of random functions? We answer this affirmatively below.



(a) Sample from prior (b) Transform into conditional

Figure 2.4. Illustration of pathwise conditioning of a Gaussian process. Here, we first sample a set of random functions from the prior, along with random noise variables at each of the data locations, then transform these into a samples from the posterior distribution. This view of conditioning is termed *pathwise*, since it is defined directly at the level of random functions.

Posterior Gaussian process (pathwise)

Corollary 2.5. *For $\boldsymbol{y} \mid f$, f, and $\boldsymbol{f}$ defined in Proposition 2.4, we have*

$$(f \mid \boldsymbol{y})(\omega, \boldsymbol{\gamma}) = f(\omega, \cdot) + \mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma})^{-1}(\boldsymbol{\gamma} - f(\omega, \boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega)) \qquad (2.21)$$

where $\boldsymbol{\varepsilon} \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$.

Proof. Apply Theorem 2.2 to a set of finite-dimensional marginals. ■

Note that in this expression, the prior is evaluated jointly at all locations—the random variables $f(\omega, \cdot)$ and $f(\omega, \boldsymbol{x})$ are *dependent*. Similarly, equality holds in distribution, because the random variable $(f \mid \boldsymbol{y})(\cdot, \boldsymbol{\gamma})$ is only defined in distribution to begin with.

Corollary 2.5 gives an alternative way of representing posterior Gaussian processes: (i) sample the prior and all auxillary random variables such as the noise term $\boldsymbol{\varepsilon}$, and (ii) transform the sampled function to form the posterior as a random function. This is shown in Figure 2.4.

This strategy can be carried out if we know the locations we wish to evaluate the posterior at. In this case, we sample the prior at the data and evaluation locations jointly, and transform the resulting samples into posterior samples. Examining the computational costs, we see these are $\mathcal{O}(n^3)$ with respect to data size, and $\mathcal{O}(n_*^3)$ with respect to the number of evaluation locations. At this stage, then, we have seemingly only gained numerical stability.

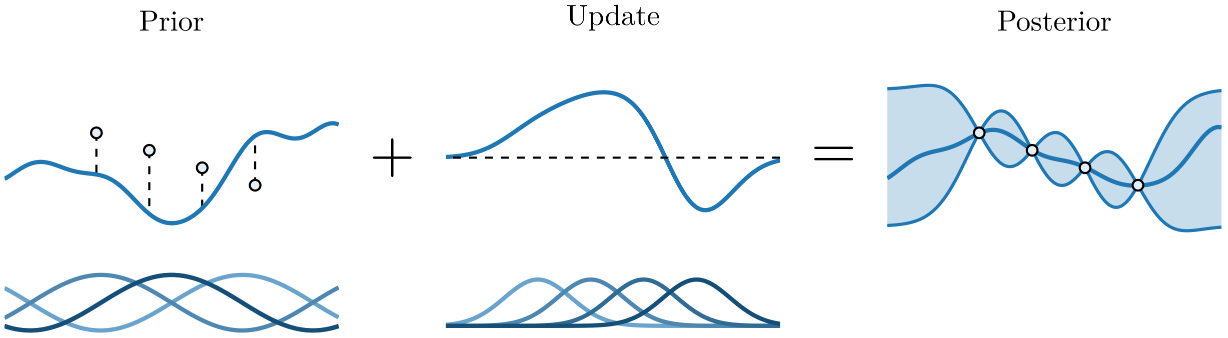


Figure 2.5. Approximate pathwise conditioning, with bases on bottom row.

Corollary 2.5, however, is not merely a computational result: it gives us a powerful way of thinking about posterior Gaussian processes. In particular, we can use the point of view it offers to construct posterior approximations. Observe that the cubic costs $\mathcal{O}(n_*^3)$ occur entirely due to the need to jointly sample the prior at all evaluation locations. Suppose that we can approximately express the prior using a set of *finite basis functions* as

$$f(\omega,\cdot) \approx \tilde{f}(\omega,\cdot) = \underbrace{\sum_{i=1}^{\ell} w_i(\omega)\phi_i(\cdot)}_{\text{finite basis functions}} \qquad w_i \sim \mathrm{N}(0,1). \tag{2.22}$$

We can substitute this approximation into Corollary 2.5 to obtain

$$(f \mid \boldsymbol{y})(\omega,\boldsymbol{\gamma}) \approx \tilde{f}(\omega,\cdot) + \mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \mathbf{\Sigma})^{-1}(\boldsymbol{\gamma} - \tilde{f}(\omega,\boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega)). \tag{2.23}$$

Once the random weights $\boldsymbol{w}$ are sampled, the posterior becomes a deterministic function, which can be evaluated at $\mathcal{O}(n_*)$ costs. Thus, under this class of approximations, our cubic costs with respect to the number of evaluation locations become *linear*. We show in the sequel that for appropriate choices of $\tilde{f}$, such approximations can achieve excellent error control, ensuring they perform effectively in practice.

We now examine a different aspect of the pathwise representation of posterior Gaussian processes: the role of the data. By re-expressing the matrix-vector product of the kernel matrix term $\mathbf{K}_{(\cdot)\boldsymbol{x}}$ as a sum, we obtain

$$(f \mid \boldsymbol{y})(\omega,\boldsymbol{\gamma}) = f(\omega,\cdot) + \underbrace{\sum_{j=1}^{n} v_j(\omega)k(x_j,\cdot)}_{\text{canonical basis functions}} \tag{2.24}$$

where $\boldsymbol{v} = (\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma})^{-1}(\boldsymbol{\gamma} - f(\omega, \boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega))$. This identifies the effect of the data in a posterior Gaussian process as the addition of *canonical basis functions* $k(x_i, \cdot)$ to the prior. We explore this interpretation further in what follows. From this viewpoint, if we define $\tilde{\boldsymbol{v}}$ analogously to $\boldsymbol{v}$, but with f replaced with $\tilde{f}$, our approximate posterior is

$$(f \mid \boldsymbol{y})(\omega, \boldsymbol{\gamma}) \approx \sum_{i=1}^{\ell} w_i(\omega)\phi_i(\cdot) + \sum_{j=1}^{n} \tilde{v}_j(\omega)k(x_j, \cdot) \tag{2.25}$$

which shows that our proposed approximation involves writing the posterior Gaussian process of interest as a sum of *two* finite sets of basis functions—one for the prior, another for the data. The basis coefficients w_i and $\tilde{v}_j$ here are dependent random variables. We illustrate the overall approximation in Figure 2.5. Summarizing, pathwise representations of posterior Gaussian processes provide a useful framework for constructing posterior approximations which are actual random functions.

In Bayesian optimization, such approximations offer a promising avenue to avoid the computational difficulties that result from working with finite-dimensional marginals. Computing the Thompson sampling acquisition function

$$x_{t+1}(\omega) = \operatorname*{arg\,max}_{x \in X} \alpha_t(\omega, x) \qquad\qquad \alpha_t \sim f \mid \boldsymbol{y} \tag{2.26}$$

or any other quantities that require evaluating the Gaussian process at arbitrary locations can be done using a simple Monte Carlo approach as follows.

1. Sample the random weights $(\boldsymbol{w}, \boldsymbol{v})$ to form the approximate posterior.
2. Maximize the approximate posterior using any numerical procedure.

The key advantage of this approach is that once the random weights are sampled, the posterior—a random function—effectively becomes deterministic. This allows us to not only evaluate it in linear time, but also to differentiate through posterior samples using automatic differentiation—here, enabling us to maximize them using gradient descent without computing gradient processes or employing special considerations of any kind.

In total, we obtain a simple, accurate, and efficient way to compute acquisition functions such as Thompson sampling. To obtain a complete algorithm, all that remains is to find finite basis approximations to Gaussian process priors: this will require additional structure present in specific classes of kernels. We proceed to explore a number of techniques for doing so.

2.3. Sampling from prior Gaussian processes

In the preceding section, we explored a class of approximate posterior Gaussian processes constructed by plugging a finite-basis-function-based approximate prior into the pathwise update. Such approximate priors can be constructed in many ways, including by expressing the true prior within a basis of an appropriate space of functions, and truncating the resulting infinite sum. Different choices of bases will result in different applicability, ease of use, and approximation error. We now explore possible choices.

2.3.1. Random feature methods. One of the most popular classes of kernels is the class of stationary kernels defined on Euclidean spaces, which are kernels that factorize through a one-argument function depending only on the difference between points. For such kernels, random feature methods, originally proposed by Rahimi and Recht [87], can be used to construct approximate priors whose properties make them a particularly attractive choice. We examine these now.

To construct a basis function expansion, our strategy will be to find a *feature map* $\varphi : X \to H_k$ that maps states into vectors in the *reproducing kernel Hilbert space* induced by k. This is defined as follows.

Reproducing kernel Hilbert space

Definition 2.6. *Let X be a set, and let $H \subseteq \mathbb{R}^X$ be a Hilbert space of functions. We say that H is a* REPRODUCING KERNEL HILBERT SPACE *if, for any $x \in X$, we have $\operatorname{ev}_x \in H^*$ where $\operatorname{ev}_x : H \to \mathbb{R}$ is called the* EVALUATION MAP *and is defined by $\operatorname{ev}_x f = f(x)$.*

Note that in this definition, the statement $\operatorname{ev}_x \in H^*$ means that ev_x is a bounded linear functional. Reproducing kernel Hilbert spaces therefore impose continuity requirements on their pointwise evaluation functionals.

Ostensibly, this definition has nothing to do with kernels, and it is unclear what a reproducing kernel Hilbert space *induced by k* actually means. A consequence of the above definition is that given a reproducing kernel Hilbert space H, we can define the function $k_H(x, x') = \langle \Psi_H^{-1} \operatorname{ev}_x, \Psi_H^{-1} \operatorname{ev}_{x'} \rangle$, called the *reproducing kernel*, where $\Psi_H : H \to H^*$ is the bijective linear isometry given by the Riesz Representation Theorem. It is easy to see that k_H is positive semi-definite. It turns out a converse statement also holds.

Reproducing kernel

Moore–Aronszajn Theorem

Result 2.7. *Let $k : X \times X \to \mathbb{R}$ be a symmetric positive semi-definite kernel. Then there is a unique Hilbert space $H_k \subseteq \mathbb{R}^X$ of real-valued functions for which k is the reproducing kernel.*

Proof. Paulsen and Raghupathi [84], Proposition 2.13 and Theorem 2.14. ■

This gives another point of view from which one can study and understand kernels and, by proxy, Gaussian processes. The reproducing kernel Hilbert space induced by the covariance kernel of a Gaussian process is sometimes called its *Cameron–Martin space* or *native space*, and encodes its mathematical properties. This perspective can be studied abstractly, leading to the theory of *isonormal Gaussian processes*. See Wendland [119], Lifshits [71], and Le Gall [65] for details on these and related ideas.

Feature map

We will need one final notion. We say that a function $\varphi : X \to H_k$ is a *feature map* if $k(x, x') = \langle \phi(x), \phi(x') \rangle$. Now, we introduce the key idea for constructing our approximate Gaussian prior: suppose we have a *finite-dimensional* approximation for such a feature map, namely a vector-valued function $\boldsymbol{\phi} : X \to \mathbb{R}^\ell$ such that $\boldsymbol{\phi}(x)^T \boldsymbol{\phi}(x') \approx \langle \varphi(x), \varphi(x') \rangle$. Then

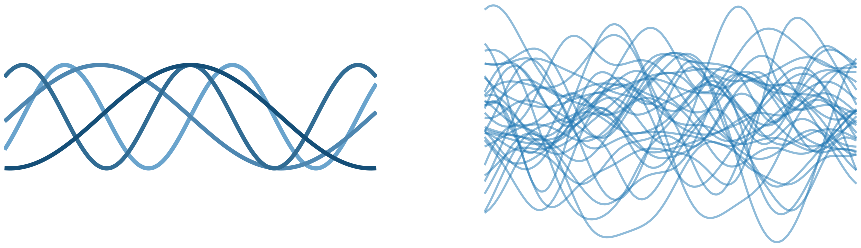
$$\tilde{f}(\omega, \cdot) = \sum_{i=1}^{\ell} w_i(\omega) \phi_i(\cdot) \qquad\qquad w_i \sim \mathrm{N}(0, 1) \tag{2.27}$$

by direct calculation has covariance approximately equal to that of f. An example can be seen in Figure 2.6. Therefore, to construct an approximate prior, it suffices to find a finite-dimensional approximate feature map.

For stationary kernels, techniques for constructing approximate feature maps are well-studied, originally motivated by questions arising in kernel support vector machines. We now introduce the *random Fourier feature* method for constructing such maps, beginning with a brief description of the stationary setting.

Stationary kernel

Let $X = \mathbb{R}^d$. In the remaining subsection, we use bold italic $\boldsymbol{x}$ to denote linear rather than product structure. We say that a kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ is called *stationary* if $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x} - \boldsymbol{x})$ for a function $k : \mathbb{R}^d \to \mathbb{R}$. A stationary kernel, then, is a two-argument positive definite function which factorizes through a one-argument function depending only on the difference between two points. Note that such a kernel is invariant under translation and can be characterized as such. We will need a result known as *Bochner's Theorem*.



(a) Fourier basis functions (b) Approximate prior samples

Figure 2.6. Illustration of random Fourier feature methods for sampling from approximate priors. Here, we show a small subset of randomly sampled Fourier basis functions, along with approximate prior samples constructed using the random Fourier basis functions.

Bochner's Theorem

Result 2.8. *For every stationary continuous positive definite kernel with $k(\boldsymbol{x},\boldsymbol{x}) = 1$ there is a symmetric probability measure ρ on $\mathbb{R}^d$ which we call the* SPECTRAL MEASURE *of k. Moreover, k admits the representation*

$$k(\boldsymbol{x}-\boldsymbol{x}') = \int_{\mathbb{R}^d} e^{2\pi i \boldsymbol{\varpi}^T(\boldsymbol{x}-\boldsymbol{x}')} \,\mathrm{d}\rho(\boldsymbol{\varpi}). \tag{2.28}$$

Conversely, every probability measure on $\mathbb{R}^d$ gives rise to such a kernel via its Fourier transform.

Proof. Paulsen and Raghupathi [84], Theorem 10.4. ■

Here, *symmetry* of ρ refers to invariance under reflection about the origin. This result is true more generally if X is replaced by a locally compact Abelian group, for which the corresponding spectral measure will be supported on the Pontryagin dual group—see Paulsen and Raghupathi [84]. We omit this level of generality because we will not need it. We do briefly note, however, that this means that spectral measures of kernels on compact spaces are *discrete*—this behavior will reappear under a different guise in Chapter 3.

From here, it is clear that the feature map we seek can be constructed by Monte Carlo approximation of the integral representation given by Bochner's Theorem. To maintain order, we introduce a second probability space $(\Xi, \mathcal{G}, \mathbb{Q})$ to distinguish the stochasticity associated with the random feature expansion from stochasticity associated with the Gaussian process. Letting $k(\boldsymbol{x},\boldsymbol{x}) = \sigma^2$, write

$$
\begin{aligned}
k(\boldsymbol{x} - \boldsymbol{x}') &= \sigma^2 \int_{\mathbb{R}^d} e^{2\pi i \boldsymbol{\varpi}^T (\boldsymbol{x} - \boldsymbol{x}')} \,\mathrm{d}\rho(\boldsymbol{\varpi}) && (2.29) \\
&= \sigma^2 \int_{\mathbb{R}^d} e^{2\pi i \boldsymbol{\varpi}^T \boldsymbol{x}} \overline{e^{2\pi i \boldsymbol{\varpi}^T \boldsymbol{x}'}} \,\mathrm{d}\rho(\boldsymbol{\varpi}) && (2.30) \\
&\approx \frac{\sigma^2}{\ell} \sum_{j=1}^{\ell} e^{2\pi i \boldsymbol{\varpi}_j(\xi)^T \boldsymbol{x}} \overline{e^{2\pi i \boldsymbol{\varpi}_j(\xi)^T \boldsymbol{x}'}} && (2.31) \\
&= \frac{\sigma^2}{\ell} \sum_{i=1}^{\ell} \begin{aligned} &\cos(2\pi \boldsymbol{\varpi}_i(\xi)^T \boldsymbol{x}) \cos(2\pi \boldsymbol{\varpi}_i(\xi)^T \boldsymbol{x}') \\ &+ \sin(2\pi \boldsymbol{\varpi}_i(\xi)^T \boldsymbol{x}) \sin(2\pi \boldsymbol{\varpi}_i(\xi)^T \boldsymbol{x}') \end{aligned} && (2.32) \\
&= \boldsymbol{\phi}(\xi, \boldsymbol{x})^T \boldsymbol{\phi}(\xi, \boldsymbol{x}') && (2.33)
\end{aligned}
$$

where $\overline{\cdot}$ denotes complex conjugation, and

$$
\phi_i(\xi, \boldsymbol{x}) = \left[\begin{array}{ll} \frac{\sigma}{\sqrt{\ell}} \cos(2\pi \boldsymbol{\varpi}_i(\xi)^T \boldsymbol{x}), & i \text{ odd} \\ \frac{\sigma}{\sqrt{\ell}} \sin(2\pi \boldsymbol{\varpi}_{i-1}(\xi)^T \boldsymbol{x}), & i \text{ even} \end{array} \right. \tag{2.34}
$$

which, for ℓ even, gives our approximate prior, shown in Figure 2.6 as

$$
\tilde{f}(\cdot) = \sum_{i=1}^{\ell} w_i(\omega) \phi_i(\xi, \cdot) \qquad w_i \sim \mathrm{N}(0, 1) \qquad \boldsymbol{\varpi}_i \sim \rho. \tag{2.35}
$$

One remarkable property of random feature methods is that their approximation error decays at a *dimension-free* rate in the Monte Carlo sense: for details, see Sutherland and Schneider [109]. This limits the effect of the curse of dimensionality to constant factors. Better yet, random feature methods are well-understood owing to their widespread use in other areas such as support vector machines [87, 73]. For this reason, random feature methods are often the technique of choice for stationary Euclidean kernels.

A number of different random Fourier feature approximations have been proposed in the literature [87, 125, 24, 73], along with techniques for their theoretical analysis [109, 107, 23, 70, 73]. For a comprehensive review of random feature methods, see Liu et al. [73]. We now consider other classes of approximations.

2.3.2. Karhunen–Loève expansions. Among all techniques for constructing approximate priors, different techniques will generally yield different amounts of error for the same number of basis functions. One can then ask:

is there an optimal choice? Of course, the answer to this question will depend on what one actually means by the word *optimal.*

If we take $X \subset \mathbb{R}^d$ to be compact, and we use expected mean squared error—which we recall is equivalent to the $L^2(\Omega; L^2(X;\mathbb{R}))$ norm—as our notion of optimality, one can affirmatively answer the above question. Recall that a continuous kernel $k : X \times X \to \mathbb{R}$ induces a covariance operator

$$\mathcal{K} : L^2(X;\mathbb{R}) \to L^2(X;\mathbb{R}) \qquad \mathcal{K} : \phi \mapsto \int_X \phi(x)k(x,\cdot)\,\mathrm{d}x \tag{2.36}$$

where by writing $\|\mathcal{K}\phi\|_{L^2(X;\mathbb{R})} \leq \operatorname{vol}(X)\|k\|_{C^0(X\times X;\mathbb{R})}\|\phi\|_{L^2(X;\mathbb{R})}$ using compactness and boundedness of k we affirm correctness of the operator's domain and range. By compactness, $\mathcal{K}$ will admit a countable set of eigenvalues and eigenfunctions. It turns out, that, by the Karhunen–Loève Theorem, the Gaussian process itself can be written in terms of the same eigenvalues and eigenfunctions as well.

Karhunen–Loève Theorem

Result 2.9. *Let $X \subseteq \mathbb{R}^d$ be compact, and let f be a Gaussian process with continuous covariance function. Then we have*

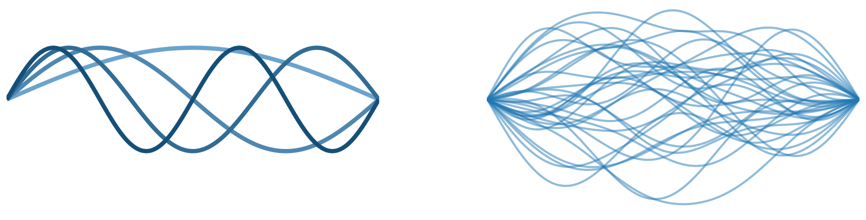
$$f(\omega,\cdot) = \sum_{i=1}^{\infty} w_i(\omega)\phi_i(\cdot) \qquad w_i \sim \mathrm{N}(0,1) \tag{2.37}$$

where convergence holds almost surely, and ϕ_i are an orthogonal basis on $L^2(X;\mathbb{R})$ given by rescaled eigenfunctions of the covariance operator. Moreover, for every $\ell \in \mathbb{N}$, truncating this series yields an $L^2(\Omega; L^2(X;\mathbb{R}))$-optimal approximation among all ℓ-term sums of $L^2(X;\mathbb{R})$-orthogonal functions.

Proof. Lord et al. [74], Theorem 5.28, as well as Ghanem and Spanos [42], Section 2.3.2. ■

More generally, an analogous result, albeit with a weaker form of convergence, also holds for general *square-integrable* stochastic processes which may be non-Gaussian. Since we will not consider such processes, we do not pursue this direction here.

For a general kernel, finding the eigenfunctions of the covariance operator may be difficult. In many cases, however, they can be understood by utilizing additional structure present in the problem. For example Solin and Kok [102] study certain classes of Gaussian processes constructed to satisfy boundary conditions. Here, the eigenfunctions of the covariance operator coincide with



(a) Karhunen–Loève basis (b) Approximate prior samples

Figure 2.7. Illustration of a Karhunen–Loève expansion for a boundary-constrained squared exponential kernel on the unit interval. For this kernel, the basis functions coincide with the eigenfunctions of the Dirichlet Laplacian. We show the first four eigenfunctions, along with approximate prior samples using the first 87 terms in the expansion. This truncation was chosen because the remaining terms are zero in floating-point arithmetic, highlighting the approximation's efficiency.

the eigenfunctions of a boundary-constrained Laplacian, allowing them to be obtained numerically. These are shown in Figure 2.7.

Kernels induced by eigenvalues and eigenfunctions of appropriately-defined Laplace operators are also a common tool in non-Euclidean settings [103, 104, 25], and we explore them further in Chapter 3. For the moment, however, we consider a third class of approximations.

2.3.3. Finite element methods. We now introduce the third class of approximations we will consider: those constructed via *finite element approximations* of solutions of stochastic partial differential equations [74, 75, 60]. Many Gaussian process priors, including the widely-used Matérn class, following Whittle [120, 121] and Lindgren et al. [72], can be expressed in such a manner. Such Gaussian processes are also of direct interest in the non-Euclidean settings of Chapter 3.

Let X be a subset of a Euclidean space, or a Riemannian manifold. Suppose our Gaussian process f satisfies the equation

$$\mathcal{L}f = \mathcal{W} \tag{2.38}$$

where $\mathcal{L} : H \to L^2(X;\mathbb{R})$ is a bounded linear operator acting on a certain reproducing kernel Hilbert space which uniquely determines the Gaussian process, and $\mathcal{W}$ is a white noise process over $L^2(X;\mathbb{R})$—a random distribution whose properties are determined by the Lebesgue space. Slightly more precisely, the equation is meant as either an almost sure or a distributional equality

$$f(\omega, \mathcal{L}^* h) = \mathcal{W}(\omega, h) \tag{2.39}$$

between generalized Gaussian fields—a formal treatment is given in Chapter 3. Assume that the generalized Gaussian field $f : \Omega \times H \to \mathbb{R}$ can be written as the integral of a Gaussian process $f : \Omega \times X \to \mathbb{R}$ as

$$f(\omega, h) = \int_X f(\omega, x) h(x) \, \mathrm{d}x. \tag{2.40}$$

This amounts to requiring the stochastic partial differential equation's solution f to possess some regularity, and in particular on compact domains X follows from sample-continuity of f. If we define the bilinear form

$$a(\phi, \psi) = \int_X \phi(x) (\mathcal{L}^* \psi)(x) \, \mathrm{d}x \tag{2.41}$$

then our stochastic partial differential equation becomes

$$a(f(\omega, \cdot), h) = \mathcal{W}(\omega, h) \tag{2.42}$$

which is an equation between a Gaussian process, bilinear form, and random linear functional. Now, we introduce approximations: suppose that $f(\omega, x) \approx \tilde{f}(\omega, x) = \sum_{i=1}^{\ell} w_i(\omega) \phi_i(x)$ and that $h(x) \approx \sum_{j=1}^{\ell} v_j \psi_j(x)$. Plugging this in and differentiating to remove the coefficients v_j yields the system of equations

$$\sum_{i=1}^{\ell} w_i(\omega) a(\phi_i, \psi_j) = \mathcal{W}(\omega, \psi_j) \tag{2.43}$$

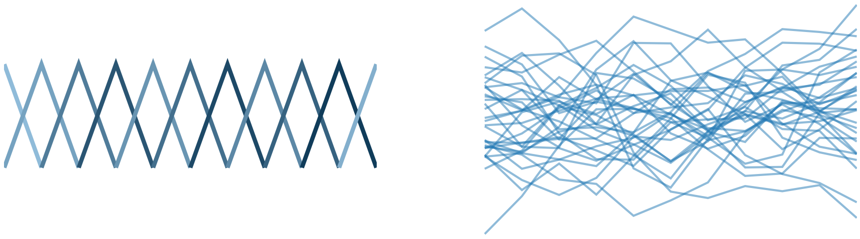
which by defining $A_{ij} = a(\phi_i, \psi_j)$ and $b_j(\omega) = \mathcal{W}(\omega, \psi_j)$ can be recognized as a random linear system, more compactly written

$$\mathbf{A} \boldsymbol{w}(\omega) = \boldsymbol{b}(\omega). \tag{2.44}$$

If we let $\mathbf{M} = \operatorname{Cov}(\boldsymbol{b})$ with $\operatorname{Cov}(b_i, b_j) = \langle \psi_i, \psi_j \rangle$—where we note that since $\mathcal{W}$ is a white noise process, the matrix $\mathbf{M}$ coincides with the finite-element mass matrix—we obtain the distribution for the basis coefficients. Our approximate prior can therefore be written

$$\tilde{f}(\omega, x) = \sum_{i=1}^{\ell} w_i(\omega) \phi_i(x) \qquad \boldsymbol{w} \sim \mathrm{N}(\mathbf{0}, \mathbf{A}^{-1} \mathbf{M} \mathbf{A}^{-T}) \; . \tag{2.45}$$

What is particularly powerful about this technique is that it gives a significant degree of freedom for what kinds of finite sets of basis functions we can choose



(a) Finite element basis (b) Approximate prior samples

Figure 2.8. Illustration of finite element approximate prior samples for a Matérn kernel with smoothness 3/2 and length scale κ. In one dimension, the bilinear form for this kernel is $a(f,h) = \int_X \frac{3}{\kappa^2} f(x)h(x) + \nabla f(x) \cdot \nabla h(x) \,\mathrm{d}x$. Since this bilinear form is of first order in both arguments, we employ a piecewise linear finite element basis consisting of compactly supported triangle-shaped functions, using it to represent both f and h. This results in matrices $\mathbf{A}$ and $\mathbf{M}$ which are symmetric tridiagonal.

for ϕ_i and ψ_j. If $\mathcal{L}$ is a differential operator of sufficiently low order, a fruitful choice, shown in Figure 2.8 for the model studied by Lindgren et al. [72], is to take ϕ_i to be compactly supported piecewise linear functions, since this will cause the matrices $\mathbf{A}$ and $\mathbf{M}$ to be sparse. This can enable one to use a much larger number of basis functions compared to alternative methods.

The main issue with finite element methods is that they typically demand more from practitioners compared to alternatives. In particular, one usually needs to understand the stochastic partial differential equations governing the Gaussian process of interest with a reasonable degree of detail to know how to choose basis functions well. Additionally, sparse linear algebra in a parallel environment with automatic differentiation can be cumbersome.

This concludes our presentation of techniques for constructing approximate priors. In general, the best choice for a particular application will depend on the detailed requirements of the situation. We now proceed to study other aspects of the pathwise viewpoint.

2.4. Approximating pathwise data-dependent terms

In the preceding section, we discussed techniques for approximating the prior using a finite set of basis functions with random coefficients. This enabled us to construct approximate pathwise representations of posterior Gaussian processes with $\mathcal{O}(n_*)$ computational complexity, where we recall again that n_* is the number of points where we wish to evaluate the posterior. We now consider the other computational costs in the formula: $\mathcal{O}(n^3)$, where n is the size of the training data. These costs arise because computing

$$(f \mid \boldsymbol{y})(\omega, \boldsymbol{\gamma}) = f(\omega, \cdot) + \mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma})^{-1}(\boldsymbol{\gamma} - f(\omega, \boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega)) \tag{2.46}$$

requires us to invert the $n \times n$ matrix $\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \boldsymbol{\Sigma}$. We now ask: from a pathwise perspective, what techniques are available for reducing these costs?

2.4.1. Inducing points. Consider a simple approach to reducing the above cubic costs: instead of conditioning the Gaussian process on the full data $(x_1, y_1), .., (x_n, y_n)$, find a different data set $(z_1, \mu_1), .., (z_m, \mu_m)$, for which $m \ll n$ and yet the posterior is approximately the same. This idea underlies *inducing point* approximations [100, 114, 83, 49].

To ensure the approximation is expressive enough, we can either (i) make $\boldsymbol{\mu}$ random with a learnable mean and covariance, or, following Opper and Archambeau [83], (ii) introduce a learned noise covariance $\boldsymbol{\Lambda}$ rather than reusing the one from the original model. From a pathwise perspective, the latter approach gives

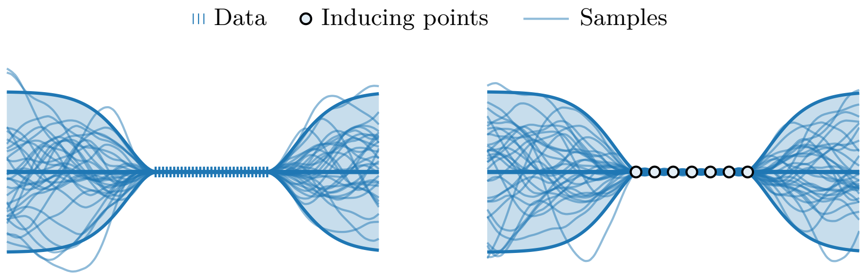
$$(f \mid \boldsymbol{u})(\omega, \boldsymbol{\mu}) = f(\omega, \cdot) + \mathbf{K}_{(\cdot)\boldsymbol{z}}(\mathbf{K}_{\boldsymbol{z}\boldsymbol{z}} + \boldsymbol{\Lambda})^{-1}(\boldsymbol{\mu} - f(\omega, \boldsymbol{z}) - \boldsymbol{\epsilon}(\omega)) \tag{2.47}$$

where $f \mid \boldsymbol{u}$ is the posterior under the modified likelihood employing the noise covariance $\boldsymbol{\Lambda}$. For $\boldsymbol{z}$, as with $\boldsymbol{x}$, we also use bold italics to denotes product structure. This approximation has a particularly simple interpretation: we can re-write it as

$$(f \mid \boldsymbol{u})(\omega, \boldsymbol{\mu}) = f(\omega, \cdot) + \underbrace{\sum_{j=1}^{m} v_j(\omega) k(z_j, \cdot)}_{\text{sparse basis functions}} \tag{2.48}$$

where $\boldsymbol{v}(\omega) = (\mathbf{K}_{\boldsymbol{z}\boldsymbol{z}} + \boldsymbol{\Lambda})^{-1}(\boldsymbol{\mu} - f(\omega, \boldsymbol{z}) - \boldsymbol{\epsilon}(\omega))$. Thus, we see that all we have done is replaced a large sum of data-dependent functions $k(x_j, \cdot)$ with $j = 1, .., n$ induced by the kernel with a much smaller sum involving a sparsified set of functions $k(z_j, \cdot)$ with $j = 1, .., m$ and $m \ll n$. This interpretation applies to most variational approximations, including the classical framework of Titsias [114], which can be seen in Figures 2.9 and 2.10.

Given such an approximate posterior, how should we select the introduced hyperparameters $\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$ to ensure quality? At minimum, we should attempt



(a) Posterior Gaussian process (b) Variational approximation

Figure 2.9. Inducing point approximation using a variation family built using randomized process values. Here, we use seven inducing points to represent the posterior distribution under thirty-one data points. The inducing point approximation approximates the posterior as accurately as possible, using sparsified Gaussian processes as the variational family.

to guarantee consistency of the approximation, in the sense that if $m = n$ then one can choose $\boldsymbol{z} = \boldsymbol{x}$, $\boldsymbol{\mu} = \boldsymbol{\gamma}$, and $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}$ to recover the desired posterior. We therefore seek to minimize some notion of *distance* on the space of probability measures.

The most natural choice one can consider is arguably the one that emerged from the analysis of Bayes' Rule presented in Chapter 1: namely, the *Kullback–Leibler divergence*. Minimizing this amounts to replacing the measure space in the variational formulation of Bayes' Rule of Proposition 1.6 with the parameterized subspace of measures induced by the set of approximate posteriors with different parameter values

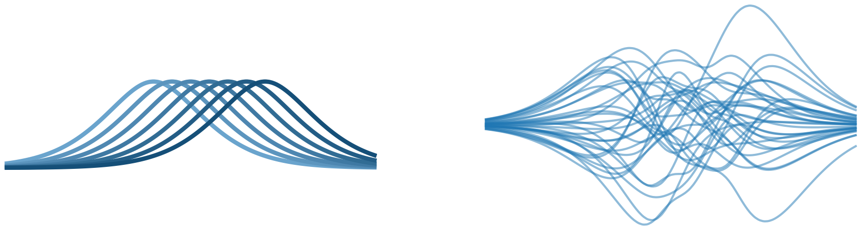
Variational family

Assume mutual absolute continuity between the distributions of the prior π_f and true posterior $\pi_{f|\boldsymbol{y}}$. Define the *variational family* $\mathbb{Q}$ to be the set of all measures equal to the distribution of $f \mid \boldsymbol{u}$ for some choice of variational parameter values $\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}$. Restricting the variational formulation of Bayes' Rule to $\mathbb{Q}$ yields the optimization problem

$$\operatorname*{arg\,min}_{\mathbb{q}_f \in \mathbb{Q}} D_{\mathrm{KL}}(\mathbb{q}_f \mid\mid \pi_f) + \frac{1}{2} \mathop{\mathbb{E}}_{f \sim \mathbb{q}_f} (\boldsymbol{\gamma} - f(\boldsymbol{x}))^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\gamma} - f(\boldsymbol{x})) \tag{2.49}$$

Variational approximation

where we have dropped constant terms from the likelihood density because they do not affect the optima. Our *variational approximation* is obtained by solving this optimization problem. By the chain rule for Kullback–Leibler divergences, $D_{\mathrm{KL}}(\mathbb{q}_f \mid\mid \pi_f) = D_{\mathrm{KL}}(\mathbb{q}_{f(\boldsymbol{z})} \mid\mid \pi_{f(\boldsymbol{z})})$ reduces to the Kullback–



(a) Canonical basis functions

(b) Update terms

Figure 2.10. The type of approximation made using inducing points can be understood in a pathwise manner: the inducing approximation employs the seven displayed canonical basis functions, rather than the thirty-one used in the true posterior, which possess significantly more overlap. The update term fades to zero as we move away from the data, highlighting the role of the prior in representing uncertainty.

Leibler divergence between the respective finite-dimensional marginal distributions at the inducing locations $\boldsymbol{z}$. It follows that the optimization objective of the variational approximation is finite.

We thus see that the *inducing point* approximation studied by Opper and Archambeau [83], when re-interpreted using the variational inference framework of Titsias [114], coincides with the pathwise variational approximation constructed here. The same is true for other classes of inducing points, including the variational family originally proposed by Titsias [114], where $\boldsymbol{\mu}$ is randomized.

Better yet, the constructions above are straightforward, principled, and mathematically sound. In particular, we do not rely on heuristic arguments involving *evidence lower bounds* which, in the absence of connections with the Kullback–Leibler divergence, require us to *posit* that optimizing certain quantities will result in improved posterior approximation. Instead, by virtue of the Kullback–Leibler divergence generating a Hausdorff topology on the space of probability measures, this is proven.

Inducing point approximations perform well in many settings, and are particularly effective in cases involving large quantities of data that contain redundant information about the posterior. In certain asymptotic regimes, Burt et al. [16] have shown that the number of inducing points can be taken logarithmic in the size of the data while maintaining approximation accuracy.

Among such approximations, the variational family presented above is particularly interesting because the matrix $\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \mathbf{\Lambda}$ which needs to be inverted is generally better-behaved numerically compared to matrices present in other approximations, owing to the presence of $\mathbf{\Lambda}$, whose learned values tend to concentrate away from zero. This makes this choice particularly attractive for use in combination with iterative solvers [31, 40, 85, 77, 86]: we discuss this idea, along with other avenues for future work, in Chapter 4.

This concludes our study of inducing point methods from the pathwise perspective, which are identified with sparsified pathwise approximations where instead of a sum of n kernel terms, we have a smaller sum of $m \ll n$ kernel terms. We now consider another class of approximations that one can consider using in practice.

2.4.2. Approximate priors. In the preceding sections, we presented a number of posterior approximations which were built by approximating individual terms within the pathwise formalism. We considered approximations where the prior term is replaced with a finite basis function approximation, and where the data term is replaced with a sparser analog. An obvious question one can ask is, instead of approximating the prior term, why not just change the model by using a finite basis function prior to begin with?

The answer to this question is that the resulting models become fundamentally finite-dimensional, and lose expressive capacity, often resulting in reduced performance [88]. This is often particularly pronounced in Bayesian optimization using Fourier feature methods [118, 79], which have nonetheless attracted attention in that setting owing to their convenience [51, 97]. This can be seen precisely by comparing the Fourier feature pathwise update

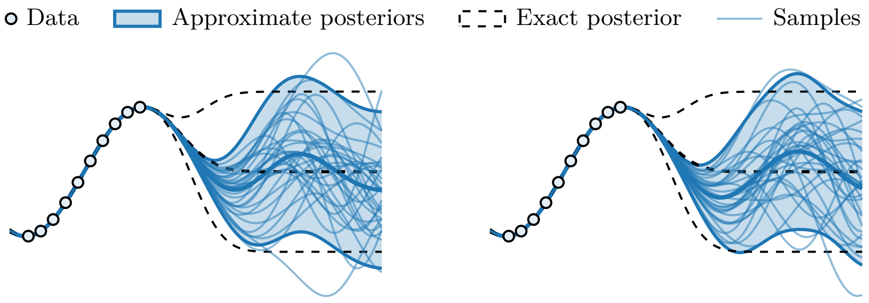
$$\tilde{f}(\omega, \cdot) + \underbrace{\boldsymbol{\phi}(\cdot)^T \boldsymbol{\phi}(\boldsymbol{x})}_{\text{replaces } \mathbf{K}_{(\cdot)\boldsymbol{x}}}(\underbrace{\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x})}_{\text{replaces } \mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}} + \mathbf{\Sigma})^{-1}(\boldsymbol{\gamma} - \tilde{f}(\omega, \boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega)) \tag{2.50}$$

to the original pathwise update

$$f(\omega, \cdot) + \mathbf{K}_{(\cdot)\boldsymbol{x}}(\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}} + \mathbf{\Sigma})^{-1}(\boldsymbol{\gamma} - f(\omega, \boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega)) \tag{2.51}$$

under the infinite-dimensional prior. Observe that all that has changed is that the prior f has been replaced with $\tilde{f}$, and the kernel matrices $\mathbf{K}_{(\cdot)\boldsymbol{x}}$ and $\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}$ have been replaced with the approximations $\boldsymbol{\phi}(\cdot)^T \boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\phi}(\boldsymbol{x})$. Observe further that since $\tilde{f}(\omega, \cdot) = \boldsymbol{\phi}(\cdot)^T \boldsymbol{w}(\omega)$ where $w_i \sim \mathrm{N}(0, 1)$, we can write

$$(f \mid \boldsymbol{y}) \approx \boldsymbol{\phi}(\cdot)^T(\boldsymbol{w}(\omega) + \boldsymbol{v}'(\omega)) \tag{2.52}$$



(a) Sine and cosine

(b) Random phase

Figure 2.11. Example of the *variance starvation* phenomenon in two different random Fourier feature models, compared to the true posterior. We see two problems: the mean spuriously oscillates and the error bars grow too slowly away from the data. This can cause the upper confidence bound acquisition, which is the global minima of the error bars, to appear in the completely wrong location. With the given hyperparameter choices, $n = 10$ data points is enough to exhibit considerable approximation error.

where $\boldsymbol{v}'(\omega) = \boldsymbol{\phi}(\boldsymbol{x})(\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x}) + \boldsymbol{\Sigma})^{-1}(\boldsymbol{\gamma} - \tilde{f}(\omega, \boldsymbol{x}) - \boldsymbol{\varepsilon}(\omega))$ are random weights. Therefore, if we change the model, then the posterior becomes a sum of the specified finite basis functions, which do not grow in quantity as data size increases.

If we think of the dimension of the vector space spanned by all basis functions as the representational capacity of the finite-dimensional model, we see that for Fourier feature posteriors this capacity does not grow as data size increases. This is to be contrasted with the approximate pathwise update, for which the representational capacity grows in spite of its finite-dimensional nature due to the presence of n kernel functions $k(x_j, \cdot)$ in the sum.

The result is that if the true posterior takes on a difficult-to-represent shape—which will often be the case if n is large enough, but might also occur even if it is not—then performance can degrade disastrously. This can be seen in Figure 2.11. This phenomenon has been called *variance starvation* [118, 79] in the literature for its tendency to produce error bars that are too narrow due to lack of representational capacity—we note this name is misleading because it is also capable of producing error bars which are far too large.

Variance starvation can be alleviated in a number of ways. One way is to avoid using Fourier bases for representing the posterior, and rely instead on basis functions which are compactly supported or otherwise possess some sense of locality. This mirrors ideas in numerical analysis surrounding *Runge's phenomenon* [36, 28], where similar behavior occurs in polynomial interpolation. From this angle, the pathwise viewpoint offers a canonical way to select the functions used for representing the data.

By not approximating terms that do not need to be approximated, pathwise approximations avoid limiting representational capacity and thus retain performance. Variance starvation has been an ongoing difficulty in Bayesian optimization algorithms for some time—in our view, pathwise approximations of the kind described here, when applicable, largely resolve the issue.

2.5. Error analysis

Pathwise posterior approximations are, ultimately, approximations. Therefore, a key question one can ask is: how accurate are they? To quantify this, we need an appropriate notion of *distance* to quantify how far away the two random variables of interest are. This notion should possess a number of key properties.

1. It should be *distributional* in nature to reflect the fact that it is the information contained in the posterior that is of interest to us, rather than the precise way in which the random variables are generated.

2. The distance between the true posterior and pathwise approximations should be finite, in order to facilitate meaningful comparisons.

Wasserstein distance

The *Wasserstein distance* [117] between probability measures is defined as

$$W_{p,d}(\pi, \pi')^p = \inf_{\gamma \in \Gamma(\pi,\pi')} \int_{A \times A} d(a, a')^p \, \mathrm{d}\gamma(a, a') \tag{2.53}$$

where $p \geq 1$, d is a metric on A, and $\Gamma(\pi, \pi')$ is this set of all *couplings* of π with π', namely probability measures γ supported on $A \times A$ whose marginals equal π and π'. We adopt this expression as our notion of distance.

This distributional distance can be understood as the expected distance between two random variables a and a', with distributions π and π', where the random numbers used to generate a and a' are linked in order to make the expectation as small as possible. The smallest possible expected distance is zero, which only occurs if the random variables can be made identical to one another—or, in other words, if their distributions coincide.

This definition satisfies both requirements. By virtue of varying random numbers—or, more precisely, optimizing over couplings—it compares distributions. Moreover, unlike alternatives such as the Kullback–Leibler divergence or total variation distance, Wasserstein distances metrize the topology of weak convergence [117] , do not impose restrictive absolute continuity requirements, and are finite in the cases of interest.

We also analyze error in a second way: using the supremum norm between kernels. Observe that pathwise approximations under mean-zero priors possess the exact same mean as the true posterior. Therefore, when restricted to pathwise approximations under mean-zero priors, this notion gives an actual metric between probability distributions in the given class. We now state our main results, suppressing ranges from function spaces to ease notation.

Pathwise posterior Wasserstein error bound

Proposition 2.10. *Assume $X \subseteq \mathbb{R}^d$ is compact with volume $\operatorname{vol}(X)$ and that $f \sim \operatorname{GP}(0, k)$ is almost surely continuous. Let $\tilde{f}(\omega, x) = \sum_{i=1}^{\ell} w_i(\omega)\phi_i(x)$, and let $(\tilde{f} \mid \boldsymbol{y})(\omega, x) = \tilde{f}(\omega, x) + \mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}(\boldsymbol{y} - \tilde{f}(\boldsymbol{x}))$. Then we have*

$$W_{2,L^2(X)}(\tilde{f} \mid \boldsymbol{y}, f \mid \boldsymbol{y}) \leq C_1 W_{2,C^0(X)}(\tilde{f}, f) \tag{2.54}$$

where $C_1 = \Big(2\operatorname{vol}(X)\Big(1 + \|k\|^2_{C^0(X\times X)}\|\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\|^2_{L(\ell^\infty;\ell^1)}\Big)\Big)^{1/2}$.

Proof. The idea is to first use the pathwise update to prove a pointwise bound, then take expectations with respect to a minimizing coupling to obtain a Wasserstein bound. Using Hölder's inequality with $p = 1$ and $q = \infty$, write

$$\begin{aligned}
&\big|(\tilde{f} \mid \boldsymbol{y})(\cdot) - (f \mid \boldsymbol{y})(\cdot)\big|^2 && (2.55)\\
&\quad\leq 2\big|\tilde{f}(\cdot) - f(\cdot)\big|^2 + 2\big|\mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}(\tilde{f}(\boldsymbol{x}) - f(\boldsymbol{x}))\big|^2 && (2.56)\\
&\quad\leq 2\|\tilde{f} - f\|^2_{L^\infty(X)} + 2\|\mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\|^2_{\ell^1}\|\tilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})\|^2_{\ell^\infty} && (2.57)\\
&\quad\leq 2\Big(1 + \|\mathbf{K}_{(\cdot)\boldsymbol{x}}\|^2_{\ell^\infty}\|\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\|^2_{L(\ell^\infty;\ell^1)}\Big)\|\tilde{f} - f\|^2_{L^\infty(X)} && (2.58)\\
&\quad\leq \underbrace{2\Big(1 + \|k\|^2_{C^0(X\times X)}\|\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\|^2_{L(\ell^\infty;\ell^1)}\Big)}_{C_0}\|\tilde{f} - f\|^2_{C^0(X)} && (2.59)
\end{aligned}$$

where $\|\cdot\|_{L(A;B)}$ is the operator norm between A and B, $\|\cdot\|_{\ell^p}$ is the Euclidean p-norm, and where we have used almost sure continuity of sample paths to replace $\|\cdot\|_{L^\infty(X)}$ with $\|\cdot\|_{C^0(X)}$. We now lift this bound to a bound on the Wasserstein distance by integrating both sides with respect to an optimal coupling $\gamma \in \Gamma(\tilde{\pi}, \pi)$, where $\tilde{\pi}$ and π are the distributions of $\tilde{f}$ and f, respectively. Writing

$$W_{2,L^2(X)}(\tilde{f} \mid \boldsymbol{y}, f \mid \boldsymbol{y})^2 = \inf_{\gamma\in\Gamma(\tilde{\pi},\pi)} \mathbb{E}_\gamma \|(\tilde{f} \mid \boldsymbol{y}) - (f \mid \boldsymbol{y})\|^2_{L^2(X)} \tag{2.60}$$

$$\leq C_0 \operatorname{vol}(X) \inf_{\gamma\in\Gamma(\tilde{\pi},\pi)} \mathbb{E}_\gamma \|\tilde{f} - f\|_{C^0(X)} \tag{2.61}$$

$$\leq C_1^2 W_{2,C^0(X)}(\tilde{f}, f)^2 \tag{2.62}$$

and noting that $C^0(X)$ is a separable metric space, which ensures that the Wasserstein distance over it is well-defined, gives the claim. ∎

Pathwise posterior kernel error bound

Proposition 2.11. *Under the same assumptions as Proposition 2.10, letting $k^{(f|\boldsymbol{y})}$, $k^{(\tilde{f}|\boldsymbol{y})}$, and $k^{(\tilde{f})}$ be the covariance kernel of these respective processes, we have*

$$\|k^{(\tilde{f}|\boldsymbol{y})} - k^{(f|\boldsymbol{y})}\|_{C^0(X\times X)} \leq C_2 \|k^{(\tilde{f})} - k\|_{C^0(X\times X)} \tag{2.63}$$

where $C_2 = n\Big(1 + \|k\|_{C^0(X\times X)} \|\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\|_{L(\ell^\infty;\ell^1)}\Big)^2$.

Proof. The idea is again to apply standard function-analytic inequalities to the pathwise update. For a kernel k, define the linear operator $M_k : C^0(X \times X) \to C^0(X \times X)$ by

$$(M_k c)(\cdot, \cdot') = c(\cdot, \cdot') - c(\cdot, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)} - \mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1} c(\boldsymbol{x}, \cdot) \tag{2.64}$$

$$+ \mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1} c(\boldsymbol{x}, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}. \tag{2.65}$$

By construction, we have that

$$k^{(f|\boldsymbol{y})} = M_k k \qquad\qquad k^{(\tilde{f}|\boldsymbol{y})} = M_k k^{(\tilde{f})}. \tag{2.66}$$

Thus, it suffices to prove that M_k is bounded and calculate its operator norm. To do so, write

$$\|M_k c\|_{C^0(X\times X)} \leq \|c\|_{C^0(X\times X)} + 2\|c(\cdot, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}\|_{C^0(X\times X)} \tag{2.67}$$

$$+ \|\mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1} c(\boldsymbol{x}, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}\|_{C^0(X\times X)}. \tag{2.68}$$

We bound these term-by-term. For the second term, using Hölder's inequality with $p = 1$ and $q = \infty$, write

$$\begin{aligned}
&\left\lVert c(\cdot, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}\right\rVert_{C^0(X\times X)} && (2.69)\\
&\quad= \sup_{\boldsymbol{x}',\boldsymbol{x}''\in X} c(\boldsymbol{x}', \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}''} && (2.70)\\
&\quad\leq \sup_{\boldsymbol{x}',\boldsymbol{x}''\in X} \left\lVert c(\boldsymbol{x}', \boldsymbol{x})\right\rVert_{\ell^\infty}\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)}\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}''}\right\rVert_{\ell^\infty} && (2.71)\\
&\quad\leq \lVert c\rVert_{C^0(X\times X)}\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)}\lVert k\rVert_{C^0(X\times X)} && (2.72)
\end{aligned}$$

where we have used continuity of k to obtain the last inequality. For the third term, similarly, write

$$\begin{aligned}
&\left\lVert\mathbf{K}_{(\cdot)\boldsymbol{x}}\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}c(\boldsymbol{x}, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\mathbf{K}_{\boldsymbol{x}(\cdot)}\right\rVert_{C^0(X\times X)} && (2.73)\\
&\quad= \sup_{\boldsymbol{x}',\boldsymbol{x}''\in X} c(\boldsymbol{x}', \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}c(\boldsymbol{x}, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}c(\boldsymbol{x}, \boldsymbol{x}'') && (2.74)\\
&\quad\leq \lVert k\rVert_{C^0(X\times X)}^2\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}c(\boldsymbol{x}, \boldsymbol{x})\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)} && (2.75)\\
&\quad\leq n\lVert k\rVert_{C^0(X\times X)}^2\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)}^2\lVert c\rVert_{C^0(X\times X)} && (2.76)
\end{aligned}$$

where n factor appears due to use of $\lVert\cdot\rVert_{L(\ell^\infty;\ell^1)}$. Putting these inequalities together gives

$$\begin{aligned}
\frac{\lVert M_k c\rVert_{C^0(X\times X)}}{\lVert c\rVert_{C^0(X\times X)}} &\leq \Big(1 + 2\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)}\lVert k\rVert_{C^0(X\times X)} && (2.77)\\
&\qquad + n\left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)}^2\lVert k\rVert_{C^0(X\times X)}^2\Big) && (2.78)\\
&\leq n\Big(1 + \left\lVert\mathbf{K}_{\boldsymbol{x}\boldsymbol{x}}^{-1}\right\rVert_{L(\ell^\infty;\ell^1)}\lVert k\rVert_{C^0(X\times X)}\Big)^2 && (2.79)
\end{aligned}$$

and so the claim follows. ■

The above arguments shows that, for a given dataset size, posterior approximation error is controlled by prior approximation error. Thus, if we increase approximation accuracy in the prior, we are guaranteed to also increase approximation accuracy in the posterior.

The key idea in the above arguments was to analyze the pathwise approximation by applying standard function-analytic inequalities to the pathwise representation itself. By varying the resulting norms, a wide set of similar inequalities follow: we make no attempt to optimize bounds, and instead focus on presenting the technique in the simplest manner.

Approximation error and asymptotic posterior contraction

The bounds presented are not tight, particularly in large-data settings where the inverse matrix norm in the constant becomes large. This may tempt one to conclude that approximation error will grow, but this is often false—instead, the bounds become loose. One way to see this is by considering posterior contraction: in large-data settings, the posterior will be concentrated, and since our approximations are exact with respect to the mean, there is little room left for error, in the absolute sense, to accumulate.

Different prior approximations possess different error behavior. The random Fourier feature approach is particularly attractive because its Monte Carlo nature makes it decay at a dimension-free rate of $\sqrt{\ell}$: Sutherland and Schneider [109] provide bounds on this term. Note also that the terms in our kernel bounds depend on the dimension only through the kernel matrix, which itself depends on the dimension of the data and not the ambient space. In this sense, our bounds are very well-behaved with respect to dimension.

This concludes our theoretical presentation and analysis. We now move to evaluating these ideas in practical settings, and discussing the broad picture painted by the ideas presented here and in the preceding sections.

2.6. Parallel Thompson sampling

We now study empirically how pathwise approximations affect downstream performance of Gaussian process models, using a simple experiment to showcase the key behavior. For this, we perform a Bayesian optimization experiment using parallel Thompson sampling for acquisition. This variant of Thompson sampling consists of sampling p independent posterior draws, then computing the acquisition and evaluating the target function at each location simultaneously. Specifically, for $s = 1, .., p$, define

$$x_{t+1,s}(\omega) = \operatorname*{arg\,min}_{x\in X} \alpha_{t,s}(\omega, x) \qquad\qquad \alpha_{t,s} \sim f \mid y_1, .., y_t \tag{2.80}$$

where f is the Gaussian process. Our goal is to understand how approximation error affects performance of Bayesian optimization using this acquisition function, particularly among domains of different dimension. We focus on data-efficiency rather than computational costs.

We now define the Bayesian model. Let the domain X to be the unit hypercube in dimensions $d = 2$, $d = 4$, and $d = 8$. Let f be assigned a Matérn prior with unit variance, length scale $\kappa = \sqrt{d/100}$, and smoothness $\nu = 5/2$.

Each observation is assigned an independent Gaussian likelihood with error variance $\sigma^2 = 10^{-3}$. From these, we obtain the posterior distribution.

We select target functions to minimize by sampling from a random Fourier feature approximation of the prior, and obtain their minima for purposes of regret using multi-start gradient descent. To account for variable problem difficulty, we allow the number of evaluations to vary according to dimension, setting $n = 64$ for $d = 2$, $n = 256$ for $d = 4$, and $n = 1024$ for $d = 8$. Similarly, we set the degree of parallelism to allow $p = d$ simultaneous evaluations. We repeat each experiment 32 times to assess variability.

We consider two sequential baselines and three approximate acquisition strategies using the posterior Gaussian process. For the sequential baselines, we use *random search* [6] and *dividing rectangles* [56], selected for their simplicity. For the Gaussian process baselines, we consider (a) a *Cholesky* grid-based approach, (b) a *random Fourier feature* posterior process, and (c) a *pathwise* approximate posterior using the same Fourier feature approximation for the prior term. We now describe these in more detail.

The Cholesky grid-based approach works by (i) drawing a set with $250,000$ points uniformly at random, (ii) sampling a vector of independent Gaussians whose mean and variance are equal to the posterior predictive distribution at those points, (iii) choosing the 2048 smallest elements, (iv) sampling the posterior Gaussian process jointly at the chosen locations, and (v) choosing the smallest value. This gives an approximation to the true acquisition locations using the candidate locations on the grid.

The random Fourier feature and pathwise approximate posterior approaches do not use grids. Instead, we (i) draw a set of $250,000$ points uniformly at random, (ii) evaluate $f \mid \boldsymbol{y}$ at those locations to choose the 32 smallest points, and (iii) run multi-start optimization using L-BFGS-B [17] from each candidate location to find the minima. This procedure is selected so as to be relatively similar to the grid-based approach and ensure a fair comparison.

To understand the effect of posterior approximation, we examine both basis-function-based Gaussian processes with a total of $\ell = 256$, $\ell = 1024$, and $\ell = 4096$ basis functions. For the pathwise approximate posterior, this is the sum of both the number of basis functions used for the prior with the number of canonical basis functions—this choice helps avoid unfairly penalizing the random fourier Feature model. We use random-phase-based Fourier features in both approximate posteriors.

Results can be seen in Figure 2.12, where we plot median regret curves,

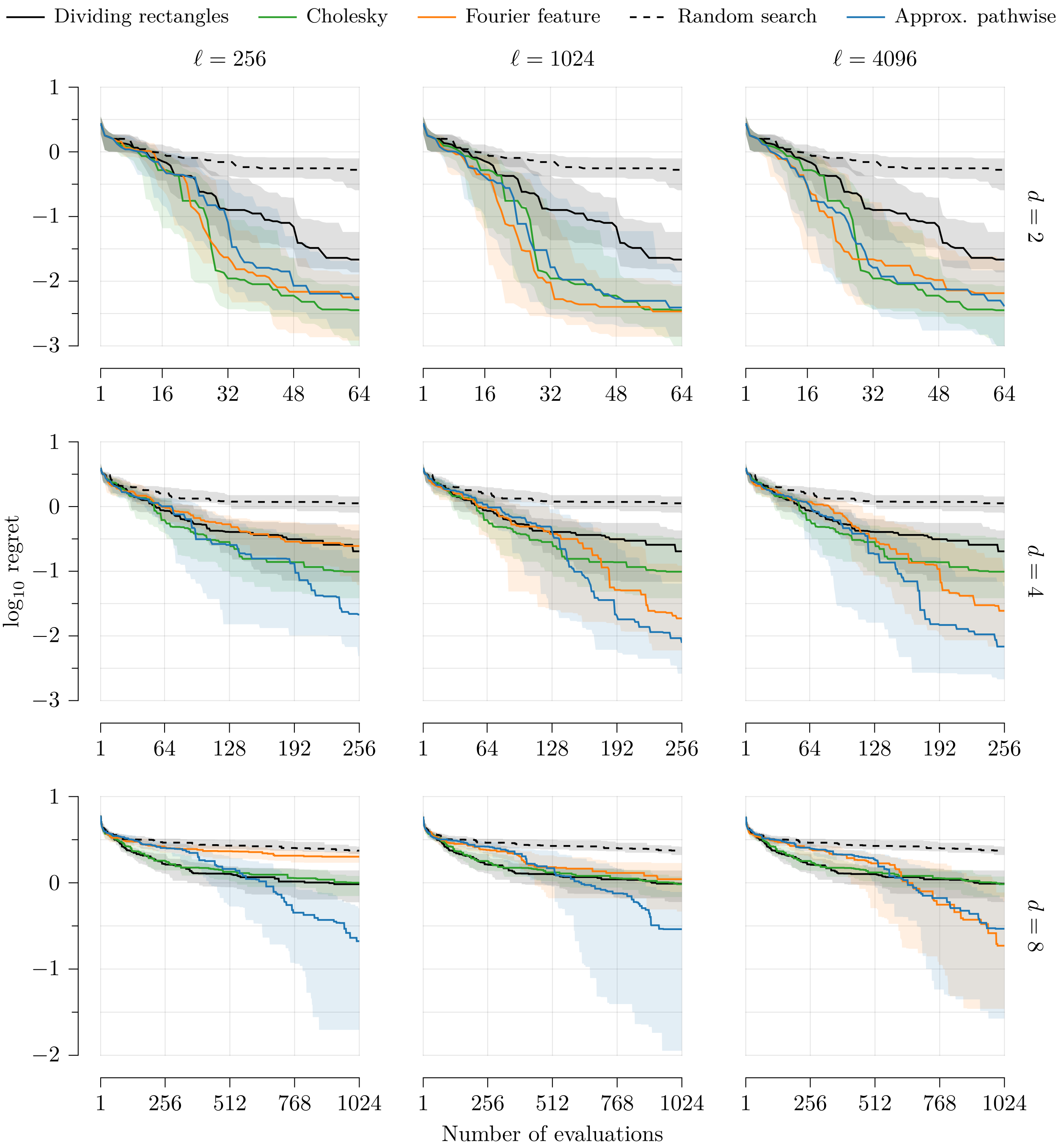


Figure 2.12. Parallel Thompson sampling benchmark results.

along with first and third quartiles. Immediately, we see that the effect of the strategy used for computing the acquisition function varies according to dimension.

In the two-dimensional setting, all Gaussian process methods perform comparably, outperforming the random search and dividing rectangles baselines. This occurs even in the regime with $\ell = 256$ basis functions, and shows that posterior approximation error does not play a particularly significant role for the given comparisons and parameters.

In the four-dimensional setting, the behavior is different. This time, in the $\ell = 256$ case, the pathwise approximate posterior outperforms both other Gaussian process baselines. Here, the random Fourier feature baseline performs worse than the Cholesky baseline. If we increase the amount of basis functions, the random Fourier feature baseline recovers this difference, yielding relatively comparable performance to the pathwise approximation for $\ell = 1024$ and $\ell = 4096$, and outperforming the Cholesky baseline.

In the eight-dimensional setting, the pathwise approximation outperforms all baselines for $\ell = 256$ and $\ell = 1024$. In those cases, both the random Fourier feature and Cholesky baselines are hampered by the curse of dimensionality, and perform no better than the dividing rectangles baseline—for $\ell = 256$, random Fourier features are comparable to random search. With $\ell = 4096$ basis functions, the performance of Fourier features recovers. In contrast, the pathwise approximation is insensitive to the number of basis functions.

In summary, these results largely confirm the viewpoint developed using the preceding theory. We saw previously that the pathwise approximate posterior can be seen as modifying the random Fourier feature posterior by replacing certain Monte Carlo approximations with their expected values. It is therefore reasonable to suppose that this gives a more accurate approximation of the true posterior. Our results are thus consistent with the true Gaussian process being a better-performing model for Bayesian optimization.

For the Matérn prior used, this mirrors modern understanding: low-rank and other finite-dimensional models are generally less expressive than true Gaussian processes and mostly favored in cases where their properties help control computational costs. Using the true Gaussian process, or a more accurate approximation thereof, can result in better performance.

The benefits of using a method that avoids solving arbitrarily large linear systems at test-time are clear from the point of view of both computational complexity and numerical stability. From these perspectives, our results show that already in $d = 4$ the improvements are enough to make a noticeable difference in downstream tasks.

2.7. Conclusion

In the Gaussian case, the distributional notion of *conditioning* can be reformulated using random variables, yielding a notion of *pathwise conditioning*. In the preceding sections, we developed and studied this point of view for Gaussian processes, allowing us to express a posterior Gaussian process as the sum of a *prior term*, and a *data-dependent term*.

Pathwise conditioning gives a powerful way to think about posterior Gaussian processes. Using this notion, we re-interpreted classical methods such as *random Fourier features* and *inducing points*, by observing that they correspond to approximations made to individual terms within the pathwise update. Crucially, we observed that one could approximate different terms within the formula individually.

Using this observation, we constructed accurate finite-dimensional posterior approximations which nonetheless yield actual *random functions*. These functions are sums of two sets of finite basis functions with dependent random coefficients. These coefficients can be sampled in advance, after which the functions effectively become deterministic and can be evaluated at arbitrary points, as well as differentiated using standard techniques.

The resulting approximations are particularly useful in decision-making settings such as *Bayesian optimization*, where acquisition functions constructed from Gaussian process sample paths need to be optimized or evaluated at arbitrary locations. This makes it possible to implement Thompson sampling using automatic differentiation in a straightforward manner without any sophisticated bookkeeping. In particular, there is no need to track at what points the Gaussian process has already been evaluated at.

Using the presented technique, minimizing a Gaussian process sample path can be done in linear time and without incurring large approximation error that degrades performance as part of the iterative minimization procedure itself. This is particularly useful in higher-dimensional settings, where grid-based methods and other alternatives may be hampered by the curse of dimensionality.

The performance of pathwise approximate posterior Gaussian processes depends on the approximation accuracy of the prior term. We presented a number of methods for doing so, each suited to their respective settings. For stationary kernels, random Fourier feature methods are particularly attractive due to their good behavior with respect to dimension, but are by no means the only choice. We hope these developments prompt further study of other possible choices for approximate priors.

Pathwise approximations are also of interest as a potential organizing principle for designing Gaussian process software packages. In particular, one can consider implementing a Gaussian process with separate methods for evaluating the process and re-sampling its random weights. Depending on the situation, this might be a more convenient application programming interface than working with distributions at user-specified locations. Exploring these alternatives is a promising avenue for further work.

Similarly, pathwise approximations make a strong case that efficiently sampling from a Gaussian process prior is a key software primitive that a Gaussian process package should support as part of its kernel implementations. Understanding how to organize different approximations, which may possess different properties and may also be computed in different ways, is another promising avenue for future work.

Gaussian processes have been applied in many settings, ranging from areas such as spatial statistics where they are mostly used by humans, to areas such as Bayesian optimization and model-based reinforcement learning where they are mostly used by computers. We hope that the contributions presented here improve their ease of use, broaden their applicability, and enable new applications not yet considered. We now proceed to study a different route for expanding the set of settings Gaussian processes can be applied in.

Chapter 3

Non-Euclidean Matérn Gaussian Processes

Stationary kernels on Euclidean spaces are one of the most widely-used Gaussian process model classes. These kernels are attractive because they work effectively and it is generally easy to understand the kind of prior information they introduce into a problem. This makes them a valuable tool for practitioners to use in the manner required for the task at hand.

Our goal throughout has been to expand the settings in which Gaussian process models and decision systems built atop them can be used. We have so far focused on doing so by making existing models easier to work with, but now pursue a different approach: namely, we focus on expanding the scope of models one can consider working with. We again emphasize constructiveness and making abstract ideas accessible to practitioners.

We focus on the setting of Riemannian geometry, which describes a widely-occurring class of geometric shapes and spaces. We thus study Gaussian processes whose domains are manifolds, rather than Euclidean spaces. Working with manifolds often involves thinking carefully about discretization: we therefore also study purely discrete settings involving meshes and weighted undirected graphs, which are of inherent interest in their own right.

Our guiding theme is to ask: how can we make Gaussian processes on Riemannian manifolds be just as effective a model class as their Euclidean counterparts? We proceed to introduce, develop, and explore this topic.

3.1. Riemannian Matérn Gaussian processes

We begin with a key question: how should one generalize the most widely-used class of Gaussian process models to the Riemannian manifold setting? There are multiple potential definitions one can introduce, some of which turn out to be much better-behaved mathematically than others. In order to pursue these questions, we begin by introducing and reviewing the setting of differential geometry.

3.1.1. Review of differential geometry. We now briefly review the notions needed to define and understand manifolds.[1] These are sets equipped with additional structure encoding their geometric shape. Many geometric shapes, in turn, can be realized as manifolds: two examples are shown in Figure 3.1. We use the works of Lee [68, 67, 66] as our primary references on this topic. For a machine-learning-oriented text, see Bronstein et al. [14].

Topological space

A *topological space* is a set X together with a *topology*, which is a collection of subsets $\mathcal{O}_X \subseteq 2^X$ called the *open sets*—these include the empty set, the space itself, and are closed under pairwise intersections and arbitrary unions. A given set can admit many topologies. Topological spaces admit notions of *locality*, *convergence*, *continuity*, *compactness*, *paracompactness*, *denseness*, and many others. The *Hausdorff* property implies that limits are unique.

Topological manifold

A paracompact Hausdorff topological space $(X, \mathcal{O}_X)$ is called a d-dimensional *topological manifold* if it is locally homeomorphic to $\mathbb{R}^d$ equipped with the standard topology. For such a manifold, there is a set $\mathcal{A}_X$ called the *atlas* whose elements are the local homeomorphisms, called *charts*. An atlas is *maximal* if it is the largest possible such set: for any given atlas, a maximal atlas containing it it is unique.

Smooth differentiable manifold

A *smooth differentiable manifold* is a triple $(X, \mathcal{O}_X, \mathcal{A}_X)$ where X is a topological manifold, and $\mathcal{A}_X$ is a C^∞-atlas. A C^∞-atlas is an atlas in which, for any charts $x, y \in \mathcal{A}_X$ with overlapping domain, the *chart transition map* $y \circ x^{-1}$ is an infinitely differentiable function. Homeomorphisms compatible with such charts are called *diffeomorphisms*. A smooth manifold is called *oriented* if the Jacobian determinant of all chart transition maps in its atlas

[1] The Russian word for *manifold* is *mnogoobrazie* (mnəgɐɐbrˈazʲɪjə). The word *mnogo* means *many*, and the word *obraz* roughly means *form*, *view*, or *image*. Thus, a *manifold* is a *space with many views into it*—a name one could easily call well-chosen and evocative.

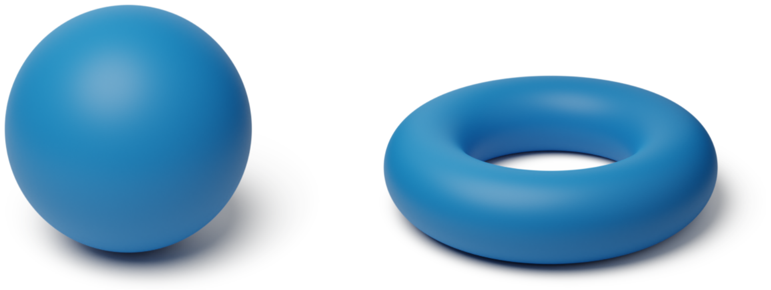


Figure 3.1. Two common manifolds: the sphere $\mathbb{S}^2$ and torus $\mathbb{T}^2$.

is positive. Such atlases admit maximal analogs.

Smooth function

The notion of smoothness extends to real-valued functions on manifolds, and functions between manifolds. For a smooth manifold X, we say that $f \in C^\infty(X; \mathbb{R})$ if it is infinitely differentiable in any chart. Similarly, we say that a map $f : X \to Y$ between a pair of smooth manifolds is smooth if it is infinitely differentiable in any pair of charts. Smooth maps between manifolds are automatically continuous. To ease notation, we often omit real ranges from function spaces that occur in the sequel.

Manifold with boundary

A *manifold with boundary* is one of a wide class of spaces generalizing the notion of a manifold while preserving most geometric characteristics that make manifolds useful and interesting in the first place. Such a space is defined analogously to an ordinary manifold, except that it is also possible for it to be locally homeomorphic to a Euclidean half-plane. The ideas we consider admit natural generalizations to such settings, but we do not pursue these, and in particular only consider manifolds without boundary.

Tangent bundle

Every smooth manifold X gives rise to its *tangent bundle* $(TX, \operatorname{proj}_X, X)$, which contains a smooth manifold TX and a surjective projection map $\operatorname{proj}_X : TX \to X$. The former is defined as $TX = \bigcup_{x\in X}\{(x, v) : v \in T_x X\}$, where $T_x X$ is the tangent space, defined as a vector space of equivalence classes of directional derivatives of smooth curves through x. This set can be equipped with the structure of a smooth manifold, arising canonically from the smooth structure of X.

Pushforward map

Given a smooth map $f : X \to Y$ that transforms a smooth manifold X into another smooth manifold Y, the *pushforward map* $f_* : TX \to TY$ describes how each tangent space $T_x X$ transforms into $T_{f(x)} Y$ by the action of f. Since f is smooth, each tangent space is transformed linearly: in this sense, f_*

encodes the local behavior of f around each point on its domain.

Vector field

Section

A map $f : X \to TX$ satisfying $\operatorname{proj}_X \circ f = \operatorname{id}_X$ is called a *vector field.* Note that this is *not the same* as a map $X \to \mathbb{R}^d$ and often cannot usefully be expressed in this way: we expand on this in the sequel. We say that f is a *cross-section*, or simply a *section*, of the tangent bundle. If f, when viewed as a map between manifolds, is smooth, we write $f \in \Gamma(TX)$. Functions can multiply against vector fields by scaling them pointwise.

Cotangent bundle

Covector field

Tensor field

Analogously, we can define the *cotangent bundle* $T^*X = \bigcup_{x \in X}\{(x, \phi) : \phi \in T_xX^*\}$ where, compared to the tangent bundle, we have replaced the vector space T_xX with its topological dual T_xX^*. A section of this bundle is called a *covector field.* We can similarly define a number of other bundles, each equipped with a projection map onto the base space. Introducing sections of such bundles gives rise to the notion of a *tensor field* and other generalizations. All such maps are called smooth if they are smooth maps between manifolds.

Vector bundle

Vector bundles are a particularly important class of bundles. Given the projection map proj_X onto the base space, the *fiber* over x is defined as the preimage $\operatorname{proj}_X^{-1}\{x\}$. In a vector bundle, all fibers admit the structure of a vector space. This means that a space of sections of a vector bundle can itself be given the structure of a vector space. Many vector spaces which generalize the usual function spaces, such as spaces of vector fields equipped with additional structure, are constructed using suitable vector bundles.

Interior product

Exterior derivative

Vector fields can be inserted into covector fields by pairing vectors in each tangent space: this defines the *interior product* $\lrcorner$, which extends to tensor fields as long as the pairing being considered makes sense. We say that a totally antisymmetric $(0, k)$-tensor field is a k-form, and that a smooth function is a 0-form. The *exterior derivative*, denoted by d, maps k-forms into $(k + 1)$-forms. Interior products preserve antisymmetry.

Volume

Integration

Differential forms are often used to formalize and encode geometric structure in a way that is amenable to analysis. For example, a nowhere-vanishing d-form is called a *volume form*—such a form induces a notion of a *volume density*, which induces a notion of *integration* of smooth functions, and in turn, by the Riesz–Markov–Kakutani representation theorem, a Radon measure on the manifold, which we call the *volume measure.* On a general smooth manifold, the choice of a volume form is heavily non-unique.

Riemannian manifold

A *Riemannian manifold* is a quadruple $(X, \mathcal{O}_X, \mathcal{A}_X, g)$, usually written (X, g), where X is a smooth manifold and g is the *metric tensor*, which is a smooth symmetric positive definite $(0, 2)$-tensor field. The metric tensor can be thought of as an algebraic object encoding the manifold's quantitative shape, and canonically gives rise to notions such as *volume*, *integration*, and *geodesics*. In particular, we denote the volume form, volume measure, and volume density induced by the metric tensor as vol_g.

Nash Embedding

Riemannian manifolds are, by definition, topological spaces with additional structure. By Nash's Embedding Theorem, every Riemannian manifold can be isometrically embedded within an $\mathcal{O}(d^2)$-dimensional Euclidean space. This identifies Riemannian manifolds as geometric shapes located within Euclidean spaces. This perspective is often avoided for both technical and conceptual reasons: if the spacetime of the universe is a manifold, can an ambient Euclidean space actually have physical meaning?

3.1.2. The Laplace–Beltrami operator. Riemannian manifolds admit many different objects, such as connection one-forms, Ricci tensors, and other constructions which encode and mathematically describe their geometric properties. These are generally built using the metric along with derivatives and other operations. The *Laplace–Beltrami operator* is one such object, and forms a basic building block for defining differential equations on manifolds.

Laplace–Beltrami operator

Definition 3.1. *Let (X, g) be a Riemannian manifold, assumed oriented without loss of generality. Define the* DIVERGENCE *of a vector field to be the unique map*

$$\mathrm{div}_g : \Gamma(TX) \to C^\infty(X) \quad \mathrm{d}(v \lrcorner \mathrm{vol}_g) = \mathrm{div}_g\, v \cdot \mathrm{vol}_g \quad \forall v \in \Gamma(TX) \tag{3.1}$$

where $\cdot$ is pointwise multiplication of differential forms by smooth functions, and the GRADIENT *of a scalar function to be the unique map*

$$\mathrm{grad}_g : C^\infty(X) \to \Gamma(TX) \quad g(\mathrm{grad}_g f, v) = (\mathrm{d}f) \lrcorner v \quad \forall v \in \Gamma(TX). \tag{3.2}$$

Define the LAPLACE–BELTRAMI OPERATOR *to be*

$$\Delta_g : C^\infty(X) \to C^\infty(X) \qquad \Delta_g f = \mathrm{div}_g\, \mathrm{grad}_g\, f. \tag{3.3}$$

The Laplace–Beltrami operator, then, maps functions into the divergence of their gradient at every point. There are multiple equivalent ways of defining the Laplace–Beltrami operator: an alternative is to define it as the trace of the Riemannian Hessian, and another is to define it in coordinates. The latter expression illustrates that a Laplace–Beltrami operator maps functions into their locally averaged analogs, in a sense. For us, the relatively rich spectral properties possessed by this operator will be of key interest.

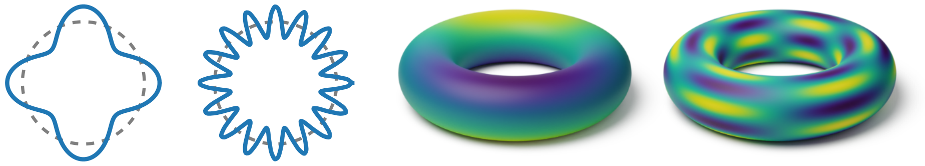


Figure 3.2. Eigenfunctions of the Laplace–Beltrami operator on a circle and torus.

Result 3.2. *Let (X, g) be a compact Riemannian manifold. Then the operator $-\Delta_g : C^\infty(X) \to C^\infty(X)$ is positive semi-definite, and extends uniquely to a self-adjoint unbounded positive operator $-\Delta_g : D(\Delta_g) \to L^2(X)$, where $D(\Delta_g) \subseteq L^2(X)$.*

Proof. Strichartz [108], Theorem 2.4. ■

Viewed in this way, the range of the Laplace–Beltrami operator is a Hilbert space, enabling us to use ideas from spectral theory to better understand its behavior. In particular, one can show that, owing to compactness, the spectrum of $-\Delta_g$ is discrete—this gives a particularly simple view of $-\Delta_g$ through its eigenvalues and eigenfunctions.

Sturm–Liouville decomposition

Result 3.3. *Let (X, g) be a compact Riemannian manifold. Then there exists an orthonormal basis f_n, $n \in \mathbb{Z}_+$, of $L^2(X)$ such that $-\Delta_g f_n = \lambda_n f_n$ with $0 = \lambda_0 \leq \lambda_1 \leq .. \leq \lambda_n$ and $\lambda_n \to \infty$ as $n \to \infty$. Moreover, $-\Delta_g$ admits the representation*

$$-\Delta_g f = \sum_{n=0}^{\infty} \lambda_n \langle f, f_n \rangle f_n \tag{3.4}$$

which converges unconditionally in $L^2(X)$ for all $f \in D(\Delta_g)$.

Proof. Chavel [21], page 139, or Canzani [19], Theorem 44. ■

This is a powerful result: the eigenfunctions f_n, shown in Figures 3.2 and 3.3, can be viewed as analogs of the Fourier basis adapted to the manifold's geometry. By expanding a function within the basis of eigenfunctions, we obtain an infinite sequence of basis coefficients—just like representing a periodic function in $L^2([-\pi, \pi]; \mathbb{R})$ by an infinite sum of complex exponentials. The Sturm–Liouville decomposition gives rise to a notion of *functional calculus*, which is key for our purposes, and which we now introduce.

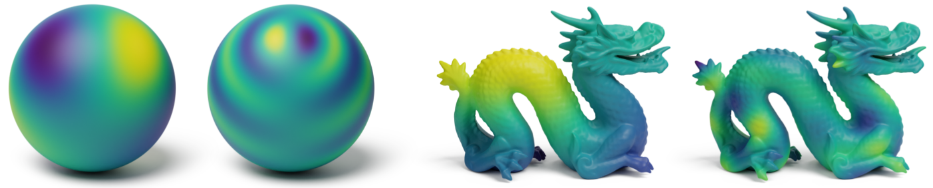


Figure 3.3. Eigenfunctions of the Laplace–Beltrami operator on a sphere and dragon manifold.

Functional calculus

Definition 3.4. *Let* $\Phi : [0, \infty) \to \mathbb{R}$*. Define the (possibly unbounded) operator* $\Phi(-\Delta_g) : D(\Phi(-\Delta_g)) \to L^2(X)$ *by*

$$\Phi(-\Delta_g) f = \sum_{n=0}^{\infty} \Phi(\lambda_n) \langle f, f_n \rangle f_n \tag{3.5}$$

where $D(\Phi(-\Delta_g)) = \{ f \in L^2(X) : \sum_{n=0}^{\infty} |\Phi(\lambda_n)|^2 |\langle f, f_n \rangle|^2 f_n < \infty \}$*.*

Functional calculus lets us extend the idea of applying functions from numbers to operators. This is done by applying the function of interest to the eigenvalues of the operator. We will use this to construct Gaussian processes as solutions of stochastic partial differential equations. First, however, we take a step back and examine why one would want to take this rather roundabout approach in the first place as opposed to a more direct alternative.

3.1.3. A no-go theorem for kernels on manifolds. Here, we begin exploring the idea of defining Gaussian processes whose domains are Riemannian manifolds. Recall that to define such a Gaussian process, we need to define its kernel, which is a positive semi-definite function $k : X \times X \to \mathbb{R}$.

Before attempting to do so in generality, consider first how to extend the Euclidean squared exponential kernel. The simplest idea one can consider is to replace the Euclidean distance $\|x - x'\|$ with the Riemannian *geodesic distance* $d_g(x, x')$, defined as the length of the shortest path between x and x'. To ensure d_g behaves like the Euclidean distance, assume that X is *complete* with respect to d_g, meaning that all sequences which eventually become close in d_g also converge with respect to X. Consider

$$\sigma^2 \exp\left(-\frac{d_g(x, x')^2}{2\kappa^2} \right) \tag{3.6}$$

as a candidate kernel. We must then ask: is this expression necessarily well-defined? In particular, is it positive semi-definite for all κ?

In the Euclidean setting, we can prove positive-semi-definiteness by first proving it for linear kernels, then showing sums, products, and limits of kernels are positive semi-definite, thereby constructing the kernel piece-by-piece. The argument clearly cannot extend to the manifold setting, where there are no linear kernels. It turns out that no argument can, because in the manifold setting the corresponding claim isn't true.

Result 3.5. *Let (X, g) be a complete Riemannian manifold. If the geodesic squared exponential kernel is positive semi-definite for all $\kappa > 0$, then X is isometric to a Euclidean space.*

Proof. Feragen et al. [38], Theorem 2. ■

It turns out that even more can be said. In a metric space (X, d), the *length* of a path $\gamma : [0, L] \to X$ is defined as the least upper bound on the total distance between finite sets of successive points along the curve. A path is called *geodesic* between x and x' if $\gamma(0) = x$, $\gamma(L) = x'$, and $d(\gamma(t), \gamma(t')) = |t - t'|$ for all $t, t' \in [0, L]$. A metric space is called a *geodesic space* if every pair of points is connected by a geodesic.

Result 3.6. *Let (X, d) be a geodesic space. If the geodesic squared exponential kernel is positive semi-definite for all $\kappa > 0$, then X is flat in the sense of Alexandrov.*

Proof. Feragen et al. [38], Theorem 2. ■

See Villani [117], Chapter 26 for a definition of flatness in the above sense, and a discussion on its interpretation and relationship to various notions of curvature. Complete Riemannian manifolds are geodesic spaces, but the former result is sharper than the latter: in particular, the torus equipped with the product metric is flat, but is not isometric to a Euclidean space.

These results are an absolute disaster for the geodesic squared exponential kernel, and give good reason to completely abandon this approach. The fundamental issue is that there are few useful tools for proving positive-semi-definiteness of geodesic kernels, and, in light of the above results, it isn't obvious which functions are going to be positive semi-definite in the first place and which are not. We therefore explore the alternative, differential-equation-based approach mentioned in the preceding section.

3.1.4. Stochastic partial differential equations. We now develop an appropriate formalism for defining Gaussian processes on Riemannian manifolds. Rather than building such processes by defining kernels, loosely speaking, we construct them directly as affine maps of white noise processes, which we think of as infinite-dimensional standard Gaussians. This yields positive semi-definite kernels implicitly defined as covariances of said processes.

We begin by discussing key notions of abstract Gaussian processes defined on general vector spaces. Specifically, we work with Gaussian processes in the sense of duality whose test functionals are determined by a Hilbert space.

Generalized Gaussian field

Definition 3.7. *A* CENTERED GENERALIZED GAUSSIAN FIELD f *over a Hilbert space* H *is a stochastic process* $f : \Omega \times H \to \mathbb{R}$ *satisfying two key properties.*

1. $\mathbb{E}(f(\cdot, h)) = 0$ *for all* $h \in H$.
2. *There exists a bounded linear self-adjoint non-negative operator* $\mathcal{K}$ *on* H*, called the* COVARIANCE OPERATOR, *such that*
$$\mathbb{E}(f(\cdot, h)f(\cdot, h')) = \langle \mathcal{K}h, h' \rangle \tag{3.7}$$
for all $h, h' \in H$.

We take the right-hand-sides of our stochastic partial differential equations to be such stochastic processes. Before continuing, we prove that if H is a reproducing kernel Hilbert space, then generalized Gaussian fields can be reinterpreted as Gaussian processes in the classical sense.

Proposition 3.8. *Let* $f : \Omega \times H \to \mathbb{R}$ *be a centered generalized Gaussian field defined over a reproducing kernel Hilbert space* H *with identity covariance operator* $\operatorname{id} : H \to H$. *Then letting* $\operatorname{ev}_x \in H^*$ *be a pointwise evaluation functional and* $\Psi : H \to H^*$ *be the bijective linear isometry given by the Riesz Representation Theorem, the stochastic process* $f : \Omega \times X \to \mathbb{R}$ *defined by* $f(\omega, x) = f(\omega, \Psi^{-1} \operatorname{ev}_x)$ *is a Gaussian process whose covariance is given by the reproducing kernel of* H.

Proof. It is clear that the resulting map is a centered Gaussian process, so it suffices to compute its covariance. Let $\operatorname{ev}_x \in H^*$ and $\operatorname{ev}_{x'} \in H^*$ be pointwise evaluation functionals. Then

$$\begin{aligned}\operatorname{Cov}(f(\cdot,x),f(\cdot,x')) &= \operatorname{Cov}(f(\cdot,\Psi^{-1}\operatorname{ev}_x),f(\cdot,\Psi^{-1}\operatorname{ev}_{x'})) && (3.8)\\ &= \left\langle \Psi^{-1}\operatorname{ev}_x,\Psi^{-1}\operatorname{ev}_{x'}\right\rangle_H && (3.9)\\ &= \langle \operatorname{ev}_x,\operatorname{ev}_{x'}\rangle_{H^*} && (3.10)\\ &= k(x,x') && (3.11)\end{aligned}$$

using the reproducing property, and the claim follows. ■

Note that the resulting Gaussian process in general will *not* have sample paths in H almost surely, not even up to a choice of version. Sample paths will instead generally lie in a less regular function space—this subtlety provides much of the motivation behind introducing generalized Gaussian fields in the first place, rather than working purely in terms of sample paths. In the other direction, if k is a kernel, define the *covariance operator* by

$$\mathcal{K} : \phi \mapsto \int_X \phi(x)k(x,\cdot)\,\mathrm{d}x. \tag{3.12}$$

One can see that a centered Gaussian process $f : \Omega \times X \to \mathbb{R}$ induces a centered generalized Gaussian field $f : \Omega \times H \to \mathbb{R}$ with covariance operator $\mathcal{K}$, provided that H is chosen appropriately. For instance, if f is regular enough that its samples lie in $L^2(X)$ almost surely, one can take $H = L^2(X)$ and $\mathcal{K}$ as above. This clarifies how this general notion relates to Gaussian processes in the standard sense. To continue, we introduce the notion of a *solution* of a stochastic partial differential equation.

Stochastic partial differential equation

Definition 3.9. *Let H be a Hilbert space, let $\mathcal{L} : F \to H$ be a bounded linear operator, and let $\mathcal{W}$ be a centered generalized Gaussian field on H. Then the zero-mean generalized Gaussian field f over F is a solution of the abstract stochastic partial differential equation*

$$\mathcal{L}f = \mathcal{W} \tag{3.13}$$

if, letting $\mathcal{L}^*$ be the adjoint of $\mathcal{L}$, for every $h \in H$ we have that

$$f(\omega, \mathcal{L}^*h) = \mathcal{W}(\omega, h) \tag{3.14}$$

holds almost surely.

For our purposes, it also suffices to replace the almost sure equality with equality in distribution. Even if one considers this weaker notion, the same kind of results and calculations follow in our setting, and work equally well in both cases. We therefore do not dwell on this distinction.

The idea behind this definition is to take advantage of Gaussianity and use it to avoid even attempting to construct a pathwise solution theory on random variables. Instead, the operator $\mathcal{L}$ is thought of as a map between the associated Hilbert spaces: if one or both are function spaces admitting reproducing kernels, the generalized Gaussian fields respectively yield Gaussian processes in the classical sense. In the given setting, this solution concept ends up being useful, thanks to the following general result.

Solution of a stochastic partial differential equation

Result 3.10. *If $\mathcal{L}$ is invertible, then*

$$f(\omega, h) = \mathcal{W}(\omega, \mathcal{L}^{-*} h) \tag{3.15}$$

is the unique solution to $\mathcal{L}f = \mathcal{W}$.

Proof. Lototsky and Rozovsky [75], Theorem 4.2.2. ■

It is remarkable that such a minimalist solution theory, which relies very fundamentally on the fact that Gaussian processes are uniquely determined by their associated reproducing kernel Hilbert spaces or generalizations thereof, gives a description concrete enough for our purposes. Indeed, this result generally determines but does not reveal what function space f lies in as a random variable, along with its regularity properties such as continuity—however, for Bayesian learning, we don't actually need these.

For this construction to yield a Gaussian process in the classical sense, we should ensure the space f lies in is a reproducing kernel Hilbert space, since this lets us build the Gaussian processes of interest from pointwise evaluation functionals as described previously. The main challenge, then, is finding appropriate spaces and operators in order to apply the preceding ideas.

3.1.5. The Riemannian Matérn kernel. We now use the above results to define Riemannian Gaussian processes directly, and, following this, calculate their kernels in order to obtain workable numerical expressions. To proceed, we need to make concrete choices for which Hilbert spaces and operators to use. To do so, we study the class of equations considered by Whittle [120, 121] and Lindgren et al. [72], which we now review.

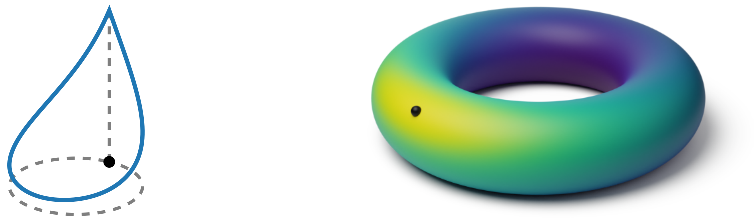


Figure 3.4. The Matérn-1/2 kernel $k_{1/2}(x, \cdot)$ defined on a circle and torus, where the point x is marked by a black dot. The kernel on the circle is computed using the Poisson summation formula, which relates Laplacian eigenvalues and eigenfunctions on the circle with those on the real line, and therefore also relates kernels on the circle with kernels on the real line.

Suppose temporarily that $X = \mathbb{R}^d$ is Euclidean. In that setting, Whittle [120, 121] has shown that, if we suppose that f is, in a purely formal sense, a solution to the stochastic partial differential equation

$$\left(\frac{2\nu}{\kappa^2} - \Delta\right)^{\frac{\nu}{2}+\frac{d}{4}} f = \mathcal{W} \tag{3.16}$$

Matérn kernel

then its covariance kernel must be the *Matérn kernel*

$$k_\nu(x, x') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\|x - x'\|}{\kappa}\right)^\nu K_\nu\left(\sqrt{2\nu}\frac{\|x - x'\|}{\kappa}\right) \tag{3.17}$$

where Γ is the gamma function and K_ν is the modified Bessel function of the second kind [45]. This kernel is very well-studied and among the most widely-used Euclidean kernels in practice. Building on this idea, Lindgren et al. [72] proposed a formalism based on Galerkin finite element analysis for giving precise meaning to such calculations, including treatment of boundary conditions and other considerations.

Riemannian Matérn kernel

In particular, Lindgren et al. [72] define the *Riemannian Matérn kernel* to be the covariance kernel of the solutions of analogs of the above stochastic partial differential equation, where $\mathbb{R}^d$ is replaced with a Riemannian manifold X, and the Euclidean Laplacian Δ is replaced by the Laplace–Beltrami operator Δ_g. The resulting kernel, shown in Figures 3.4 and 3.5, is well-defined, but implicit: Lindgren et al. [72] provide techniques for training the resulting processes by solving the stochastic partial differential equation numerically.

We develop an alternative, more constructive formalism. This is done by improving on the above prior results in two ways: (1) we bypass finite element

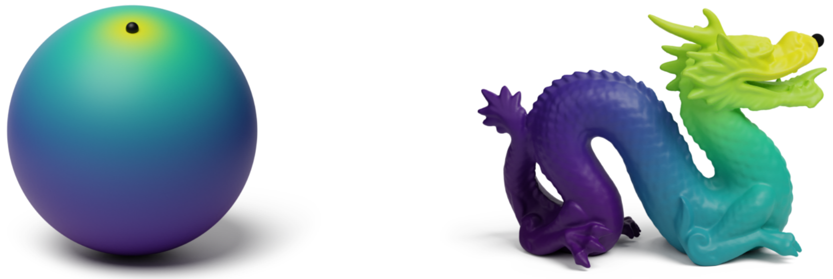


Figure 3.5. The Matérn-1/2 kernel $k_{1/2}(x, \cdot)$ defined on a sphere and dragon manifold, where the point x is marked by a black dot.

analysis by instead working with the previously-introduced theory of Gaussian stochastic partial differential equations described by Lototsky and Rozovsky [75], and (2) we deduce numerical expressions for calculating the kernels of the resulting processes, enabling them to be trained using standard methods. To start, we define the right-hand-side of our equations.

Riemannian white noise

Definition 3.11. *Define the* WHITE NOISE PROCESS $\mathcal{W}_g : \Omega \times L^2(X) \to \mathbb{R}$ *to be a centered generalized Gaussian field with identity covariance operator* $\mathrm{id} : L^2(X) \to L^2(X)$*, where we recall that the inner product on* $L^2(X)$ *is defined by integration against the Riemannian volume measure. By applying Kolmogorov's Extension Theorem to a family of finite-dimensional marginals indexed by* $L^2(X)$*, we conclude such a process exists and is well-defined.*

This stochastic process *cannot* be viewed as a scalar-valued random function. Though Kolmogorov's Extension Theorem does imply there is a random variable $\mathcal{W}_g : \Omega \to \mathbb{R}^{L^2(X)}$ whose finite-dimensional marginals coincide with $\mathcal{W}_g : \Omega \times L^2(X) \to \mathbb{R}$, this is next-to-useless because $\mathbb{R}^{L^2(X)}$ is highly irregular and we can say little about which subspace $\mathcal{W}_g$ concentrates on without additional considerations that we may conveniently avoid.

Next, we define the left-hand-side of the stochastic partial differential equations under study, including the Hilbert spaces and operators $\mathcal{L}$ of interest using functional calculus. For this, a result on Riemannian reproducing kernel Hilbert spaces will be of key interest.

Riemannian Sobolev and diffusion spaces

Result 3.12. *Define the Riemannian Sobolev space $H^s(X)$ by*

$$H^s(X) = \left\{ f \in D'(X) : f = (1 - \Delta_g)^{-s/2} h : h \in L^2(X) \right\} \tag{3.18}$$

and the Riemannian diffusion space $\mathcal{H}^s(X)$ by

$$\mathcal{H}^s(X) = \left\{ f \in D'(X) : f = e^{\frac{s}{2}\Delta_g} h : h \in L^2(X) \right\} \tag{3.19}$$

where the operators are defined using functional calculus. Then $H^s(X)$ with $s > \frac{d}{4}$ and $\mathcal{H}^s(X)$ with $s > 0$ are reproducing kernel Hilbert spaces.

Proof. De Vito et al. [29], Theorem 3 and Theorem 6. ■

This gives the key technical pillar upon which our calculations rest. It both provides appropriate Hilbert spaces to use within the solution theory, and, by virtue of admitting reproducing kernels, guarantees that they are spaces of actual functions $f : X \to \mathbb{R}$ on the Riemannian manifold. This enables us to plug pointwise evaluation functionals into the obtained generalized Gaussian field, yielding Riemannian Gaussian processes in the classical sense—the objects we sought to construct in the first place.

Riemannian squared exponential kernel

Note that the operators $e^{\frac{s}{2}\Delta_g}$ can be viewed as limits of appropriately rescaled versions of the operators $(1 - \Delta_g)^{-s/2}$. Given that the Euclidean Matérn kernel converges to the Euclidean squared exponential kernel as $\nu \to \infty$, we therefore view stochastic partial differential equations induced by $e^{\frac{s}{2}\Delta_g}$ as giving rise to the *Riemannian squared exponential kernel*—we make this perspective precise shortly.

Generalized spectral measure

Our strategy will be to represent the kernels of interest as infinite sums of Laplace–Beltrami eigenfunctions. Here, compactness of X is key: this ensures that the total number of eigenfunctions is countable, so that summing eigenfunctions makes sense. The coefficients in the sum can be viewed as a kind of *generalized spectral measure*, which, in our case, owing to compactness, is supported on the non-negative integers rather than on $\mathbb{R}^d$. This shows one way in which geometry of spaces is reflected in properties of kernels.

To carry the necessary calculations out and compute the kernels of the Gaussian processes defined by our stochastic partial differential equations, we need to relate the Sobolev and diffusion spaces of De Vito et al. [29] with the equations studied by Whittle [120, 121] and Lindgren et al. [72]. This gives series expansions of the kernels in terms of Laplace–Beltrami eigenpairs. Convergence rates of these series depend on the eigenvalue growth rate, which is quantified by Weyl's law [126]. We now state and prove the main result.

Theorem 3.13. *Let X be a compact Riemannian manifold. For $\nu > 0$ and $\kappa > 0$, define the stochastic partial differential equations*

Riemannian Matérn and squared exponential kernels

$$\left(\frac{2\nu}{\kappa^2} - \Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} f = \mathcal{W}_g \qquad\qquad e^{-\frac{\kappa^2}{4}\Delta_g} f = \mathcal{W}_g \tag{3.20}$$

where, respectively, we have

$$\left(\frac{2\nu}{\kappa^2} - \Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} : H^{\nu+\frac{d}{2}}(X) \to L^2(X) \tag{3.21}$$

$$e^{-\frac{\kappa^2}{4}\Delta_g} : \mathcal{H}^{\frac{\kappa^2}{2}}(X) \to L^2(X). \tag{3.22}$$

Then, letting (λ_n, f_n) be the eigenvalues and eigenfunctions of the Laplace–Beltrami operator, in both cases the unique solutions f are Gaussian processes with absolutely convergent respective covariance kernels

$$k(x,x') = \sum_{n=0}^{\infty} \left(\frac{2\nu}{\kappa^2} + \lambda_n\right)^{-\nu-\frac{d}{2}} f_n(x) f_n(x') \tag{3.23}$$

$$k(x,x') = \sum_{n=0}^{\infty} e^{-\frac{\kappa^2}{2}\lambda_n} f_n(x) f_n(x') \tag{3.24}$$

which we call the Riemannian Matérn *and* Riemannian squared exponential *kernels.*

Proof. Note first that the operators corresponding to the Matérn kernel coincide with those used by De Vito et al. [29] in defining the Sobolev spaces of interest if we have a fixed length scale given by $\kappa = \sqrt{2\nu}$. Similarly, the operators corresponding to the squared exponential kernel always coincide. In this setting, the reproducing kernels are given by De Vito et al. [29], Proposition 2 as

$$k(x,x') = \sum_{n=0}^{\infty} \left(\frac{2\nu}{\kappa^2} + \lambda_n\right)^{-\nu-\frac{d}{2}} f_n(x) f_n(x') \tag{3.25}$$

$$k(x,x') = \sum_{n=0}^{\infty} e^{-\frac{\kappa^2}{2}\lambda_n} f_n(x) f_n(x') \tag{3.26}$$

which are shown to be absolutely convergent in that work. This proves the claim for the squared exponential case. However, we are interested in general length scales $\kappa > 0$, so this does not suffice for the Matérn case. To extend this, the idea will be to let $\tilde{g} = \frac{2\nu}{\kappa^2} g$ be a rescaled metric tensor on X, and define the equations

$$\left(\frac{2\nu}{\kappa^2} - \Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} f = \mathcal{W}_g \qquad (1 - \Delta_{\tilde{g}})^{\frac{\nu}{2}+\frac{d}{4}} \tilde{f} = \mathcal{W}_{\tilde{g}} \tag{3.27}$$

for which we would like to show that $f = \left(\frac{\kappa^2}{2\nu}\right)^{\frac{\nu}{2}+\frac{d}{2}} \tilde{f}$. To begin, we first prove these operators are well-defined, by checking that they are bounded and invertible for all positive ν and κ. By Result 3.12 we know that for every $f \in H^{\nu+\frac{d}{2}}(X)$, there is an $h \in L^2(X)$ such that $f = (1 - \Delta_g)^{-\frac{\nu}{2}-\frac{d}{4}} h$. Moreover, both f and h can be expressed in the orthonormal basis given by Laplacian eigenfunctions f_n as

$$h(x) = \sum_{n=0}^{\infty} \alpha_n f_n \qquad f(x) = \sum_{n=0}^{\infty} \left(\frac{1}{1+\lambda_n}\right)^{\frac{\nu}{2}+\frac{d}{4}} \alpha_n f_n \tag{3.28}$$

where the expression for f follows by applying the operator $(1 - \Delta_g)^{-\frac{\nu}{2}-\frac{d}{4}}$ to the eigenfunctions. Finally, note that since $\lambda_n \geq 0$ we have

$$\min\left(\frac{2\nu}{\kappa^2}, 1\right) \leq \frac{\frac{2\nu}{\kappa^2} + \lambda_n}{1 + \lambda_n} \leq \max\left(1, \frac{2\nu}{\kappa^2}\right). \tag{3.29}$$

Using these identities, write

$$\begin{aligned} \left\| \left(\frac{2\nu}{\kappa^2} - \Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} f \right\|_{L^2(X)}^2 &= \left\| \sum_{n=0}^{\infty} \left(\frac{\frac{2\nu}{\kappa^2} + \lambda_n}{1+\lambda_n}\right)^{\frac{\nu}{2}+\frac{d}{4}} \alpha_n f_n \right\|_{L^2(X)}^2 & (3.30) \\ &\leq \sum_{n=0}^{\infty} \max\left(1, \frac{2\nu}{\kappa^2}\right)^{\nu+\frac{d}{2}} \alpha_n^2 & (3.31) \\ &= \max\left(1, \frac{2\nu}{\kappa^2}\right)^{\nu+\frac{d}{2}} \|f\|_{H^{\nu+\frac{d}{2}}(X)}^2 & (3.32) \end{aligned}$$

and similarly

$$\begin{aligned} \left\| \left(\frac{2\nu}{\kappa^2} - \Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} f \right\|_{L^2(X)}^2 &\geq \sum_{n=0}^{\infty} \min\left(\frac{2\nu}{\kappa^2}, 1\right)^{\nu+\frac{d}{2}} \alpha_n^2 & (3.33) \\ &= \min\left(\frac{2\nu}{\kappa^2}, 1\right)^{\nu+\frac{d}{2}} \|f\|_{H^{\nu+\frac{d}{2}}(X)}^2 & (3.34) \end{aligned}$$

where we have also used $L^2(X)$-orthonormality of f_n as well as

$$\sum_{n=0}^{\infty} \alpha_n^2 = \|h\|_{L^2(X)}^2 = \|f\|_{H^{\nu+\frac{d}{2}}(X)}^2 \tag{3.35}$$

which follows from orthonormality of f_n along with the definition of h and Result 3.12. This proves boundedness and invertibility. To complete the argument, we check that the desired identity holds under a change of metric. The change of metric expressions of interest are given by

$$\Delta_{\tilde{g}} = \frac{\kappa^2}{2\nu}\Delta_g \quad \tilde{\lambda}_n = \frac{\kappa^2}{2\nu}\lambda_n \quad \tilde{f}_n = \left(\frac{2\nu}{\kappa^2}\right)^{-\frac{d}{4}} f_n \quad \operatorname{vol}_{\tilde{g}} = \left(\frac{2\nu}{\kappa^2}\right)^{\frac{d}{2}} \operatorname{vol}_g \tag{3.36}$$

thus starting from $(1-\Delta_{\tilde{g}})^{\frac{\nu}{2}+\frac{d}{4}}\tilde{f} = \mathcal{W}_{\tilde{g}}$ we can write

$$(1-\Delta_{\tilde{g}})^{\frac{\nu}{2}+\frac{d}{4}} = \left(1 - \frac{\kappa^2}{2\nu}\Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} = \left(\frac{\kappa^2}{2\nu}\right)^{\frac{\nu}{2}+\frac{d}{4}}\left(\frac{2\nu}{\kappa^2} - \Delta_g\right)^{\frac{\nu}{2}+\frac{d}{4}} \tag{3.37}$$

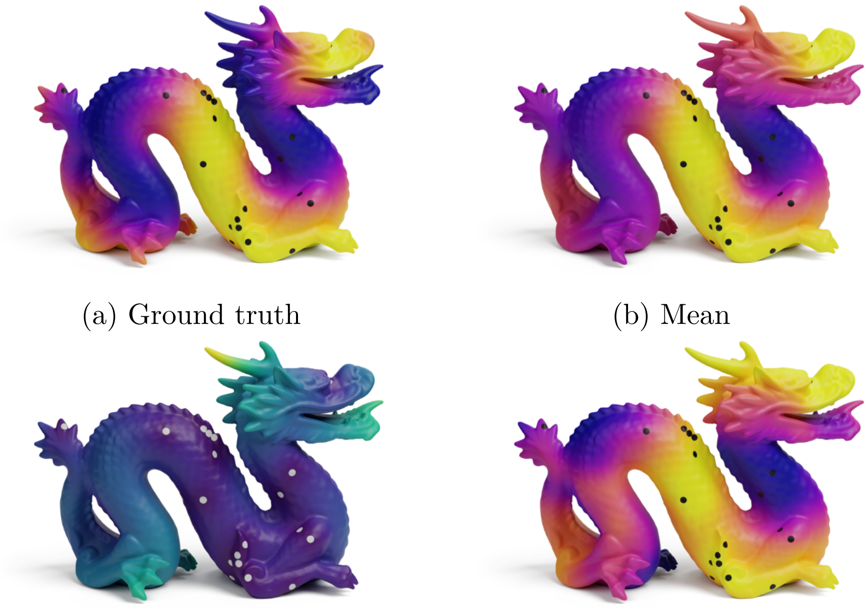
as well as

$$\mathcal{W}_{\tilde{g}} = \left(\frac{2\nu}{\kappa^2}\right)^{\frac{d}{4}} \mathcal{W}_g \tag{3.38}$$

which when combined gives $f = \left(\frac{\kappa^2}{2\nu}\right)^{\frac{\nu}{2}+\frac{d}{2}}\tilde{f}$. On the other hand, under a change of metric, the series representation of the Matérn kernel is

$$\begin{aligned}
\tilde{k}(x,x') &= \sum_{n=0}^{\infty}(1+\tilde{\lambda}_n)^{-\nu-\frac{d}{2}}\tilde{f}_n(x)\tilde{f}_n(x') && (3.39)\\
&= \sum_{n=0}^{\infty}\left(1+\frac{\kappa^2}{2\nu}\lambda_n\right)^{-\nu-\frac{d}{2}}\left(\frac{2\nu}{\kappa^2}\right)^{-\frac{d}{2}} f_n(x)f_n(x') && (3.40)\\
&= \left(\frac{\kappa^2}{2\nu}\right)^{-\nu-d}\sum_{n=0}^{\infty}\left(\frac{2\nu}{\kappa^2}+\lambda_n\right)^{-\nu-\frac{d}{2}} f_n(x)f_n(x') && (3.41)
\end{aligned}$$

where $\tilde{k}$ is the transformed kernel, for which we see that $\left(\frac{\kappa^2}{2\nu}\right)^{\nu+d}\tilde{k}(x,x') = k(x,x')$. But this exactly matches the covariance kernel of f according to the given transformation of the Gaussian process, and the claim follows. ■



(a) Ground truth

(b) Mean

(c) Standard deviation

(d) One posterior sample

Figure 3.6. A posterior Matérn Gaussian process on the dragon manifold. We plot true function values, along with the posterior mean and standard deviation. Here, the black dots represent data. We observe that the standard deviation generally increases as we move away from the training locations.

3.1.6. Illustrated examples. Using the developed tools, we have defined Riemannian Gaussian processes of interest as solutions of stochastic partial differential equations, and derived expressions for their kernels which are explicit enough to be amenable to approximation. Here, we explore using these processes as priors in Bayesian learning. Our goal is to understand how complicated geometry affects posterior uncertainty estimates.

We focus on the dragon manifold from the Stanford 3D scanning repository [27], approximated numerically as a mesh. This mesh was chosen because it has no boundary and is geometrically more complex than other alternatives. We work with the largest connected component of the mesh, which has a total of 100 179 vertices and 201 010 triangular faces.

On the mesh, discrete analogs of all of the differential-geometric notions we require are available, with reasonably well-understood approximation properties [26]. We obtain 500 discretized Laplace–Beltrami eigenpairs numerically in finite element space using the *Firedrake* partial differential equation package [90]. This, in turn, gives expressions for evaluating the kernel and sampling the prior at any points on the manifold.

To generate training data, we define a ground truth function given by the sine of the geodesic distance from a distinguished point, chosen to be the first vertex on the mesh, which is located at the end of the dragon's snout. We observe this function at a total of 52 points chosen from the mesh's vertices. We train a Matérn prior with smoothness $\nu = 3/2$ on this data using gradient descent, obtaining learned variance and length scale hyperparameters. Finally, we obtain the posterior samples over the entire mesh using pathwise sampling.

Results are given in Figure 3.6. We observe that the both the mean predictions and uncertainty estimates adapt to the manifold's geometry. In particular, we see that uncertainty increases when moving away from the data. We see that the two upper and lower parts of the dragon's snout have different uncertainty estimates, in spite of the fact that they are close in ambient Euclidean distance. Overall, we conclude that the geometric Matérn models used produce uncertainty estimates which reflect the manifold's geometry.

From the computational tools used, one can see that there is more to say about the developed model class. In particular, the discrete analogs of the Laplace–Beltrami operator used within the finite element computations give rise to discrete Gaussian processes in their own right. This leads one to ask: are there interesting analogs of Matérn models in useful classes of discrete spaces? We now proceed to develop one such analog.

3.2. Graph Matérn Gaussian processes

A general maxim of geometry is that anything one can do with a manifold, one can do with a graph, given some thinking—but, the details and behavior might be a bit different. In the preceding section, we built Gaussian processes using spectral properties of the Riemannian Laplace–Beltrami operator. We now ask: can we repeat these constructions on graphs, obtaining Gaussian processes that can be used in very different settings?

3.2.1. Review of graph theory. Graphs are discrete spaces which formalize and describe interconnected networks of points. For an introduction to the aspects of graph theory of key interest for our purposes, see Spielman [105]. Here we briefly review the key notions.

Nodes, weights, edges

A weighted undirected graph consists of a finite set of *nodes*, each of which is assigned a positive real number called the *weight*, and *edges* between nodes. Graphs with unit weights can be visualized by drawing a set of points on a two-dimensional plane or in three-dimensional space, representing the nodes, and drawing lines between the points, representing the edges.

Graphs and matrices

Adjacency matrix

Degree matrix

A broad theme in graph theory is that geometric properties of graphs can be encoded as finite-dimensional vectors and matrices by picking an arbitrary ordering of nodes, and associating nodes with rows and columns of a matrix. Define the *weighted adjacency matrix* $\mathbf{W}$ by letting the matrix entry corresponding to two nodes be their respective edge weight. Similarly, define the diagonal *degree matrix* $\mathbf{D}$ by $D_{ii} = \sum_j W_{ij}$—for graphs with unit weights, this counts how many neighbors each node has.

Functions on graphs

Permutation invariance and equivariance

We can construct *functions on graphs*, which map each node into some space of interest, in a number of ways. For defining functions which depend only on neighboring nodes, working with matrices induced by the graph is often useful. When doing so, we must take care that the functions constructed do not depend on the arbitrary choice of ordering used to associate nodes with matrix rows and columns. Algebraically, this is encoded through *permutation invariance*, *permutation equivariance*, and other similar requirements.

Discretizations of manifolds

Graphs often arise as discretizations of manifolds: there are many ways of making this idea precise. For example, one can consider an infinite sequence of nested finite subsets of the manifold converging monotonically to a countable dense subset thereof. This defines a sequence of graphs by taking nodes to be the elements of each sets, and connecting neighboring nodes. Sufficiently well-behaved such sequences give rise to vector spaces of functions converging monotonically to an appropriate function space on the manifold.

3.2.2. The graph Laplacian. In the Riemannian setting, our strategy for building Gaussian process models essentially amounted to using functional calculus defined using spectral properties of the Laplace–Beltrami operator to construct the Gaussian processes of interest as solutions of stochastic partial differential equations. This strategy depended on the manifold only through the Laplace–Beltrami operator, and the white noise process, which suggests that it may also work for other classes of spaces.

Graph Laplacian

Weighted directed graphs in particular admit the notion of a *graph Laplacian*, which is a symmetric positive semi-definite matrix defined as

$$\mathbf{\Delta} = \mathbf{D} - \mathbf{W} \tag{3.42}$$

where $\mathbf{D}$ is the degree matrix and $\mathbf{W}$ is the weighted adjacency matrix. The graph Laplacian can be interpreted as a linear operator acting the space of all functions $f : G \to \mathbb{R}$ where G is the set of nodes. Each such function can be represented as a $|G|$-dimensional vector $\boldsymbol{f}$ by assigning an ordering to the nodes, as described previously. From this viewpoint, the graph Laplacian is

$$(\boldsymbol{\Delta}\boldsymbol{f})(x) = \sum_{x' \sim x} f(x) - f(x') \tag{3.43}$$

where x is a node, and sum is taken over all neighboring nodes x' of x. This expression now resembles its Euclidean and Riemannian analogs in a much more direct manner, and justifies why one would call $\boldsymbol{\Delta}$ a Laplacian in the first place. First, though, we clarify the choice of sign in this expression.

Sign convention

Remark 3.14. *Following standard practice in graph theory, we adopt a* DIFFERENT SIGN CONVENTION *for the graph Laplacian compared to its Euclidean and Riemannian analogs. One should thus view the operators*

$$\underbrace{\boldsymbol{\Delta}}_{\textit{no minus sign}} \qquad\qquad \underbrace{-\Delta}_{\textit{minus sign}} \tag{3.44}$$

as analogues of one-another—in particular, both are positive semi-definite. NOTE THE DIFFERENT MINUS SIGNS! *This corresponds to adopting the analyst's rather than geometer's convention for studying* Δ.

Functional calculus

In the graph case, we can also develop a notion of *functional calculus* just as we developed previously. This time, however, doing so is mathematically more-or-less trivial: for a function $\Phi : \mathbb{R} \to \mathbb{R}$ and a diagonal matrix $\boldsymbol{\Lambda}$, let $\Phi(\boldsymbol{\Lambda})$ be the matrix obtained by applying Φ to the diagonal. Then, letting $\boldsymbol{\Delta} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ be the eigenvalue decomposition of $\boldsymbol{\Delta}$, which by positive semi-definiteness always exists, define

$$\Phi(\boldsymbol{\Delta}) = \mathbf{U}\Phi(\boldsymbol{\Lambda})\mathbf{U}^T. \tag{3.45}$$

This gives a notion of functional calculus analogous to the one considered previously in the Riemannian case, only without the need to introduce any mathematical theory beyond elementary eigenvalue factorizations. In particular, since all matrices are finite, we do not need to consider any kind of convergence. We now examine Gaussian processes from this perspective.

3.2.3. The graph Matérn kernel. Our goal now is to construct Gaussian processes $f : G \to \mathbb{R}$ where G is the set of nodes of a weighted undirected graph. We'd like to define processes whose covariance reflects the structure of the graph. To do this, we adapt the notions of Matérn and squared exponential Gaussian processes studied previously to the graph setting.

Recall that in the Euclidean and Riemannian cases, these processes were defined as solutions of stochastic partial differential equations with left-hand-sides defined using functional calculus, and right-hand-sides consisting of white noise processes. Adapting this definition by replacing $-\Delta$ with $\mathbf{\Delta}$ and dropping dimension-dependent terms from exponents gives

$$\left(\frac{2\nu}{\kappa^2} + \mathbf{\Delta}\right)^{\frac{\nu}{2}} \boldsymbol{f}(\omega) = \boldsymbol{\mathcal{W}}(\omega) \qquad\qquad e^{\frac{\kappa^2}{4}\mathbf{\Delta}} \boldsymbol{f}(\omega) = \boldsymbol{\mathcal{W}}(\omega) \tag{3.46}$$

where $\boldsymbol{\mathcal{W}} \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$ are standard Gaussians, and $\boldsymbol{f} : \Omega \to \mathbb{R}^{|G|}$ are the random vectors defining the stochastic processes $f : \Omega \times G \to \mathbb{R}$ defined on the graph's nodes. Note that the numbers $\frac{2\nu}{\kappa^2}$ are *not* added element-wise to the graph Laplacian—instead, following the established conventions, they are added to its *eigenvalues*. By elementary algebra, the *graph Matérn Gaussian processes* and *graph squared exponential Gaussian processes* are given by

Graph Matérn Gaussian processes

$$\boldsymbol{f} \sim \mathrm{N}\left(\mathbf{0}, \left(\tfrac{2\nu}{\kappa^2} + \mathbf{\Delta}\right)^{-\nu}\right) \qquad\qquad \boldsymbol{f} \sim \mathrm{N}\left(\mathbf{0}, e^{-\frac{\kappa^2}{2}\mathbf{\Delta}}\right) \tag{3.47}$$

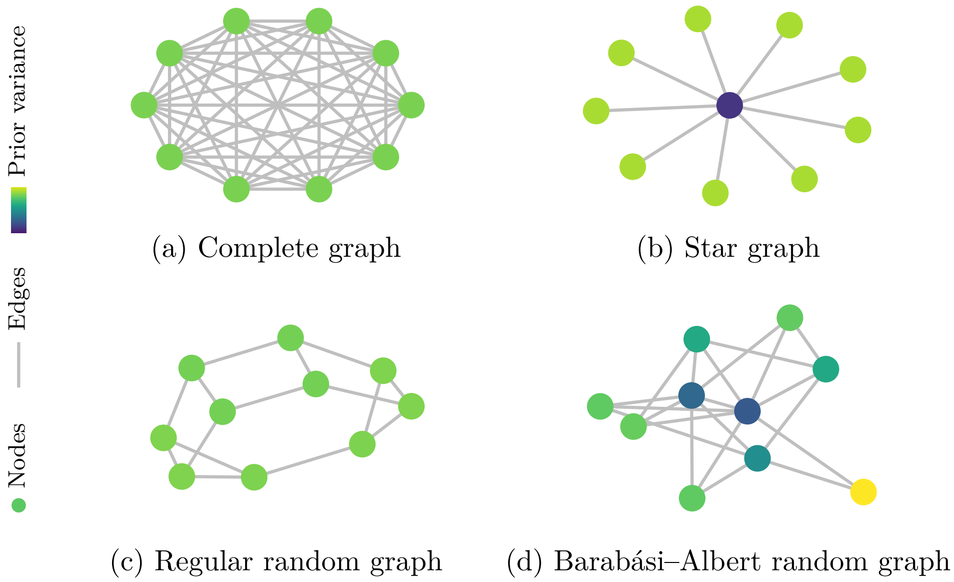
This defines the graph Matérn and squared Gaussian processes of interest. We now explore some of their properties.

Sparsity

By virtue of its definition, $\mathbf{\Delta}$ inherits *sparsity* properties from graphs. Thus, for sufficiently small integers ν and many graphs, the matrices $(\frac{2\nu}{\kappa^2} + \mathbf{\Delta})^{\frac{\nu}{2}}$ will be sparse. If the precise sparsity pattern is sufficiently well-behaved—for instance in certain planar graphs—these matrices' Cholesky factors will also be sparse, potentially reducing computational costs from cubic to linear or close to it. Alternatively, Krylov subspace methods for sparse linear systems can be applied, giving another way to leverage sparsity to improve scalability.

Non-uniform variance

Graph Matérn and graph squared exponential kernels possess *non-uniform variance*, whose precise form will vary with the graph. In particular, each node's variance is not simply a function of its degree, and instead depends on the precise geometry in a complex manner. This can be seen in Figure 3.7. Similar phenomena occur in random walk kernels studied by Urry and Sollich [115]—in that setting, variance is determined by the return time of a certain random walk defined on the graph.



(a) Complete graph (b) Star graph

(c) Regular random graph (d) Barabási–Albert random graph

Figure 3.7. Here, we show the prior variance of a graph Gaussian process at each node for a number of different graphs. Variances for all graphs are plotted using a common scale. In general, this prior variance is non-uniform and instead reflects the structure of the graph.

Symmetric normalized graph Laplacian

It's also possible to use the *symmetric normalized graph Laplacian*, which is defined as $\mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2}$, to define analogous kernels to the ones above, by using this matrix in place of the graph Laplacian. This yields the *symmetric normalized graph Matérn* and *symmetric normalized graph squared exponential* Gaussian processes. These can be preferable in domains where symmetric normalized Laplacians are customarily used.

Connection with random walks

The graph squared exponential kernel can be connected with *random walks* in a number of ways. Firstly, it is the Green's function of the graph diffusion equation. More precisely, if $\boldsymbol{\phi} : [0, \infty) \times G \to \mathbb{R}$ solves the equation

$$\dot{\boldsymbol{\phi}}_t + \mathbf{\Delta}\boldsymbol{\phi}_t = 0 \qquad\qquad \boldsymbol{\phi}_0 = \boldsymbol{v} \tag{3.48}$$

then $\boldsymbol{\phi}(\tau, \cdot) = e^{-\tau\mathbf{\Delta}}\boldsymbol{v}$, where our notation again uses the equivalence between vectors and real-valued functions on G. This equation has a strong physical interpretation: it describes heat transfer along the graph—this gives a way to understand what kind of prior information graph squared exponential kernel introduces. Similarly, if $\mathbf{\Delta}$ is replaced with the symmetric normalized graph Laplacian, then $\boldsymbol{\phi}(\tau, \cdot)$ can be interpreted as the unnormalized probability

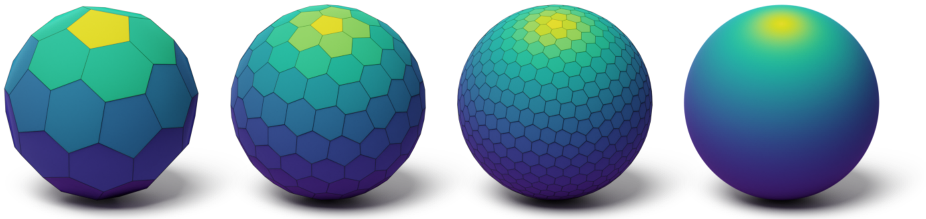


Figure 3.8. Illustration of a sequence of graph Matérn-1/2 kernels, each defined on an icosahedral graph. For every icosahedron, such a graph is defined by letting each face corresponds to a node on the graph, and neighboring faces correspond to edges on the graph. The limiting kernel for this sequence of graphs is a Riemannian Matérn-1/2 kernel on the sphere.

density of a continuous-time random walk moving along the graph.

Another way to understand the prior information contained in the given kernels is through limits. Mirroring all other settings considered, graph Matérn kernels converge to graph squared exponential kernels as $\nu \to \infty$. This is essentially immediate, since their eigenvectors coincide, and eigenvalues converge. Graph squared exponential kernels also arise as a limit of the *random walk kernel* of Smola and Kondor [99], defined using the degree matrix $\mathbf{D}$ as

$$(\mathbf{I} - (1-\alpha)\mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2})^s. \tag{3.49}$$

This kernel arises from symmetrizing the s-step transition matrix of a certain lazy random walk on a graph. Though it looks similar to the graph Matérn kernel, its structure is very different: $s > 0$ is positive rather than negative, $\alpha \in [0,1)$ is interpreted as the laziness parameter of the underlying random walk, and the Laplacian is subtracted rather than added. Still, this kernel converges to the graph squared exponential kernel: if we set $\alpha = 1 - \frac{\kappa^2}{2k}$ with κ a fixed constant, we have

$$\lim_{s\to\infty}(\mathbf{I} - (1-\alpha)\mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2})^s = e^{-\frac{\kappa^2}{2}\mathbf{D}^{-1/2}\mathbf{\Delta}\mathbf{D}^{-1/2}}. \tag{3.50}$$

This provides another view of the connection between the graph squared exponential kernel and random walk models.

Riemannian limits

Finally, the introduced graph kernels also converge to *Riemannian limits*, provided the graphs are sufficiently geometrically well-behaved and the limits are understood appropriately. In cases studied by Belkin and Niyogi [5] and Burago et al. [15], the eigenvalues and eigenvectors of the graph Laplacian converge to those of the Laplace–Beltrami operator. Using this, Sanz-Alonso and Yang [96] provide a framework for studying limits of graph Matérn kernels. One example of such a limit is shown in Figure 3.8.

An alternative way to study Riemannian limits is to embed the graph within a Euclidean space, and study vector spaces of piecewise-linear functions between neighboring nodes. Sequences of such graphs can arise as *finite element* discretizations of function spaces, which were considered previously in Chapter 2. Here, Lindgren et al. [72] show that certain graph Matérn Gaussian processes converge to their Riemannian limits.

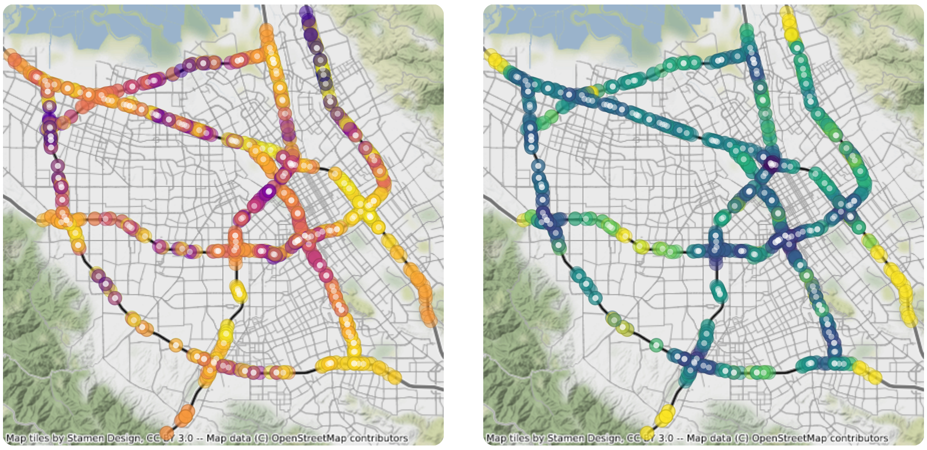
In total, these results illustrate that graph Matérn and graph squared exponential kernels are closely-connected with their Riemannian analogs, which justifies both the names they are given, and the choice of defining them using the graph Laplacian in the first place. In spite of their similarity, however, these models can be used in settings which are very different from the manifold setting that inspired them. We now illustrate a few possibilities.

3.2.4. Illustrated examples. To illustrate the graph Matérn Gaussian processes, we demonstrate their use in a setting which departs considerably from the Euclidean and Riemannian settings considered before. Our goal is to show how such models may enable applications that are very different from those in which Gaussian processes have been traditionally used.

To do so, we consider probabilistic interpolation of traffic data along a road network consisting of highways in the city of San Jose, California, obtained from OpenStreetMap [82]. In this graph, nodes are traffic sensors, and edges are roads between sensors, one for each side of the street. Edges are weighted by inverse distance. We work with the largest connected component of the graph, which has 1016 nodes and 1173 edges.

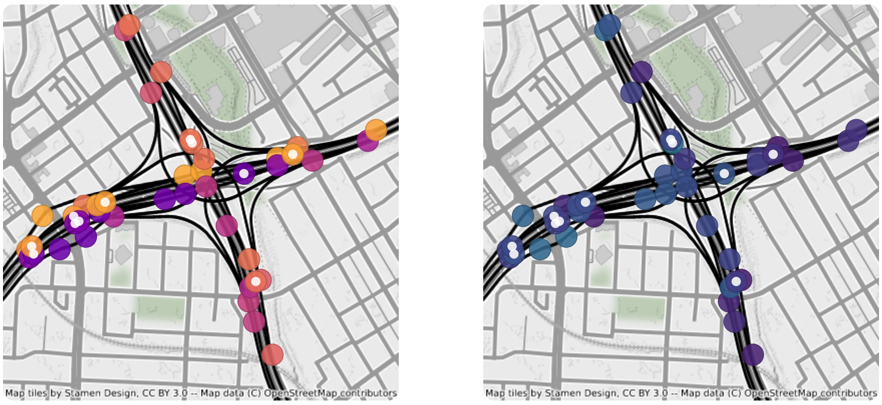
For training data, we examine traffic flow speed, which is available at 325 nodes, using 250 randomly chosen nodes for the training set and the remainder as the test set. To simplify the problem and aid visualization, we focus on a single time slice consisting of traffic on Monday at 5:30pm, and do not consider space-time behavior. We compute the kernel approximately using 500 Laplace–Beltrami eigenpairs, and train the model by optimizing kernel hyperparameters and likelihood error variance.

Results can be seen in Figures 3.9 and 3.10. Here, we see that in spite of the heavily non-manifold-like geometry present in this graph, the Gaussian process still reflects the graph's geometric structure. In particular, the width


Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

(a) Mean (b) Standard deviation

Figure 3.9. Posterior means and standard deviations, in miles per hour, for the probabilistic graph interpolation task on the road network. Nodes with white circles indicate sensor locations where data is available. We observe that standard deviation generally increases as we move away from the parts of the state space where there is data. Standard deviation values above 10 are shown clipped.


Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors
Map tiles by Stamen Design, CC BY 3.0 -- Map data (C) OpenStreetMap contributors

(a) Mean (b) Standard deviation

Figure 3.10. View of a zoomed-in portion of the posterior means and standard deviations of the probabilistic graph interpolation task on the road network. Note the small-scale variation present within the error bars.

of the error bars given by the posterior standard deviation increases as we move away from nodes where there is data. We conclude that, like in the manifold case, the Gaussian process produces uncertainty estimates which reflect the graph's structure given by the connectivity of nodes.

While this example is not particularly realistic, since we have not considered temporal interpolation nor other effects likely to be important in accurate modeling of traffic, it nonetheless illustrates that modeling heavily non-Euclidean data using Gaussian processes is possible. We hope these illustrations prompt others to imagine new and unexpected use cases for Gaussian processes. We thus end our detour and return to the manifold setting.

3.3. Gaussian vector fields on Riemannian manifolds

In the preceding sections, we studied Gaussian processes in two different non-Euclidean settings, namely those of graphs and manifolds. In both cases, the models obtained were scalar-valued. We now ask: can we use these models to define vector-valued Gaussian processes on manifolds? The first step, then, is to understand what such a notion ought to actually mean.

3.3.1. Vector fields on manifolds. A *vector field* f on a smooth manifold X is defined as a section of the tangent bundle—we recall that this is a function $f : X \to TX$ such that $\operatorname{proj}_X \circ f = \operatorname{id}_X$. This requirement is very intuitive: it says that for any point x, the tangent vector $v_x = f(x)$ must be attached to x. Vector fields on manifolds exhibit certain mathematical behaviors not present in the Euclidean case—we illustrate one of the most important kinds below.

Result 3.15. *There does not exist a nowhere-vanishing continuous section on any even-dimensional sphere.*

Corollary of Poincaré–Hopf Theorem

Proof. Lee [68], Theorem 13.32. ■

This result is also called the *hairy ball* theorem due to its interpretation: namely, it tell us in particular that if we consider a two-dimensional sphere with hair on its surface, then it is not possible to comb the hair without creating a swirl or other location where the direction of the hair changes

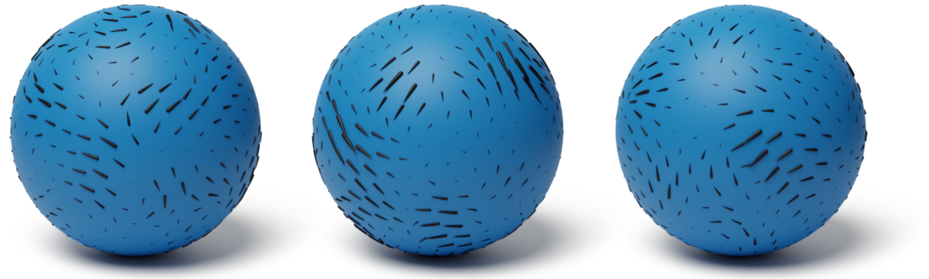


Figure 3.11. Three samples from a Gaussian vector field. Each of these samples, by virtue of being a continuous vector field on a two-dimensional sphere, automatically contains at least one location whose value is zero.

abruptly. An illustrated set of vector fields, described further in what follows, can be seen in Figure 3.11.

This result restricts the kind of technical tools available for studying vector fields on manifolds. In particular, it implies that for a generic smooth manifold, it is not possible to choose a smoothly-varying set of basis vectors in each tangent space simultaneously. Because of this, it also follows that a smooth section $f : X \to TX$ *cannot* be reinterpreted a continuous function $f : X \to \mathbb{R}^d$ in general. This makes understanding vector fields a more delicate endeavour for manifolds compared to Euclidean spaces.

We are interested in defining vector-valued Gaussian processes on manifolds. The first question that arises, then, is what should one actually mean when describing a random function $f : \Omega \times X \to TX$ as *Gaussian*? The preceding sections suggest two possible approaches.

1. Adapt the notion of *Gaussian finite-dimensional marginal distributions* to the tangent bundle setting.

2. Introduce an appropriate infinite-dimensional *vector space of sections* and work with Gaussians in the sense of duality.

Clearly, the latter approach is more elegant and more general, but it is also also more effortful and potentially less constructive. Since we are ultimately interested in working with Gaussian vector fields algorithmically, we adopt the former approach, but keep the latter view in mind to ensure soundness of our overall strategy. Recall that the vector direct sum is denoted by $\oplus$. We proceed as follows.

Definition 3.16. *Let X be a smooth manifold. We say that a random section $f : \Omega \times X \to TX$ is* GAUSSIAN *if for any finite set of locations $x_1, \ldots, x_n \in X$, the random vector $(f(\cdot, x_1), \ldots, f(\cdot, x_n)) \in T_{x_1}X \oplus \ldots \oplus T_{x_n}X$ is Gaussian in the sense of duality.*

Gaussian vector field

Thus, we see that in spite of the fact that a tangent bundle is a manifold, and not a vector space, it possesses enough linear structure to admit a sensible notion of finite-dimensional marginals. Here, we immediately see the value of the duality-based approach to Gaussianity originally developed in Chapter 1: by using this, rather than working with bases, we avoid the need for cumbersome change-of-basis consistency checks. We now examine this notion's view from the vantage point of an embedding into Euclidean space.

Proposition 3.17. *Let $i : X \to \mathbb{R}^p$ be a smooth embedding. Then*

$$f_i : \Omega \times i(X) \to \mathbb{R}^p \qquad f_i(\omega, x) = i_* f(\omega, i^{-1}(x)) \tag{3.51}$$

is a vector-valued Gaussian process in the standard sense.

Proof. Since the embedding i is bijective on its image, any set of locations $x_1^{(i)}, .., x_n^{(i)} \in i(X)$, will map uniquely onto a set of locations $x_1, .., x_n \in X$. By definition of vector pushforward maps, i_* maps tangent vectors from $T_{x_1}X \oplus .. \oplus T_{x_n}X$ into $\mathbb{R}^{n \times p}$ linearly. Since Gaussianity is preserved by linearity, every set of finite-dimensional marginals in the embedded process is Gaussian, and the claim follows. ■

This confirms that our definition of a vector-valued Gaussian process is the right one: if we embed the manifold in a Euclidean space, we obtain the correct kind of stochastic process. Figure 3.11 illustrates a set of random samples from a Gaussian vector field, constructed using the technique presented in the sequel.

As before, the mean of such a process will simply be a given vector field. The next step is to determine what is an appropriate notion of a *kernel.* In the Euclidean case, a *matrix-valued kernel* is a symmetric positive semi-definite function $k : X \times X \to \mathbb{R}^{d \times d}$. This notion more-or-less completely analogous to the scalar-valued case. A priori, this notion seems completely coordinate-dependent due to the matrix appearing in the output. To avoid this, we instead consider the map

Matrix-valued kernel

$$((x, \boldsymbol{v}), (x', \boldsymbol{v}')) \mapsto \boldsymbol{v}^T \mathbf{K}_{xx'} \boldsymbol{v}' \tag{3.52}$$

which describes the action of the kernel matrix on vectors. To continue, we need an appropriate notion of bilinearity for the given setting. We formulate this notion in general vector bundles, though we are interested in the tangent bundle TX whose fibers are the vector spaces T_xX.

Fiberwise bilinear function

Definition 3.18. *Let B be a vector bundle over X with fibers V_x, $x \in X$. We say that a function $k : B \times B \to \mathbb{R}$ FIBERWISE BILINEAR if for all pairs of points $x, x' \in X$ we have that*

$$k(\kappa\alpha_x + \lambda\beta_x, \gamma_{x'}) = \kappa k(\alpha_x, \gamma_{x'}) + \lambda k(\beta_x, \gamma_{x'}) \tag{3.53}$$

$$k(\alpha_x, \mu\gamma_{x'} + \nu\delta_{x'}) = \mu k(\alpha_x, \gamma_{x'}) + \nu k(\alpha_x, \delta_{x'}) \tag{3.54}$$

for any $\alpha_x, \beta_x \in V_x$, $\gamma_{x'}, \delta_{x'} \in V_{x'}$ and $\kappa, \lambda, \mu, \nu \in \mathbb{R}$.

This leads to the following notion of a kernel.

Positive semi-definite kernel

Definition 3.19. *A symmetric fiberwise bilinear function $k : T^*X \times T^*X \to \mathbb{R}$ is called a POSITIVE SEMI-DEFINITE KERNEL if for any set of covectors $\alpha_{x_1}, \ldots, \alpha_{x_n} \in T^*X$, we have*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} k(\alpha_{x_i}, \alpha_{x_j}) \geq 0. \tag{3.55}$$

At first, this may seem surprising: why should the kernel be a function defined on the *cotangent* bundle, rather than the tangent bundle? A clue is given by the form of the multivariate Gaussian density, which contains a $\boldsymbol{x}^T\mathbf{K}_{\boldsymbol{xx}}^{-1}\boldsymbol{x}$ term—note the presence of the inverse. This suggests that since the inverse kernel matrix acts on vectors, the kernel matrix should act on covectors. This mirrors the duality-based view described in Chapter 1, and is clarified further by considering the kernel of a given Gaussian vector field.

Cross-covariance kernel

Definition 3.20. *The CROSS-COVARIANCE KERNEL of a Gaussian vector field is defined as the map*

$$\alpha_x, \beta_{x'} \mapsto \operatorname{Cov}(\langle \alpha_x \mid f(\cdot, x)\rangle, \langle \beta_{x'} \mid f(\cdot, x')\rangle). \tag{3.56}$$

We now verify that this is indeed the correct notion of a kernel for the given setting. To do this, we need a general form of the Kolmogorov Extension Theorem, given by the following result.

Kolmogorov Extension Theorem

Result 3.21. *Let $(X_\alpha, \mathcal{B}_\alpha, \mathcal{O}_\alpha)_{\alpha\in A}$ be a family of measurable spaces, each equipped with a topology. For each finite $B \subseteq A$, let μ_B be an inner regular probability measure on $X_B = \bigtimes_{\alpha\in B} X_\alpha$ equipped with the product σ-algebra $\mathcal{B}_B$ and product topology $\mathcal{O}_B$, obeying*

$$(\operatorname{proj}_C)_* \mu_B = \mu_C \tag{3.57}$$

whenever $C \subseteq B \subseteq A$ are two nested finite subsets of A, where projections $\operatorname{proj}_C : X_B \to X_C$ are defined by $\operatorname{proj}_C(\{x_\alpha\}_{\alpha\in B}) = \{x_\alpha\}_{\alpha\in C}$, and $(\operatorname{proj}_C)_*$ denotes the measure-theoretic pushforward by proj_C. Then there exists a unique probability measure μ_A on $\mathcal{B}_A$ with the property that $(\operatorname{proj}_B)_*\mu_A = \mu_B$ for all finite $B \subseteq A$.

Proof. Tao [112], Theorem 2.4.3. ■

We are now ready to prove our bijective correspondence.

Theorem 3.22. *The distribution of every Gaussian vector field is uniquely determined by its mean vector field and cross-covariance kernel. Moreover, each such pair defines a Gaussian vector field.*

Proof. Let $x_1, .., x_n \in X$ and for $\alpha = (\alpha_{x_1}, .., \alpha_{x_n})$ and $\beta = (\beta_{x_1}, .., \beta_{x_n})$ define

$$\mu_{x_1,..,x_n} = (\mu(x_1), .., \mu(x_n)) \qquad k_{x_1,..,x_n}(\alpha, \beta) = \sum_{i=1}^{n}\sum_{j=1}^{n} k(\alpha_{x_i}, \beta_{x_j}). \tag{3.58}$$

Let $\pi_{x_1,..,x_n} \sim \mathrm{N}(\mu_{x_1,..,x_n}, k_{x_1,..,x_n})$ be defined in the sense of duality. These are our finite-dimensional marginals: by Gaussianity, it is clear these are uniquely determined by the mean and kernel. Taking X to be the index set, and $(T_x X)_{x\in X}$ with the standard topology and Borel σ-algebra to be our measurable spaces, we claim that the family of measures $(\pi_{x_1,..,x_n})_{\{x_1,..,x_n\}\subseteq X}$ satisfies the requirements of Kolmogorov's Extension Theorem. First, for any $\{x_1, .., x_m\} \subseteq \{x_1, .., x_n\}$, by direct calculation we have

$$(\operatorname{proj}_{x_1,..,x_m})_* \pi_{x_1,..,x_n} = \pi_{x_1,..,x_m} \tag{3.59}$$

using the canonical projection induced by the direct sum. Next, note that each $\pi_{x_1,..,x_n}$ is a probability measure on a finite-dimensional real space, it is automatically inner regular. Thus, it follows that there exists a unique measure on the infinite product space $\bigtimes_{x\in X} T_x X$. This is a topological

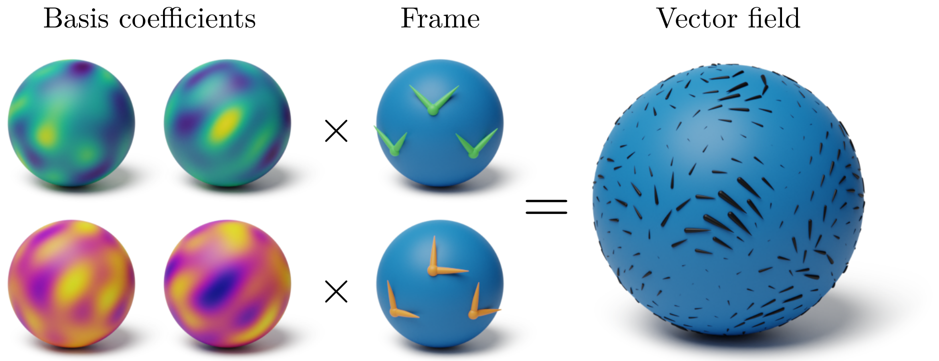


Figure 3.12. A vector field can be represented by specifying its values in any frame. Here, we illustrate the same vector field on the sphere represented using two different frames.

measure space: if we equip it with the obvious linear structure, and define the linear operator

$$\mathcal{I} : \bigtimes_{x \in X} T_x X \to \Gamma_{\text{nns}}(TX) \qquad (\mathcal{I}s)(x) = (x, \operatorname{proj}_x s) \tag{3.60}$$

where $\Gamma_{\text{nns}}(TX)$ is the space of not necessarily smooth sections, equipped with the pushforward σ-algebra, we obtain by pushforward the probability distribution of our desired process. The claim follows. ■

3.3.2. Coordinate representations. Using the preceding recipe, we now understand what needs to be defined in order to construct vector-valued Gaussian processes on manifolds. To make this construction concrete, we proceed to develop the calculations needed to implement such processes numerically. The first step is to understand how to represent vector fields in coordinates—we do so using bases in each tangent space.

Frame

Definition 3.23. *Define a* FRAME *F to be a collection of not necessarily smooth sections $e_i : X \to TX$ for $i = 1, .., d$ such that, for every $x \in X$, the vectors $e_i(x)$ form a basis of $T_x X$. Given a frame, define the* COFRAME *F^* to be the collection of sections $e^i : X \to T^*X$ defined pointwise using the dual basis with respect to F.*

The key words in the above definition are *not necessarily smooth*: recall again that for many manifolds, choosing a frame which varies smoothly in space is impossible—a single component of such a frame would yield a non-vanishing vector field, and such a vector field might not exist. An example of a vector field represented in two different frames can be seen in Figure 3.12. We can use this to calculate the coordinate expression of a cross-covariance kernel with respect to a frame. This is given by a function $\mathbf{K}_F : X \times X \to \mathbb{R}^{d\times d}$ as

$$\mathbf{K}_F(x,x') = \begin{bmatrix} k(e^1(x), e^1(x')) & \dots & k(e^1(x), e^d(x')) \\ \vdots & \ddots & \vdots \\ k(e^d(x), e^1(x')) & \dots & k(e^d(x), e^d(x')) \end{bmatrix} \tag{3.61}$$

which is a matrix-valued kernel in the usual sense. This expression is frame-dependent: to see how it transforms upon a change of frame, introduce another frame $\widetilde{F}$, and consider a function

$$f(x) = \sum_{i=1}^{d} f^i(x) e_i(x) = \sum_{i=1}^{d} \tilde{f}^i(x) \tilde{e}_i(x) \tag{3.62}$$

expressed with respect to F and $\widetilde{F}$, where $\tilde{f}^i$ and g^i are scalar-valued functions. We say that $\widetilde{F}$ is obtained from F by a *change of frame* if there is a matrix-valued function $\mathbf{A} : X \to \mathbb{R}^{d\times d}$ such that

Change of frame

$$\tilde{\boldsymbol{f}}(x) = \mathbf{A}(x)\boldsymbol{f}(x) \tag{3.63}$$

for all x. Since e_i and $\tilde{e}_i$ are bases, such a function always exists. The range of $\mathbf{A}$ is a subgroup of the general linear group $GL(d,\mathbb{R})$, and in general depends on what frame-dependent properties one wants to preserve. For instance, if one wants to preserve orthonormality of bases in each tangent space, one would consider a subgroup of orthonormality-preserving transformations, namely $SO(d)$. By direct calculation, $\mathbf{K}_F$ can be re-expressed as

$$\mathbf{K}_{\widetilde{F}}(x,x') = \mathbf{A}(x)\mathbf{K}_F(x,x')\mathbf{A}(x)^T \tag{3.64}$$

with respect to $\widetilde{F}$. This condition is called *equivariance under change of frame.* At this point, it is clear why we did not even attempt to generalize matrix-valued kernels directly by positing candidate formulas: for an arbitrary manifold, there is very little hope of simply guessing an expression that is simultaneously positive semi-definite and equivariant under change of frame.

Equivariance under change of frame

3.3.3. Projected kernels. Now that we understand how to write kernels of Gaussian vector fields in coordinates, we are ready to consider techniques for constructing such kernels and calculating them numerically. With our preparation complete, the construction we present is extremely simple, consisting of two steps.

1. Embed a set of p scalar-valued Riemannian Gaussian processes from X into $i(X) \subseteq \mathbb{R}^p$, and use them to assemble a vector-valued Gaussian process defined on on $i(X)$.

2. Project the tangent vectors onto each tangent space, obtaining a tangential vector field in the embedded space, and therefore a vector field on X.

We now proceed to make this precise. Suppose that we have an embedding

$$i : X \to \mathbb{R}^p \qquad\qquad i_{x,*} : T_x X \to \mathbb{R}^p \tag{3.65}$$

where the pushforward is defined at all points $x \in X$. Introducing a frame F allows us to define the projection map, which is a matrix-valued function $\mathbf{P}_F : X \to \mathbb{R}^{p \times d}$ defined by

$$\mathbf{P}_F(x) = \begin{bmatrix} i_{x,*}(e_i(x)) \\ \vdots \\ i_{x,*}(e_d(x)) \end{bmatrix} \tag{3.66}$$

which describes how to project arbitrary vectors onto tangent planes. Note that if X is Riemannian with metric g, then under the tangent-cotangent isomorphism, the linear map given by $\mathbf{P}_F$ is a right-inverse to the map given by $\mathbf{P}_F^T$, which describes how a vector field with respect to a frame can be re-expressed with respect to the coordinates in the embedded space. In particular, $\mathbf{P}_F^T \mathbf{P}_F = \mathbf{\Gamma}$, where $\Gamma_{ij} = g(e_i(x), e_j(x))$ is the coordinate representation of the metric. It is now clear how to make our idea precise.

Projected kernel

Definition 3.24. *Let $\boldsymbol{f} : \Omega \times X \to \mathbb{R}^{p \times p}$ be a Gaussian process defined on $i(X) \subseteq \mathbb{R}^p$, and define the Gaussian vector field*

$$f(\omega, x) = \mathbf{P}_F(x) \boldsymbol{f}(\omega, i(x)). \tag{3.67}$$

We call the cross-covariance kernel of a process constructed this way a PROJECTED KERNEL.

It follows that if $\boldsymbol{\kappa} : X \times X \to \mathbb{R}^{p \times p}$ is the matrix-valued cross-covariance kernel of $\boldsymbol{f}$, then the cross-covariance kernel of our Gaussian vector field is given by

$$\mathbf{K}_F(x, x') = \mathbf{P}_F(x) \boldsymbol{\kappa}(x, x') \mathbf{P}_F(x)^T. \tag{3.68}$$

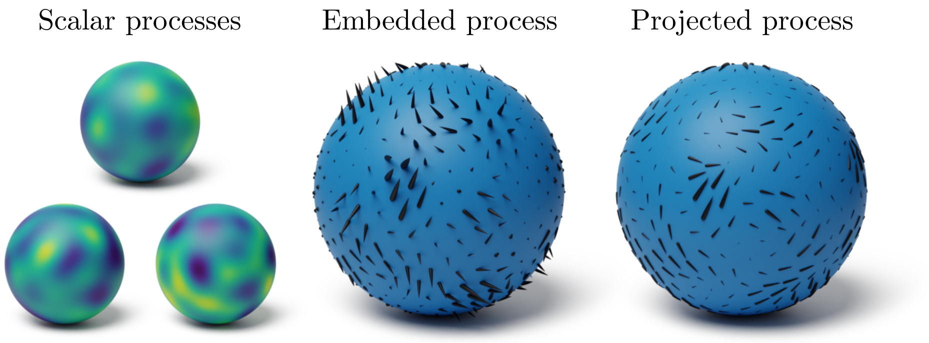


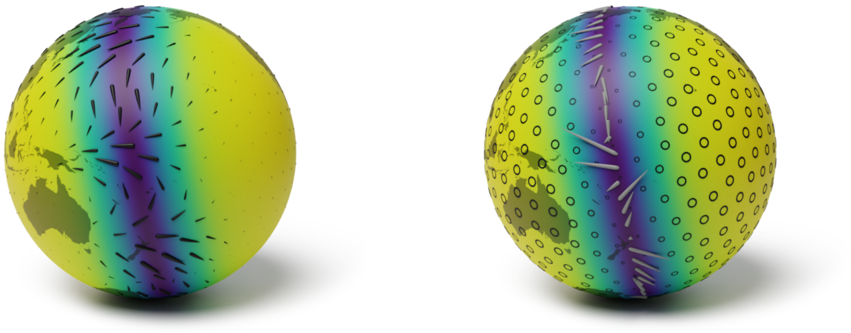
Figure 3.13. Here, we show how to construct Gaussian process on the sphere whose cross-covariance kernel is a projected kernel. First, we construct three scalar Gaussian processes on the sphere. Then, embed the sphere in $\mathbb{R}^3$, and the scalar Gaussian processes to form a vector-valued Gaussian process on the embedded sphere. Finally, we project the resulting Gaussian process onto the sphere, yielding the desired tangential vector field.

Note that here, we generally require a Riemannian structure on X in order to define a useful set of scalar-valued processes in order to obtain $\boldsymbol{f}$ or, equivalently, $\boldsymbol{\kappa}$, to begin with. The simplest approach is to make each component in the embedded space independent. Particularly in cases where the embedding is isometric, this gives a wide class of easy-to-understand kernels. An example can be seen in Figure 3.13.

It's easy to see that all cross-covariance kernels arise this way: using Nash's Theorem, we can embed an arbitrary Gaussian vector field on a Riemannian manifold into a Euclidean space, glue together the resulting scalar components, and project back to obtain the Gaussian vector field we started with.

This completes our technical development. Starting from first principles, we defined the notion of a Gaussian vector field, described in what sense such processes possess mean vectors and cross-covariance kernels, and defined a wide-ranging class of kernels which is completely constructive and can be implemented in software. In particular, our constructions are automatically compatible with standard training methods such as variational inference.

3.3.4. Illustrated examples. We now explore training Gaussian vector fields on Riemannian manifolds on observed data. Our goal is to demonstrate that the developed techniques are, in total, sufficiently explicit so as to enable Gaussian vector fields to be trained using standard methods without the


(a) Posterior mean

(b) Data and posterior covariance

Figure 3.14. Posterior mean and posterior cross-covariance, with the trace of the latter denoted by color, for a projected Gaussian vector field constructed using three independent Matérn-3/2 Gaussian processes on the sphere. Training data is shown in gray. The Gaussian process is seen to oversmooth data in locations with large jumps, such as near the easterly trade wind deviations in the center. We also observe that uncertainty increases as we move away from the training locations.

need for any additional machinery.

We work with the sphere. For training data, we consider global wind velocity, chosen because it is freely and readily available. Specifically, we consider interpolation of wind velocity deviations from historical average, at a fixed height, using wind velocity data directly observed by a satellite. For the prior, we use a Gaussian vector field model constructed using a projected kernel whose components are independent Matérn-3/2 scalar-valued Gaussian processes on the sphere.

We train the Gaussian process using exactly the same procedure as that used in the preceding sections: by optimizing kernel hyperparameters using gradient descent, and computing the value of the vector field at all locations using pathwise sampling. All computations are performed with respect to a fixed frame defined by the latitude-longitude coordinate system.

Results are shown in Figure 3.14. Immediately, we see that this dataset exhibits behavior not suitable for the chosen model. Specifically: in some regions, the data exhibits large jumps, while in other regions, it is very smooth over large distances. This means the effective length scale governing variability is spatially non-uniform, which requires a more sophisticated model to effectively represent, such as one with a spatially varying length scale.

The chosen Gaussian process model responds to this by overinterpolating the data and reverting to the overall mean in regions with rapidly changing wind velocity. While this is not ideal from a modeling perspective, it is at least a relatively graceful failure mode, compared to numerical instability or incorrect results in other regions where rapid shifts do not occur. Effects such as these are closely linked to non-linear behavior in the differential equations governing weather phenomena, and generally require special consideration.

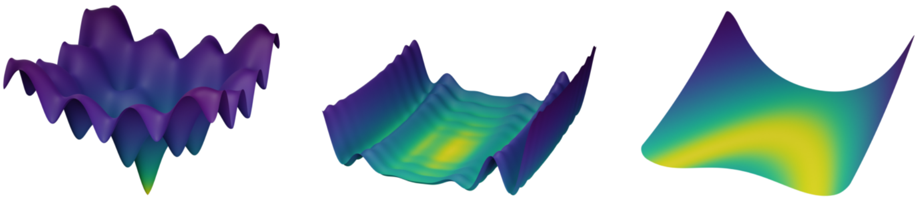
On the other hand, we observe that the trained Gaussian process model is smooth, even though the choice of frame is non-smooth at the top of the sphere—a key motivation behind developing the theory for working with vector fields in a geometrically-sound way in the first place. Similarly, we see that posterior uncertainty increases as we move away from locations where training data is available. This shows that the model correctly incorporates geometry and behaves in a manner mirroring its scalar-valued analogs.

One notable characteristic of the obtained posterior vector field is that the covariance of the output vectors at any given point is close to isotropic. This can be seen from the ellipsoids in Figure 3.14, which appear circular, though they are clearly elliptical from numerical examination. This behavior appears to be a feature of the data locations chosen in this particular example, which lie approximately on a great circle, and does not necessarily occur in general.

The model used is also not realistic in other ways, beyond its use of a single global length scale: in particular, it does not incorporate time or any kind of physical information into its design. Nonetheless, we believe that it serves as a useful demonstration that constructing vector-valued Gaussian processes using the ideas developed is practical. We hope this motivates further development to apply such models and extensions thereof more broadly, both for the sphere and for other manifolds.

3.4. Geometry-aware Bayesian optimization

We now study the role that geometry plays when working with decision systems built using Gaussian processes. To do so, we perform Bayesian optimization of standard benchmark functions on Riemannian manifolds using the Riemannian Matérn and squared exponential Gaussian processes with the numerical techniques developed herein. Following Jaquier et al. [55], we call this setting *geometry-aware Bayesian optimization.*



(a) Ackley function (b) Levy function (c) Rosenbrock function

Figure 3.15. Global optimization benchmark functions used in the geometry-aware Bayesian optimization experiments. All three are defined and shown on an ambient Euclidean space of dimension two.

We consider three manifolds: the spheres $\mathbb{S}^3$ and $\mathbb{S}^5$, and the special orthogonal group $SO(3)$, each embedded within a Euclidean space in the natural manner. We use three target functions commonly used as benchmarks in global optimization: the *Ackley* [1], *Levy* [69], and *Rosenbrock* [91] functions.[2] These are defined in the ambient spaces: we optimize their restrictions onto the manifold of interest, centered to ensure the minima is on the manifold. Figure 3.15 plots these benchmark functions for a Euclidean space.

For each manifold, we use two priors: Matérn with smoothness $\nu = 5/2$ and squared exponential, both with a Gaussian likelihood. The remaining kernel hyperparameters and likelihood error variance are learned from observed data via gradient descent. We use expected improvement for our acquisition function, computed directly from the kernel. We run for a total of $n = 200$ iterations, and repeat each experiment 30 times to assess variability.

To establish a performance baseline, we compare against Bayesian optimization in ambient Euclidean space. Here, the acquisition point is selected by performing constrained optimization on the acquisition function, ensuring that points selected for evaluation are located on that manifold.

Results can be seen in Figure 3.16. We plot median regret curves, along with first and third quartiles. From this, we see that the performance of geometry-aware Bayesian optimization depends on interplay between the manifold and the overall shape of the function being optimized.

[2] The Ackley, Levy, and Rosenbrock benchmark functions, respectively, are given as
$f_{\mathrm{A}}(x_1, .., x_d) = 20 - 20\exp(-0.2(\frac{1}{d}\sum_{i=1}^{d} x_i^2)^{1/2}) - \exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)) + e$
$f_{\mathrm{L}}(x_1, .., x_d) = \sin(\pi w_i)^2 + \sum_{i=1}^{d-1}(w_i - 1)^2(1 + 10\sin(\pi w_i + 1)^2) + (w_d - 1)^2(1 + \sin(2\pi w_d)^2)$
$f_{\mathrm{R}}(x_1, .., x_d) = \sum_{i=1}^{d-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$
where $w_i = 1 + \frac{x_i - 1}{4}$ and d is the dimension of the space on which the functions are defined.

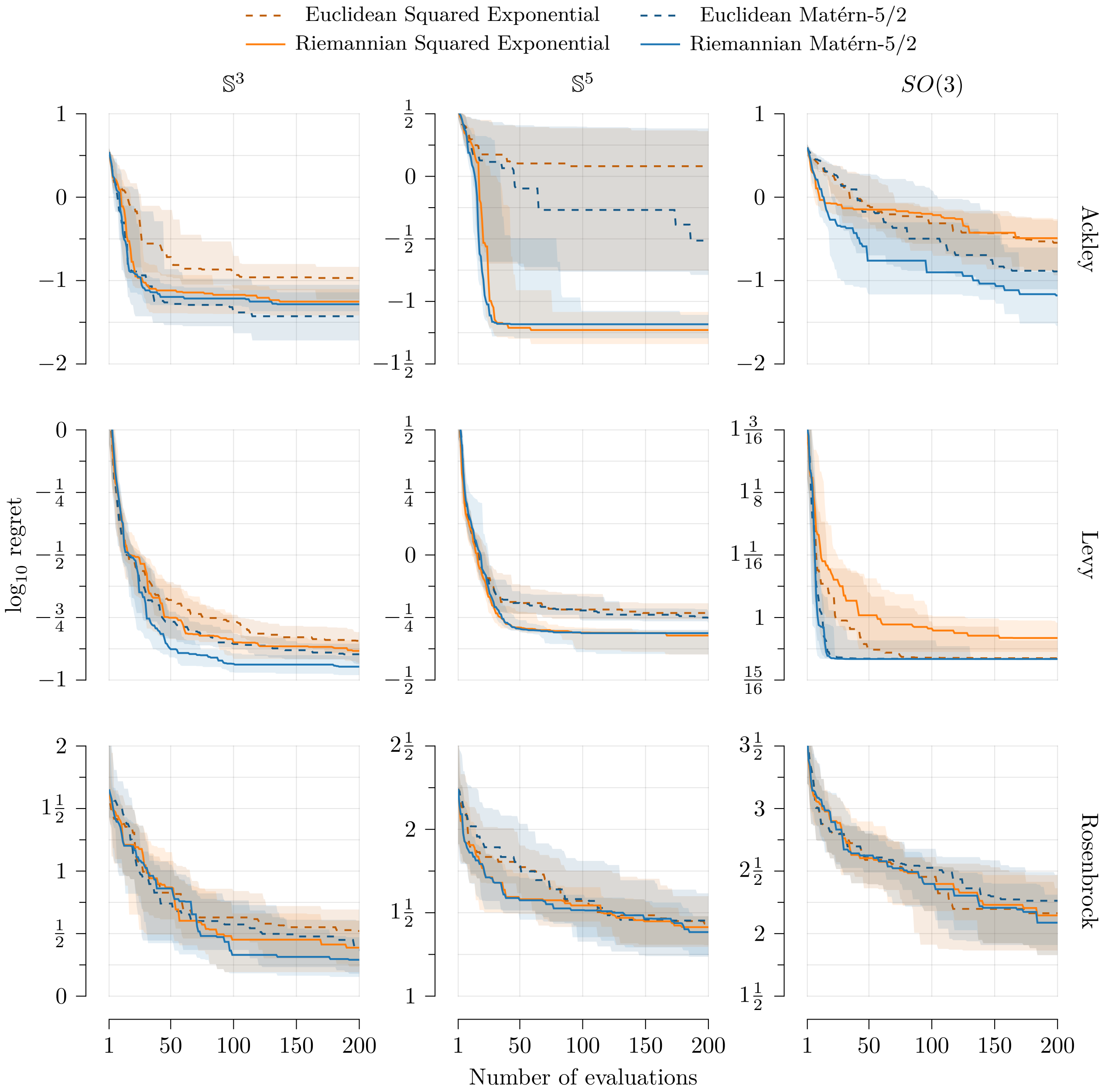


Figure 3.16. Geometry-aware Bayesian optimization benchmark results.

For the 3-dimensional sphere, all techniques produce comparable regret curves. This shows working with geometry-aware models and acquisition optimization, compared to constrained acquisition optimization on Euclidean models defined in ambient spaces, offers less benefit in cases where the ambient space is not too different from the manifold. The performance of Matérn and squared exponential models, in this case, is similar.

For the 5-dimensional sphere, the role of geometry becomes more pronounced. For the Ackley and Levy functions, we see a clear improvement from using geometry-aware methods. For the Rosenbrock function, which possesses less local variability, performance of all methods remains similar. For all three target functions, Matérn and squared exponential models perform similarly.

On the 3-dimensional special orthogonal group, geometry-aware models perform similarly to models defined on the ambient Euclidean spaces for all three benchmark functions. On the other hand, here, Matérn models outperform squared exponential models. Therefore, we see that the effect of kernel smoothness on performance of Bayesian optimization may depend on the geometry of the manifold, as well as the target function.

Overall, the results show that Riemannian models perform comparably to Euclidean ones in some cases, and outperform them in others. The effect varies according to the interplay between the target function being optimized and the manifold's geometry. We also see that slightly less smooth Matérn Gaussian processes perform comparably to squared exponential Gaussian processes in some cases, and outperform them in others.

Bayesian optimization is often used in settings where function evaluations are extremely expensive, and therefore even the relatively modest performance gains seen here can be consequential. In such cases, taking advantage of constraints, symmetries, and other geometric properties is a promising route towards improved performance. Geometry-aware Bayesian optimization offers a principled framework for doing so.

3.5. Conclusion

In the preceding section, we have developed three key classes of *non-Euclidean Gaussian process* models: the *Riemannian Matérn* and *graph Matérn* classes of scalar-valued Gaussian processes, and the *projected kernel* class of vector-valued Gaussian processes on Riemannian manifolds. Each of these provides a unifying view of previously-proposed methods, and expands Gaussian processes models' scope of applicability.

The key idea for defining Matérn Gaussian processes on Riemannian manifolds was to not attempt to define kernels using geodesic distances—such constructions, while intuitively appealing, turn out to not lead to an effective mathematical formalism. Instead, we relied on the idea of *functional calculus* built using spectral theory of the Laplace–Beltrami operator to introduce operators through which we constructed the desired Gaussian processes as transformations of Gaussian white noise processes.

Adopting this viewpoint enabled us to reinterpret models previously-proposed within a cumbersome finite element framework using a more abstract approach built via the theory of stochastic partial differential equations and reproducing kernel Hilbert spaces. From this approach, we obtained formulas for pointwise evaluation of the *Riemannian Matérn kernel* in terms of eigenvalues and eigenfunctions of the Laplace–Beltrami operator. The formalism also enabled us to define and compute the *Riemannian squared exponential kernel.*

While our original original formalism was introduced for working with manifolds, it is clear that the constructions are more general. Inspired by the discrete computations used in implementing the Riemannian Gaussian processes we defined, we explored purely-discrete analogs of these models, obtaining the *graph Matérn kernel* and *graph squared exponential kernel.* These enabled us to apply Matérn Gaussian processes to model phenomena in purely discrete spaces whose geometry departs significantly from manifold-like settings.

Finally, we returned to the manifold setting, this time studying formalism for defining *Gaussian vector fields.* This setting is non-trivial from the point of conceptualization: a vector-valued Gaussian is only *locally* a vector-valued random function, and instead should be formulated using the differential-geometric formalism of *random sections.* We therefore began by clarifying what the appropriate notion of a *Gaussian vector field* and *cross-covariance kernel* should be, and how to work with this notion numerically.

Having done so, we introduced a wide class of cross-covariance kernels for defining such processes called *projected kernels.* These kernels can be built from the Riemannian Gaussian process models already defined, together with an embedding of the manifold into an ambient Euclidean space, and produce covariances that reflect the manifold's geometry. This construction enabled us to implement Bayesian learning of vector-valued data on the sphere.

Having explored these constructions from a purely model-building point of view, we concluded by considering them within a decision-making setting, by studying *geometry-aware Bayesian optimization*. There, saw that using geometry-aware models also resulted in a modest performance improvement in cases where topology and geometry play a non-trivial role, particularly for higher-dimensional spaces.

It is clear that our ideas apply within a wider scope than what we have explored. For example, our derivations for the Matérn class apply essentially-unmodified to closely-related settings such as *manifolds with boundary*, provided one introduces boundary conditions which ensure the operators of interest behave the same way as in our setting. Extensions such as this make excellent candidates for future work.

Our differential-geometric framework also provides a promising way to study *symmetry* within Gaussian processes and Bayesian optimization. Since symmetry plays a fundamental role in geometry and in physics, understanding how to handle symmetries is a promising avenue for building richer classes of Gaussian process models, and applying techniques such as Bayesian optimization to wider classes of scientific problems.

We hope that techniques such as the ones presented here provide a foundation upon which these and other developments can be built. We therefore conclude our presentation of contributions, and move to discuss the state and overall picture of Gaussian processes and decision systems that we have studied.

Chapter 4

Discussion

Bayesian learning with Gaussian process models offers a clear and comprehensive framework for making decisions that assess and propagate uncertainty in order to resolve explore-exploit tradeoffs inherent in decision-making settings. In this dissertation, we explored two avenues for broadening applicability of Gaussian process models in such settings.

In Chapter 2, we presented pathwise conditioning methods. These methods allow a posterior Gaussian process, as an actual random function, to be expressed as a sum of a prior Gaussian process and a dependent update term. Pathwise conditioning gives rise to Gaussian process approximations for which all stochasticity can be evaluated in advance, which makes it substantially easier for upstream algorithms to interface with the Gaussian process.

In Chapter 3, we presented a collection of techniques for working with Gaussian process models defined on non-Euclidean spaces, including Riemannian manifolds and graphs. We derived constructive and practical expressions which enable such models to be trained using standard methods. For the Riemannian setting, we did so for both scalar-valued and vector-valued processes.

For both pathwise conditioning and non-Euclidean Gaussian process models, the end result of our constructions was the ability to perform Bayesian optimization in a flexible and effective manner tailored to the setting at hand. This allowed us to benchmark our ideas in one of the simplest, but most important, classes of decision systems.

4.1. Future work

One end goal of Gaussian process research is the design of decision-making systems which simply work in the settings where they apply, without the usual tuning or numerical difficulties, and fail gracefully in settings where they do not apply. The ideas presented naturally lead to a number of further research directions towards this goal, which we briefly comment on now.

Decision systems beyond optimization. Perhaps the most promising research direction made possible by pathwise conditioning methods is the design of data-efficient decision systems for tasks beyond optimization. Neiswanger et al. [80] propose *Bayesian algorithm execution*, a framework which generalizes Bayesian optimization by allowing for the task to be defined by using a general *underlying algorithm* in place of optimization.

For this, Neiswanger et al. [80] introduce *InfoBAX*, an information-theoretic acquisition function constructed directly from the algorithm being considered. This enables the introduction of *Bayesian Dijkstra's algorithm* for finding shortest paths in a graph in a data-efficient manner, and other novel decision systems.

Pathwise conditioning methods play a key role in computing the InfoBAX acquisition function. Specifically, one runs the underlying algorithm on posterior random functions, producing a set of *execution paths* of points queried by the algorithm. InfoBAX is then constructed from the execution trace and output via a set of information-theoretic calculations. Approximate pathwise conditioning makes it possible to compute the required execution traces in an efficient, accurate, and convenient manner.

Gaussian processes have not yet been widely considered for tasks not conveniently expressed as maximization of rewards. Bayesian algorithm execution enables such methods to be considered for settings beyond Bayesian optimization and model-based reinforcement learning, which are well-established and whose limitations are largely known. Developing this framework to the same degree of detail afforded to those cases is therefore a particularly promising research direction.

From a theoretical point of view, one of the first steps to be taken in understanding Bayesian algorithm execution is to understand what is the appropriate notion of *regret* for this setting, among a number of possible variations, and to develop techniques for studying it. The analysis of closely-related state-of-the-art bandit algorithms, such as *information-directed sampling* [92], offers hints on how to proceed.

In parallel, it seems fruitful to look for applications where more general data-efficient decision systems beyond optimization are potentially useful, and yet where Gaussian process models are viable. *Multi-objective Bayesian optimization* [35, 18, 50, 111], where the goal is finding a Pareto frontier, rather than obtaining a global minima, gives a concrete algorithm class to consider studying from this point of view.

From the Bayesian algorithm execution framework alone, we see that pathwise conditioning methods enable potential new avenues for constructing decision systems and therefore growing the applicability and scope of Gaussian process methods within machine learning. We hope that, in the coming years, more of these avenues are explored, enabling Gaussian processes to have larger role in the machine learning toolbox.

Connections to numerical analysis. Another promising research direction is exploring the connections between Gaussian processes and numerical analysis. Interest in these connections has grown rapidly in the Gaussian process community within machine learning in recent years, as a way to improve performance and usability of software implementations [76, 40, 116]. Ideas based on pathwise conditioning and stochastic partial differential equations offers new avenues for taking advantage of these connections.

Firstly, one can study use of iterative solvers, such as the conjugate gradient method, in combination with pathwise conditioning and inducing points. Iterative solvers are promising because they can in principle tackle larger linear systems for which Cholesky factorization simply fails, provided they are set up correctly. A large set of techniques in this class have recently been proposed for Gaussian process models [31, 40, 85, 77, 86].

The computational costs of an iterative solver depend on how many iterations are required for convergence, which vary according to the condition number of the linear system being solved [94]. Variational approximations such as the inducing point construction presented in Chapter 2 make it possible to work with linear systems that are substantially better-conditioned than those in the true posterior, due to additional diagonal terms added to the matrix. Exploring such combinations is a promising route for improving performance.

Pathwise conditioning makes it possible to avoid computation of matrix square roots when working with iterative solvers. Matrix square roots generally require further considerations than simply solving linear systems [86] and can be less convenient. With pathwise conditioning, the only additional quantities needed, beyond solution of the linear system itself, are generally log-determinant terms. Techniques for computing these terms have been proposed [95, 31], and are key part in making iterative methods effective.

In the partial differential equation and mathematical physics communities, iterative solvers for linear systems constructed using finite element methods achieve state-of-the-art performance for many problems [37, 74]. Connections with stochastic partial differential equations, such as the ones explored here, offer a fruitful avenue for seeking inspiration for improving Gaussian process performance further.

A key lesson from numerical analysis is that, in order to solve a linear system, the most important step is often the construction of a *preconditioner* to partially solve the problem before an iterative method is used [94, 33, 124]. There, the degree of sophistication used in constructing preconditioners is often at least as large as that used in constructing iterative solvers themselves.

Multi-grid methods provide a broad framework for constructing preconditioners which achieve state-of-the-art performance for many partial differential equations [33, 124]. In essence, such methods provide a framework for resolving long-range effects at a coarse resolution, and short-range effects at a fine resolution, improving computational efficiency in the process. Adapting such ideas to the Gaussian process setting offers possibilities for improvement.

Numerical analysis can also be a source of inspiration for improving variational inference. Inducing points often struggle on datasets where the overall data size is large, but the amount of data points within any given length scale ball is small. Multi-scale methods suggest that right kind of approximation for this regime should involve replacing long-range effects with averaged, sparsified versions thereof [33, 124]. Understanding how construct such a variational approximation is therefore a promising avenue for future work.

In total, ideas from numerical analysis provide a fruitful source of ideas for improving Gaussian processes further. We hope that in the coming years, these ideas come to fruition, and using Gaussian processes in most practical settings become just as easy and seamless as training neural networks on well-understood problem classes has become.

Improved understanding of kernels on geometric spaces. The constructions presented in Chapter 3 for building non-Euclidean Gaussian process models give general formulas applicable for a wide variety of spaces. Many manifolds encountered in practice, particularly those used in physics and engineering, possess additional mathematical structure, which can be used to both construct new kernels and provide avenues for more efficiently computing the Matérn and squared exponential kernels studied previously.

For example, the eigenfunctions of the Laplace–Beltrami operator on the sphere are the spherical harmonics [21, 19]. The spherical harmonics, in turn, can be re-expressed in terms of the *Gegenbauer polynomials*, which are a set of polynomials on $[-1, 1]$ which are orthogonal with respect to a certain weight function [45]. The Riemannian Matérn and squared exponential kernels can therefore be written in terms of these polynomials, and turns out to only depend on the geodesic distance between two points.

This is a substantial simplification compared to the general case, and one can use it to pre-compute the kernel for various values of the geodesic distance, and interpolate the resulting values, rather than evaluate an infinite series every time the kernel needs to be computed. Note that the spherical squared exponential kernel, when expressed using this simplification, is not the squared exponential of the geodesic distance: instead, it is a different, more complicated function of the geodesic distance.

Similar simplifications also occur on the special orthogonal group, suggesting that there may be a general theory that describes them. Finding such a theory is a promising direction for future work. To do so, the first step would be to understand how symmetries of a manifold affect kernels defined over it via spectral methods similar to the ones we have used.

Simplifications may also occur due to other structures beyond symmetries. Lie groups are particularly important in many areas, including robotics, computer vision, physics, and many engineering disciplines. Due to their prominence, it therefore seems fruitful to study potential methods for computing kernels on Lie groups, provided they are equipped with a Riemannian metric compatible with the group structure.

In particular, studying kernels on non-compact Lie groups, and related spaces such as certain matrix manifolds, is a promising direction towards a general theory of Riemannian kernels which does not require compactness. Here, one would need more general forms of spectral theory than Sturm–Liouville decompositions. These are mathematically sophisticated, requiring additional machinery such as integration over projection-valued measures [63]. Exploring such generalizations is a promising avenue for further work.

Finally, one can consider extending our constructions to more general spaces than manifolds, such as for example manifolds with boundary, which possess similar spectral properties. Spaces like this are often important in applications areas [103, 60, 102, 25]. Studying heat kernels, which have been considered in settings as general as metric measure spaces [47, 46], gives hints about the total scope which may be possible.

Summarizing, the constructions we have studied only begin to describe the full scope of geometry-aware kernels such as the Matérn and squared exponential class constructed using Sturm–Liouville theory and described here. We hope this first step leads to further substantive understanding of this class of methods within machine learning.

4.2. Conclusion

The contributions presented in this thesis expand the set of settings where Gaussian processes can be used. In particular, pathwise conditioning methods reduce the barriers needed to deploy Gaussian process models once they have been trained, and make it easier to obtain posterior quantities of interest for the setting at hand. Simultaneously, the obtained kernel expressions for non-Euclidean Gaussian processes enable such models to be trained using off-the-shelf methods in a standard manner.

Overall, these ideas reduce the barriers that need to be overcome in order for practitioners to use Gaussian processes in day-to-day work. We hope the ideas presented enable more people to consider using Gaussian process models as part of their workflow, and bring decision-making systems such as Bayesian optimization to new areas of science, technology, and engineering.

References

[1] D. Ackley. *A Connectionist Machine for Genetic Hillclimbing.* Springer, 1987. Cited on page 140.

[2] R. Agrawal. Sample Mean Based Index Policies with $O(\log n)$ Regret for the Multi-Armed Bandit Problem. *Advances in Applied Probability*, 27(4):1054–1078, 1995. Cited on page 41.

[3] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows in Metric Spaces and in the Space of Probability Measures.* Springer, 2008. Cited on page 28.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time Analysis of the Multiarmed Bandit Problem. *Machine Learning*, 47(2):235–256, 2002. Cited on pages 41, 46.

[5] M. Belkin and P. Niyogi. Convergence of Laplacian Eigenmaps. In *Advances in Neural Information Processing Systems*, 2007. Cited on page 126.

[6] J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. Cited on page 97.

[7] D. P. Bertsekas. *Reinforcement Learning and Optimal Control.* Athena Scientific, 2019. Cited on page 34.

[8] P. Billingsley. *Probability and Measure.* Wiley, 2008. Cited on page 33.

[9] V. I. Bogachev. *Gaussian Measures.* American Mathematical Society, 1998. Cited on page 60.

[10] V. I. Bogachev. *Measure Theory*, volume 1. Springer, 2007. Cited on page 24.

[11] V. I. Bogachev. *Measure Theory*, volume 2. Springer, 2007. Cited on pages 24, 28.

[12] V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. P. Deisenroth, and N. Durrande. Matérn Gaussian Processes on Graphs. In *Artificial Intelligence and Statistics*, 2021. Cited on page 23.

[13] V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Matérn Gaussian Processes on Riemannian Manifolds. In *Advances in Neural Information Processing Systems*, 2020. Cited on page 23.

[14] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges. Working manuscript, Imperial College London, 2021. Cited on page 104.

[15] D. Burago, S. Ivanov, and Y. Kurylev. A Graph Discretization of the Laplace–Beltrami Operator. *Journal of Spectral Theory*, 4(4):675–714, 2014. Cited on page 126.

[16] D. R. Burt, C. E. Rasmussen, and M. van der Wilk. Rates of Convergence for Sparse Variational Gaussian Process Regression. In *International Conference on Machine Learning*, 2019. Cited on page 89.

[17] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. Cited on page 97.

[18] P. Campigotto, A. Passerini, and R. Battiti. Active Learning of Pareto Fronts. *IEEE Transactions on Neural Networks and Learning Systems*, 25(3):506–519, 2014. Cited on page 147.

[19] Y. Canzani. Analysis on Manifolds via the Laplacian. Lecture notes, Harvard University, 2013. Cited on pages 108, 149.

[20] J. T. Chang and D. Pollard. Conditioning as Disintegration. *Statistica Neerlandica*, 51(3):287–317, 1997. Cited on page 28.

[21] I. Chavel. *Eigenvalues in Riemannian Geometry.* Academic Press, 1984. Cited on pages 108, 149.

[22] J.-P. Chilès and C. Lantuéjoul. Prediction by Conditional Simulation: Models and Algorithms. In *Space, Structure and Randomness*, pages 39–68. Springer, 2005. Cited on page 72.

[23] K. Choromanski, M. Rowland, T. Sarlós, V. Sindhwani, R. Turner, and A. Weller. The Geometry of Random Features. In *International Conference on Artificial Intelligence and Statistics*, 2018. Cited on page 82.

[24] K. Choromanski and V. Sindhwani. Recycling Randomness with Structure for Sublinear time Kernel Expansions. In *International Conference on Machine Learning*, 2016. Cited on page 82.

[25] S. Coveney, C. Corrado, C. H. Roney, D. O'Hare, S. E. Williams, M. D. O'Neill, S. A. Niederer, R. H. Clayton, J. E. Oakley, and R. D. Wilkinson. Gaussian Process Manifold Interpolation for Probabilistic Atrial Activation Maps and Uncertain Conduction Velocity. *Philosophical Transactions of the Royal Society A*, 378(2173):20190345, 2020. Cited on pages 84, 150.

[26] K. Crane and M. Wardetzky. A Glimpse into Discrete Differential Geometry. *Notices of the American Mathematical Society*, 64(11):1153–1159, 2017. Cited on page 120.

[27] B. Curless and M. Levoy. A Volumetric Method for Building Complex Models from Range Images. In *Conference on Computer Graphics and Interactive Techniques*, pages 303–312, 1996. Cited on page 120.

[28] G. Dahlquist and Å. Björck. *Numerical Methods in Scientific Computing*. Society for Industrial and Applied Mathematics, 2008. Cited on page 92.

[29] E. De Vito, N. Mücke, and L. Rosasco. Reproducing Kernel Hilbert Spaces on Manifolds: Sobolev and Diffusion Spaces. *Analysis and Applications*, 19(3):363–396, 2020. Cited on pages 116, 117.

[30] C. de Fouquet. Reminders on the Conditioning Kriging. In *Geostatistical Simulations*, pages 131–145. Springer, 1994. Cited on page 73.

[31] K. Dong, D. Eriksson, H. Nickisch, D. Bindel, and A. G. Wilson. Scalable Log Determinants for Gaussian Process Kernel Learning. In *Advances in Neural Information Processing Systems*, 2017. Cited on pages 90, 147, 148.

[32] A. Doucet. A Note on Efficient Conditional Simulation of Gaussian Distributions. Technical report, University of British Columbia, 2010. Cited on page 72.

[33] W. E. *Principles of Multiscale Modeling*. Cambridge University Press, 2011. Cited on page 148.

[34] X. Emery. Conditioning Simulations of Gaussian Random Fields by Ordinary Kriging. *Mathematical Geology*, 39(6):607–623, 2007. Cited on page 73.

[35] M. Emmerich, N. Beume, and B. Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 62–76, 2005. Cited on page 147.

[36] J. F. Epperson. On the Runge Example. *The American Mathematical Monthly*, 94(4):329–341, 1987. Cited on page 92.

[37] L. C. Evans. *Partial Differential Equations.* American Mathematical Society, 2010. Cited on page 148.

[38] A. Feragen, F. Lauze, and S. Hauberg. Geodesic Exponential Kernels: When Curvature and Linearity Conflict. In *Conference on Computer Vision and Pattern Recognition*, 2015. Cited on page 110.

[39] P. I. Frazier. Bayesian Optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*, pages 255–278. Institute for Operations Research and the Management Sciences, 2018. Cited on page 45.

[40] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*, 2018. Cited on pages 90, 147.

[41] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis.* CRC Press, 3rd edition, 2014. Cited on page 23.

[42] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach.* Springer, 1991. Cited on page 83.

[43] S. Ghosal and A. van der Vaart. *Fundamentals of Nonparametric Bayesian Inference.* Cambridge University Press, 2017. Cited on page 23.

[44] E. Giné and R. Nickl. *Mathematical Foundations of Infinite-Dimensional Statistical Models.* Cambridge University Press, 2015. Cited on page 23.

[45] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products.* A. Jeffrey and D. Zwillinger, editors. Academic Press, 7th edition, 2014. Cited on pages 114, 149.

[46] A. Grigoryan, J. Hu, and K.-S. Lau. Heat Kernels on Metric Measure Spaces. In *Geometry and Analysis of Fractals*, pages 147–207. Springer, 2014. Cited on page 150.

[47] A. Grigoryan, J. Hu, and K.-S. Lau. Heat Kernels on Metric Measure Spaces and an Application to Semilinear Elliptic Equations. *Transactions of the American Mathematical Society*, 355(5):2065–2095, 2003. Cited on page 150.

[48] M. Hairer. An Introduction to Stochastic PDEs. Lecture notes, University of Warwick, 2009. Cited on page 60.

[49] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian Processes for Big Data. In *Uncertainty in Artificial Intelligence*, 2013. Cited on page 87.

[50] D. Hernández-Lobato, J. M. Hernández-Lobato, A. Shah, and R. P. Adams. Predictive Entropy Search for Multi-objective Bayesian Optimization. In *International Conference on Machine Learning*, 2016. Cited on page 147.

[51] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani. Predictive Entropy Search for Efficient Global Optimization of Black-box Functions. In *Advances in Neural Information Processing Systems*, 2014. Cited on pages 47, 90.

[52] Y. Hoffman and E. Ribak. Constrained Realizations of Gaussian Fields: A Simple Algorithm. *The Astrophysical Journal*, 380:L5–L8, 1991. Cited on page 73.

[53] M. J. Hutchinson, A. Terenin, V. Borovitskiy, S. Takao, Y. W. Teh, and M. P. Deisenroth. Vector-valued Gaussian Processes on Riemannian Manifolds via Gauge Independent Projected Kernels. In *Advances in Neural Information Processing Systems*, 2021. Cited on page 23.

[54] N. Jaquier, V. Borovitskiy, A. Smolensky, A. Terenin, T. Asfour, and L. Rozo. Geometry-aware Bayesian Optimization in Robotics using Riemannian Matérn Kernels. In *Conference on Robot Learning*, 2021. Cited on page 23.

[55] N. Jaquier, L. Rozo, S. Calinon, and M. Bürger. Bayesian Optimization Meets Riemannian Manifolds in Robot Learning. In *Conference on Robot Learning*, 2020. Cited on page 139.

[56] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian Optimization Without the Lipschitz Constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993. Cited on page 97.

[57] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998. Cited on page 47.

[58] A. G. Journel and C. J. Huijbregts. *Mining Geostatistics.* Academic Press London, 1978. Cited on page 73.

[59] O. Kallenberg. *Foundations of Modern Probability.* Springer, 2006. Cited on pages 24, 28, 33.

[60] E. Krainski, V. Gómez-Rubio, H. Bakka, A. Lenzi, D. Castro-Camilo, D. Simpson, F. Lindgren, and H. Rue. *Advanced Spatial Modeling with Stochastic Partial Differential Equations using R and INLA*. CRC Press, 2018. Cited on pages 84, 150.

[61] H. J. Kushner. A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise. *Journal of Basic Engineering*, 86(1):97–106, 1964. Cited on page 47.

[62] T. L. Lai and H. Robbins. Asymptotically Efficient Adaptive Allocation Rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. Cited on pages 40, 41.

[63] S. Lang. *Real and Functional Analysis*. Springer, 2012. Cited on pages 49, 149.

[64] T. Lattimore and C. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. Cited on pages 38, 45, 47.

[65] J.-F. Le Gall. *Brownian Motion, Martingales, and Stochastic Calculus*. Springer, 2016. Cited on page 80.

[66] J. M. Lee. *Introduction to Riemannian Manifolds*. Springer, 2018. Cited on page 104.

[67] J. M. Lee. *Introduction to Smooth Manifolds*. Springer, 2012. Cited on page 104.

[68] J. M. Lee. *Introduction to Topological Manifolds*. Springer, 2010. Cited on pages 104, 129.

[69] A. V. Levy, A. Montalvo, S. Gomez, and A. Calderon. Topics in Global Optimization. In *Numerical Analysis*, pages 18–33. Springer, 1982. Cited on page 140.

[70] Z. Li, J.-F. Ton, D. Oglic, and D. Sejdinovic. Towards A Unified Analysis of Random Fourier Features. In *International Conference on Machine Learning*, 2019. Cited on page 82.

[71] M. Lifshits. *Lectures on Gaussian Processes*. Springer, 2012. Cited on pages 49, 60, 80.

[72] F. Lindgren, H. Rue, and J. Lindström. An Explicit Link Between Gaussian Fields and Gaussian Markov Random Fields: the Stochastic Partial Differential Equation Approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011. Cited on pages 84, 86, 113, 114, 116, 127.

[73] F. Liu, X. Huang, Y. Chen, and J. A. K. Suykens. Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. Cited on page 82.

[74] G. J. Lord, C. E. Powell, and T. Shardlow. *An Introduction to Computational Stochastic PDEs*. Cambridge University Press, 2014. Cited on pages 83, 84, 148.

[75] S. V. Lototsky and B. L. Rozovsky. *Stochastic Partial Differential Equations*. Springer, 2017. Cited on pages 84, 113, 115.

[76] A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian Process Library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017. Cited on page 147.

[77] G. Meanti, L. Carratino, L. Rosasco, and A. Rudi. Kernel Methods Through the Roof: Handling Billions of Points Efficiently. In *Advances in Neural Information Processing Systems*, 2020. Cited on pages 90, 147.

[78] J. Močkus. On Bayesian Methods for Seeking the Extremum. In *Optimization Techniques International Federation for Information Processing Technical Conference*, pages 400–404, 1975. Cited on page 47.

[79] M. Mutny and A. Krause. Efficient High Dimensional Bayesian Optimization with Additivity and Quadrature Fourier Features. In *Advances in Neural Information Processing Systems*, 2018. Cited on pages 90, 91.

[80] W. Neiswanger, K. A. Wang, and S. Ermon. Bayesian Algorithm Execution: Estimating Computable Properties of Black-box Functions Using Mutual Information. In *International Conference on Machine Learning*, 2021. Cited on page 146.

[81] D. S. Oliver. On Conditional Simulation to Inaccurate Data. *Mathematical Geology*, 28(6):811–817, 1996. Cited on page 73.

[82] OpenStreetMap contributors. Retrieved from HTTPS://PLANET.OSM.ORG, 2017. Cited on page 127.

[83] M. Opper and C. Archambeau. The Variational Gaussian Approximation Revisited. *Neural Computation*, 21(3):786–792, 2009. Cited on pages 87, 89.

[84] V. I. Paulsen and M. Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge University Press, 2016. Cited on pages 80, 81.

[85] G. Pleiss, J. R. Gardner, K. Q. Weinberger, and A. G. Wilson. Constant-Time Predictive Distributions for Gaussian Processes. In *International Conference on Machine Learning*, 2018. Cited on pages 90, 147.

[86] G. Pleiss, M. Jankowiak, D. Eriksson, A. Damle, and J. R. Gardner. Fast Matrix Square Roots with Applications to Gaussian Processes and Bayesian Optimization. In *Advances in Neural Information Processing Systems*, 2020. Cited on pages 90, 147, 148.

[87] A. Rahimi and B. Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems*, 2008. Cited on pages 79, 82.

[88] C. E. Rasmussen and J. Quinoñero-Candela. Healing the Relevance Vector Machine through Augmentation. In *International Conference on Machine Learning*, 2005. Cited on page 90.

[89] C. E. Rasmussen and C. K. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Cited on pages 68, 73.

[90] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: Automating the Finite Element Method by Composing Abstractions. *ACM Transactions on Mathematical Software*, 43(3):1–27, 2016. Cited on page 121.

[91] H. Rosenbrock. An Automatic Method for Finding the Greatest or Least Value of a Function. *The Computer Journal*, 3(3):175–184, 1960. Cited on page 140.

[92] D. Russo and B. Van Roy. Learning to Optimize via Information-Directed Sampling. In *Advances in Neural Information Processing Systems*, 2014. Cited on pages 47, 147.

[93] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen. A Tutorial on Thompson Sampling. *Foundations and Trends in Machine Learning*, 11(1):1–96, 2018. Cited on page 46.

[94] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2003. Cited on pages 147, 148.

[95] A. K. Saibaba, A. Alexanderian, and I. C. F. Ipsen. Randomized Matrix-Free Trace and Log-Determinant Estimators. *Numerische Mathematik*, 137(2):353–395, 2017. Cited on page 148.

[96] D. Sanz-Alonso and R. Yang. The SPDE Approach to Matérn Fields: Graph Representations. *Statistical Science*, 2021. Cited on page 127.

[97] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015. Cited on page 90.

[98] A. Slivkins. Introduction to Multi-Armed Bandits. *Foundations and Trends in Machine Learning*, 12(1–2):1–286, 2019. Cited on pages 38, 40, 42.

[99] A. J. Smola and R. Kondor. Kernels and Regularization on Graphs. In *Learning Theory and Kernel Machines*, pages 144–158. Springer, 2003. Cited on page 126.

[100] E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. In *Advances in Neural Information Processing Systems*, 2006. Cited on page 87.

[101] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, 2012. Cited on page 47.

[102] A. Solin and M. Kok. Know Your Boundaries: Constraining Gaussian Processes by Variational Harmonic Features. In *Artificial Intelligence and Statistics*, 2019. Cited on pages 83, 150.

[103] A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and Interpolation of the Ambient Magnetic Field by Gaussian Processes. *IEEE Transactions on Robotics*, 34(4):1112–1127, 2018. Cited on pages 84, 150.

[104] A. Solin and S. Särkkä. Hilbert Space Methods for Reduced-Rank Gaussian Process Regression. *Statistics and Computing*, 30(2):419–446, 2020. Cited on page 84.

[105] D. Spielman. Spectral Graph Theory. In *Combinatorial Scientific Computing*, pages 495–517. CRC Press, 2012. Cited on page 121.

[106] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design. In *International Conference on Machine Learning*, 2010. Cited on page 47.

[107] B. K. Sriperumbudur and Z. Szabó. Optimal Rates for Random Fourier Features. In *Advances in Neural Information Processing Systems*, 2015. Cited on page 82.

[108] R. S. Strichartz. Analysis of the Laplacian on the Complete Riemannian Manifold. *Journal of Functional Analysis*, 52(1):48–79, 1983. Cited on page 108.

[109] D. J. Sutherland and J. Schneider. On the Error of Random Fourier Features. In *Uncertainty in Artificial Intelligence*, 2015. Cited on pages 82, 96.

[110] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, 2018. Cited on page 34.

[111] S. Suzuki, S. Takeno, T. Tamura, K. Shitara, and M. Karasuyama. Multi-objective Bayesian Optimization using Pareto-frontier Entropy. In *International Conference on Machine Learning*, 2020. Cited on page 147.

[112] T. Tao. *An Introduction to Measure Theory.* American Mathematical Society, 2011. Cited on page 133.

[113] W. R. Thompson. On the Likelihood that One Unknown Probability Exceeds Another in View of the Evidence of Two Samples. *Biometrika*, 25(3):285–294, 1933. Cited on page 46.

[114] M. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Artificial Intelligence and Statistics*, 2009. Cited on pages 87, 89.

[115] M. J. Urry and P. Sollich. Random Walk Kernels and Learning Curves for Gaussian Process Regression on Random Graphs. *Journal of Machine Learning Research*, 14(1):1801–1835, 2013. Cited on page 124.

[116] M. van der Wilk, V. Dutordoir, S. T. John, A. Artemev, V. Adam, and J. Hensman. A Framework for Interdomain and Multioutput Gaussian Processes. Technical report, PROWLER.io, 2020. Cited on page 147.

[117] C. Villani. *Optimal Transport: Old and New.* Springer, 2008. Cited on pages 28, 92, 93, 110.

[118] Z. Wang, C. Gehring, P. Kohli, and S. Jegelka. Batched Large-scale Bayesian Optimization in High-dimensional Spaces. In *Artificial Intelligence and Statistics*, 2018. Cited on pages 90, 91.

[119] H. Wendland. *Scattered Data Approximation.* Cambridge University Press, 2004. Cited on page 80.

[120] P. Whittle. On Stationary Processes in the Plane. *Biometrika*, 41:434–449, 1954. Cited on pages 84, 113, 114, 116.

[121] P. Whittle. Stochastic Processes in Several Dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994, 1963. Cited on pages 84, 113, 114, 116.

[122] J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Efficiently Sampling Functions from Gaussian Process Posteriors. In *International Conference on Machine Learning*, 2020. Cited on page 23.

[123] J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Pathwise Conditioning of Gaussian Processes. *Journal of Machine Learning Research*, 22(105):1–47, 2021. Cited on page 23.

[124] J. Xu and L. Zikatanov. Algebraic Multigrid Methods. *Acta Numerica*, 26:591–721, 2017. Cited on page 148.

[125] F. X. Yu, A. T. Suresh, K. Choromanski, D. Holtmann-Rice, and S. Kumar. Orthogonal Random Features. In *Advances in Neural Information Processing Systems*, 2016. Cited on page 82.

[126] S. Zelditch. *Eigenfunctions of the Laplacian on a Riemannian Manifold.* American Mathematical Society, 2017. Cited on page 116.