Imperial College London

Imperial College London
Department of Computing

Decision-Making with Gaussian Processes: Sampling Strategies and Monte Carlo Methods

James T. Wilson

This dissertation is submitted for the degree of
Doctor of Philosophy

February 27, 2023

Declaration

This thesis is an original work of the author, except where otherwise indicated.

Copyright Notice

Acknowledgements

I am indebted to a great number of people who have helped to shape my views on research and, in many cases, life in general.

To begin, I am deeply grateful to the many advisors I have been privileged to learn from over the years. To Kilian Q. Weinberger, who taught me to ask questions; to Amos J. Storkey, who taught me to be thorough; and, to Frank Hutter who taught me to see the bigger picture. Above all though, I would like to thank Marc P. Deisenroth, who taught me how to balance, how to lead, and how to follow.

Second, I would like to thank my wonderful colleagues and collaborators: Alex Terenin, Sanket Kamthe, Hugh Salimbeni, Viacheslav Borovitskiy, Peter Mostowsky, Steindor Saemundsson, and Riccardo Moriconi.

Lastly, I would like to thank my friends and family for their continued love and support. Sharing this ride with you has made it all the more bearable and exciting. To my parents: Thank you. If a fraction of your wisdom and patience rub off on me, I will consider myself a lucky man.

Preface

Recent years have seen an explosion in popularity of machine learning (ML) algorithms for aiding or, even, automating solutions to complex, real-world problems. While the ways in which these algorithms manifest varies by case, a stereotypical recipe is as follows: i) make inferences about the generative process that produced a given dataset; ii) predict what new data points might look like based on these inferences; iii) decide what to do by predicting the consequences of feasible actions.

This process is made difficult by the fact that the data at hand rarely allows us to fully disambiguate the aforementioned generative process. Even when outcomes are believe to be deterministic, then, it follows that we must act in the face of uncertainty. Indeed, especially when it comes to making decisions, careful treatment of uncertainty is often paramount. Chapter 1 therefore acts as a technical primer for Bayesian decision theory (BDT), a unified framework for decision-making with uncertainty. The goal of this chapter will be to sketch core arguments in sufficient mathematical detail for the reader to understand the general flavor of axiomatic approaches to BDT.

Having established these arguments in some detail, Chapter 2 proceeds to further investigate the Bayesian approach to optimization. We begin by demonstrating how various well-known methods can be seen as direct applications of BDT. Having done so, we then redirect our attention to the so-called *inner optimization problem*—the task of finding the best queries to make at each step during search—which will be the focus of this chapter. Here, we will explore various methods for efficiently solving inner optimization problems and show how these improvements lead to overall performance gains.

Due to the importance of well-calibrated predictive uncertainty and prevalence of intractable integrals, Gaussian processes (GPs) and Monte Carlo methods both play integral roles in popular applications of BDT. It sometimes happens, however, that the samples we required are sufficiently high-dimensional that traditional techniques become exorbitantly expensive. To this end, Chapter 3 introduces an alternative interpretation of Gaussian process posteriors which better lends itself to sampling. The resulting family of generative strategies is shown to produce high-quality samples at a fraction of the usual cost and brings with it a host of additional benefits.

Relationship to published papers

The content of this thesis revolves around two separate, but closely topics. Chapter 1 is intended as review and contains no technical contributions; its core arguments were adapted from Fishburn (1970).

The first line of inquiry, published as Wilson et al. (2018), is reported in Chapter 2. As is, I suspect, true in most fields, research on Bayesian optimization has often revolved around creating tools to solve new problems and improve performance on old ones. Relatively little, however, has been said about how we should go about using the tools already at our disposal. Hence, a major goal of this work was to study best practices, so that we may collectively sharpen our tools. In this regard, I hope to have succeeded: the community at large seems to be increasingly cognizant of inner optimization problems and methods championed in this work have gone on to serve as cornerstones in popular open-source packages such as BoTorch (Balandat et al., 2020).

Regarding the technical contributions, two comments are in order. First, pathwise gradient estimators of certain Gaussian-process-based acquisition functions were previously studied by Wang et al. (2016) and Wu and Frazier (2016). On this front, I therefore sought to distill and extend upon these authors' findings. Second, the question of acquisition function submodularity was first posed by David Ginsbourger. Alongside Dario Azzimonti and Henry Wynn, the four of us spent part of an afternoon sketching a bespoke proof of submodularity for Probability of Improvement. Later, I returned to analyzed this problem from scratch, leading to the general result presented in Section 2.5.

The second line of inquiry, published as Wilson et al. (2020) and later Wilson et al. (2021), was a deeply collaborative effort that could not have been completed without the help of my coauthors. Regarding the impact of these works, the former was one of two papers awarded an Honorable Mention for Best Paper at ICML 2020. The general family of techniques on offer allows us to accurately sample from Gaussian process posteriors in linear time, opening the door for real-world applications that would otherwise prove prohibitively expensive.[1]

This project began as two separate threads that were eventually woven together by Alex Terenin. I was responsible for many of the key innovations present in these works, however everyone played a role in refining these ideas into precise technical statements. This is particularly true of Section 3.1.3 and Section 3.5, which I merely

[1] Here, scaling is reported as a function of the number of test locations.

helped motivate and write. Lastly, the software offerings listed in Chapter 3 were written by myself.

Contents

CHAPTER 1

Decision Theory

Throughout this work, we will explore how probability and statistics can be used to automate decisions made in the real world. To do so, we first require a mathematical framework for answering questions such as: *Given what is currently known about a decision-making problem, what is an optimal choice of action?*

The purpose of this chapter is to show how these formalisms are derived. While comprehensive treatment of related topics is beyond the scope of the present work, we will review key arguments behind normative theories of choice and, in particular, Bayesian decision theory. These arguments take the form of axiomatic systems in which rudimentary assumptions regarding an agent's preferences are used to deduce idealized patterns of behavior. These findings will culminate in a proof of the claim that a rational agent's preferences correspond to a unique pairing of a probability measure and a (equivalence class of) utility function; and, further, that optimal actions maximize the agent's conditional expected utility.

This result will set the stage for Bayesian decision theoretic algorithms discussed in later chapters, which adhere to a simple recipe: (i) model the agent's subjective probability measure and utility function; (ii) find an action that maximizes the agent's conditional expected utility under the model. Rather than starting with practical algorithms, we therefore begin by reviewing the formal developments that precede them. To do so, let us first agree upon a firmer definition of *preference*.

Intuitively speaking, given two objects $f, g \in \mathcal{F}$, we say that the agent prefers f to g if, when compelled to choose between the two, they would select f over g. We will temporarily leave $\mathcal{F}$ generic, since different schools of decision theory formulate preferences over different classes of objects. In order for preferences to lead to coherent patterns of behavior, some assumptions must be made about how an agent

compares different pairs of choices. Perhaps the most basic of these assumptions is that pairwise preferences reflect an internally consistent way of ranking the set of choices $\mathcal{F}$. We may formalize this intuition as a simple, binary relation that induces a weak ordering on $\mathcal{F}$.

Definition 1.1 (Preference)**.** *Let $\succ$ be a binary relation on $\mathcal{F}$. Then, $\succ$ is a preference relation if, for all $f, g, h \in \mathcal{F}$, it follows that*

i. Asymmetric: $f \succ g \implies g \not\succ f$.

ii. Negatively transitive: $f \not\succ g$ and $g \not\succ h \implies f \not\succ h$.

Note that, by construction, $\succ$ is both transitive and irreflexive. This definition guarantees that it is possible to identify one or more "optimal" choice $f^* \in \mathcal{F}$ for which there is no alternative $g \in \mathcal{F}$ such that $g \succ f^*$. Two derived, binary relations will be useful here. First, a transitive and strongly connected *preference-indifference relation* $\succeq$, where $f \succeq g \iff g \not\succ f$. Second, a transitive, symmetric, and reflexive *indifference relation* $\sim$, where $f \sim g \iff f \succeq g$ and $g \succeq f$.

This representation of preference is the starting place for many axiomatic approaches to decision-making. Our interest ultimately lies in the Bayesian decision theory put forth by Leonard J. Savage. Like many decision theorists, however, Savage was greatly influenced by the work of von Neumann and Morgenstern (VNM). We, therefore, detour and first explore VNM's arguments, which will give us an opportunity to familiarize ourselves with some concepts (and notations) in a simpler setting.

1.1 von Neumann–Morgenstern utility theory

This section examines the axiomatic decision theory put forth in *Theory of Games and Economic Behavior* (von Neumann and Morgenstern, 1944). As a starting place, VNM define preferences over *lotteries* $\mathcal{P}$, defined as a family of distributions on a given σ-algebra for $\mathcal{Y}$. Prior to introducing this theory, two brief asides are in order. First, we will write $p \succ \gamma$ to indicate that $p \in \mathcal{P}$ is preferred to the degenerate lottery

$$\delta_\gamma(\gamma') = \begin{cases} 1 & \text{if } \gamma' = \gamma, \\ 0 & \text{otherwise .} \end{cases} \tag{1.1}$$

Second, we must define the notion of a compound lottery. For our purposes, this concept is best understood in terms of *mixture sets* (Herstein and Milnor, 1953).

Definition 1.2 (Mixture set)**.** *A set $\mathcal{X}$ is said to be a mixture set if, for any $x, y \in \mathcal{X}$ and $\alpha, \beta \in [0, 1]$, there is an element of $\mathcal{X}$, denoted $\alpha x + (1 - \alpha)y \in \mathcal{X}$, that satisfies:*

M1. $1x + (1 - 1)y = x$.

M2. $\alpha x + (1 - \alpha)y = (1 - \alpha)y + \alpha x$.

M3. $\alpha[\beta x + (1 - \beta)y] + (1 - \alpha)y = (\alpha\beta)x + (1 - \alpha\beta)y$.

In essence, a mixture set is a set that is closed under some abstract generalization of a convex combination. As a familiar example, convex sets in real vector spaces are mixture sets in which combinations $\alpha x + (1-\alpha)y$ are formed using scalar multiplication and vector addition. In contrast, when discussing compound lotteries $\alpha p + (1-\alpha)q$ with $p, q \in \mathcal{P}$, we will be interested in mixture sets defined in terms of probabilistic mixtures of distributions. Two immediate consequences of M1–M3 that will be useful in later developments are

M4. $\alpha x + (1-\alpha)x = x$.

M5. Let $\eta = \alpha\beta + (1-\alpha)\epsilon$ for an arbitrary choice of $\epsilon \in [0,1]$. Then,

$$\alpha[\beta x + (1-\beta)y] + (1-\alpha)[\epsilon x + (1-\epsilon)y] = \eta x + (1-\eta)y.$$

These details in order, we are now ready to explore von Neumann and Morgenstern's approach to decision theory.

Theorem 1.3 (von Neumann–Morgenstern expected utility)**.** *Let $\mathcal{P}$ be a set of probability measures on a given σ-algebra for $\mathcal{Y}$ and suppose that, for all $p, q, r \in \mathcal{P}$,*

A1. *$\succ$ on $\mathcal{P}$ is a preference relation.*

A2. *If $p \succ q$, then for all $\alpha \in (0,1]$ it follows that*

$$\alpha p + (1-\alpha)r \succeq \alpha q + (1-\alpha)r.$$

A3. *If $p \succ q \succ r$, then there exist $\alpha, \beta \in (0,1)$ for which*

$$\alpha p + (1-\alpha)r \succ q \succ \beta p + (1-\beta)r.$$

Then, there exists a utility function $u : \mathcal{Y} \to \mathbb{R}$ such that

$$p \succ q \iff \mathbb{E}_p[u(\gamma)] > \mathbb{E}_q[u(\gamma)]. \tag{1.2}$$

Further, the function u is unique up to positive affine transformations.

Some discussion of A2 and A3 is in order. The *independence* axiom, A2, posits that the agent's preference $p \succ q$ does not change when said lotteries are equivalently combined with alternatives that the agent is indifferent to. Said differently, A2 asserts that the agent compares $\alpha p + (1-\alpha)r$ and $\alpha q + (1-\alpha)r$ solely in terms of how they differ.

The Archimedean property of real numbers states that, given any two positive numbers x and y, there exists a third number a such that $ax > y$. Fittingly, then, A3 is often referred to as the *Archimedean* axiom. A3 states that there is no lottery p so vastly superior to all other lotteries q that the agent's preference $p \succ q$ cannot be reversed by mixing p with a third lottery r satisfying $q \succ r$; likewise, there is no r so vastly inferior to all q that $q \succ r$ cannot be reversed by mixing r with p. Lastly, we note that A3 is sometimes replaced by a closely related *continuity* axiom

A3*. If $p \succeq q \succeq r$, then there exists an $\alpha \in [0,1]$ for which $\alpha p + (1-\alpha)r \sim q$.

The following two sections outline the general approach for proving Theorem 1.3. Throughout, we will restrict our attention to the special case where $\mathcal{P} = \mathcal{P}_s$ is defined as the set of all simple probability measures on $\mathcal{Y}$, i.e. of all distributions with support over a finite number of outcomes $\gamma \in \mathcal{Y}$. As we will soon see, this assumption allows for an intuitive and instructive proof. For comprehensive treatment of this topic, see Fishburn (1970, Chapters 8 and 10) or Kreps (1988, Chapter 5).

1.1.1 Proof of Theorem 1.3: Order-preserving functions on $\mathcal{P}_s$

We begin our proof of Theorem 1.3 by showing that there is a function that quantifies the agent's preferences in a consistent manner. Specifically, we shall say that a function $U : \mathcal{P}_s \to \mathbb{R}$ is *order-preserving* with respect to $\succ$ if and only if

$$\forall p, q \in \mathcal{P}_s : \; p \succ q \iff U(p) > U(q). \tag{1.3}$$

Notice that $\mathcal{P}_s$ is a mixture set by construction. To better see the implications of this fact, let us introduce a few key lemmas that will carry much of the proof.

Lemma 1.4. *If $\succ$ satisfies A1–A3 on a mixture set $\mathcal{P}_s$, then:*

- **a.** *If $p \succ q$ and $1 \geq \alpha > \beta \geq 0$, then $\alpha p + (1-\alpha)q \succ \beta p + (1-\beta)q$.*
- **b.** *If $p \succeq q \succeq r$ and $p \succ r$, then there exists a unique $\alpha \in [0,1]$ for which $q \sim \alpha p + (1-\alpha)r$.*
- **c.** *If $p \sim q$ and $\alpha \in [0,1]$, then $\alpha p + (1-\alpha)r \sim \alpha q + (1-\alpha)r$ for every $r \in \mathcal{P}_s$.*

Proof This sketch was abridged from Kreps (1988, page 46). When $\alpha = 1$ or $\beta = 0$, Lemma 1.4a is trivial; so, suppose $\alpha, \beta \in (0,1)$ and write $s = \alpha p + (1-\alpha)q$. Then,

$$\begin{aligned} p \succ q &\overset{A2}{\iff} s \succ \alpha q + (1-\alpha)q \overset{M4}{\iff} s \succ q \\ &\overset{A2}{\iff} \tfrac{\beta}{\alpha}s + (1-\tfrac{\beta}{\alpha})s \succ \tfrac{\beta}{\alpha}s + (1-\tfrac{\beta}{\alpha})q \\ &\overset{M3}{\iff} \tfrac{\beta}{\alpha}s + (1-\tfrac{\beta}{\alpha})s \succ \beta p + (1-\beta)q \\ &\overset{M4}{\iff} \alpha p + (1-\alpha)q \succ \beta p + (1-\beta)q. \end{aligned} \tag{1.4}$$

For Lemma 1.4b: $p \succ r$ and Lemma 1.4a imply that there is at most one $\alpha \in [0,1]$ for which $q \sim \alpha p + (1-\alpha)r$. If $p \sim q$, then $\alpha = 1$ suffices. If $q \sim r$, then $\alpha = 0$ suffices. Henceforth, suppose $p \succ q \succ r$ and define $\alpha = \sup\{\beta \in [0,1] : q \succ \beta p + (1-\beta)r\}$. Of the three possible relations ($\succ$, $\prec$, and $\sim$) between q and $\alpha p + (1-\alpha)r$, all but $q \sim \alpha p + (1-\alpha)r$ contradict A3.

For Lemma 1.4c: The claim is trivially satisfied when $p \sim q$ for all $p, q \in \mathcal{P}_s$. Hence, let p and q be such that there exists an $s \in \mathcal{P}_s$ satisfying $s \succ p \sim q$ and define

$$t(\beta) = \alpha[\beta s + (1-\beta)q] + (1-\alpha)r. \tag{1.5}$$

It follows by A2 that $\beta s + (1-\beta)q \succ q$ and $t(\beta) \succ \alpha p + (1-\alpha)r$ for all $\beta \in (0,1]$. Now, suppose $\alpha p + (1-\alpha)r \succ \alpha q + (1-\alpha)r$. A3 would then imply that, for every $\beta \in (0,1]$, there exists an $\epsilon \in (0,1)$ such that

$$\begin{aligned} \alpha p + (1-\alpha)r &\succ \epsilon t(\beta) + (1-\epsilon)[\alpha q + (1-\alpha)r] \\ &= \alpha[\epsilon\beta s + (1-\epsilon\beta)q] + (1-\alpha)r = t(\epsilon\beta), \end{aligned} \tag{1.6}$$

where the second line is obtained by simply unpacking $t(\beta)$ and cancelling like terms. Since $\epsilon\beta > 0$, however, this statement directly contradicts the aforementioned implications of A2. A similar contradiction arises under the hypothesis that $\alpha p + (1-\alpha)r \prec \alpha q + (1-\alpha)r$. ■

Returning to the matter of order-preserving functions on $\mathcal{P}_s$, we begin by showing that $\delta_{\text{sup}} \succeq p \succeq \delta_{\text{inf}}$ for all $p \in \mathcal{P}_s$, where we have defined

$$\delta_{\text{sup}} \in \{\delta_\gamma : \gamma \succeq \lambda, \forall \lambda \in \mathcal{Y}\} \qquad \delta_{\text{inf}} \in \{\delta_\gamma : \mu \succeq \gamma, \forall \mu \in \mathcal{Y}\}. \tag{1.7}$$

Let n be the number of outcomes with support under p. By construction, then, the claim holds for $n = 1$. When $n > 1$, use Lemma 1.4 followed by M5 to show that

$$p = \sum_{i=1}^{n} p_i \gamma_i \sim \sum_{i=1}^{n} p_i[\alpha_i \delta_{\text{sup}} + (1-\alpha_i)\delta_{\text{inf}}] = \alpha\delta_{\text{sup}} + (1-\alpha)\delta_{\text{inf}}, \tag{1.8}$$

for some $\alpha \in [0,1]$. By definition, $\delta_{\text{sup}} \succeq \delta_{\text{inf}}$. Accordingly, it follows by A2 that $\delta_{\text{sup}} \succeq \alpha\delta_{\text{sup}} + (1-\alpha)\delta_{\text{inf}} \succeq \delta_{\text{inf}}$ for all $\alpha \in [0,1]$ and, in turn, that $\delta_{\text{sup}} \succeq p \succeq \delta_{\text{inf}}$ for all $p \in \mathcal{P}_s$. Notice that, when $\delta_{\text{sup}} \sim \delta_{\text{inf}}$, the agent is indifferent on $\mathcal{P}_s$, whereupon any constant function U trivially satisfies Theorem 1.5. Henceforth, suppose $\delta_{\text{sup}} \succ \delta_{\text{inf}}$.

Define U as the function that assigns to each $p \in \mathcal{P}_s$ a value $U(p) \in [0,1]$ for which

$$p \sim U(p)\delta_{\text{sup}} + (1-U(p))\delta_{\text{inf}}. \tag{1.9}$$

Since $\delta_{\text{sup}} \succ \delta_{\text{inf}}$, we may now use transitivity followed by Lemma 1.4a to show that

$$\begin{aligned} p \succ q \iff & [U(p)\delta_{\text{sup}} + (1-U(p))\delta_{\text{inf}}] \succ [U(q)\delta_{\text{sup}} + (1-U(q))\delta_{\text{inf}}] \\ \iff & U(p) > U(q). \end{aligned} \tag{1.10}$$

Moreover, since Lemma 1.4b implies there is only one value $U(p)$ that satisfies (1.9), it follows that all functions on $\mathcal{P}_s$ that agree with $\succ$ are equivalent up to multiplication by a positive number or addition of a scalar. Hence, there exists a function $U : \mathcal{P}_s \to \mathbb{R}$ that agrees with the agent's preferences on $\mathcal{P}_s$ and this function is unique.

1.1.2 Proof of Theorem 1.3: Expected utility representation

Having shown that there exists an order-preserving function $U : \mathcal{P}_s \to \mathbb{R}$, let us now prove that U takes the form of an expected utility, i.e. that there exists a function $u : \mathcal{Y} \to \mathbb{R}$ such that

$$U(p) = \mathbb{E}_p[u(\gamma)] = \sum_{\gamma \in \boldsymbol{\gamma}} p(\gamma)u(\gamma), \text{ for all } p \in \mathcal{P}_s,$$

where $\boldsymbol{\gamma} = \{\gamma : p(\gamma) > 0\}$ is the set of outcomes with support under p. We begin by showing that U is linear in the sense that

$$U(\alpha p + (1-\alpha)q) = \alpha U(p) + (1-\alpha)U(q). \tag{1.11}$$

Use transitivity and the definition of U in (1.9) to replace p and q in $\alpha p + (1-\alpha)q$ with the equivalent lotteries

$$p \sim U(p)\delta_{\text{sup}} + (1 - U(p))\delta_{\text{inf}} \qquad q \sim U(q)\delta_{\text{sup}} + (1 - U(q))\delta_{\text{inf}}, \tag{1.12}$$

Simplifying the resulting expression with the help of M5, we obtain

$$\begin{aligned} \alpha p + (1-\alpha)q \sim \\ [\alpha U(p) + (1-\alpha)U(q)]\delta_{\text{sup}} + [1 - \alpha U(p) - (1-\alpha)U(q)]\delta_{\text{inf}}. \end{aligned} \tag{1.13}$$

At the same time, however, (1.9) also implies that

$$\alpha p + (1-\alpha)q \sim U(\alpha p + (1-\alpha)q)\delta_{\text{sup}} + [1 - U(\alpha p + (1-\alpha)q)]\delta_{\text{inf}}. \tag{1.14}$$

In order for both statements to hold, it follows that U must be linear as in (1.11). Now that we know U is linear, we are ready to prove that U is an expected utility. Conceptually, we will do so by decomposing $p \in \mathcal{P}_s$ and appealing to (1.11).

First, define $u(\gamma) = U(\delta_\gamma)$ and let $\boldsymbol{\gamma}_k = \{\gamma_{n-k+1}, \ldots, \gamma_n\}$ be the final k elements of $\boldsymbol{\gamma}$ (ordered arbitrarily). Further, denote the restriction of p to $\boldsymbol{\gamma}_k$ as

$$p_k(\ \cdot\ ) = \frac{p(\ \cdot\ )\delta_{\boldsymbol{\gamma}_k}(\ \cdot\ )}{p(\boldsymbol{\gamma}_k)}, \text{ such that } p(\gamma) = p(\gamma)\delta_{\gamma_{1:n-k}}(\gamma) + p(\boldsymbol{\gamma}_k)p_k(\gamma). \tag{1.15}$$

Proceeding by induction on the number of outcomes n, it follows that:

- Base case ($n = 1$):

$$U(p) = U(\delta_{\gamma_1}) = u(\gamma_1) = \sum_{i=1}^{1} p(\gamma_i)u(\gamma_i) \tag{1.16}$$

- Inductive step ($n > 1$):

$$U(p) = p(\gamma_1)u(\gamma_1) + p(\boldsymbol{\gamma}_{n-1})U(p_{n-1}) = \sum_{i=1}^{n} p(\gamma_i)u(\gamma_i). \tag{1.17}$$

Hence, the agent's preferences correspond with (the expectation of) a unique utility function $u : \mathcal{Y} \to \mathbb{R}$ when $\succeq$ satisfies A1–A3 on $\mathcal{P}_s$. In the next section, we will see how a related set of axioms enables us to derive not only a unique utility function but also a unique probability measure that agrees with $\succ$.

1.2 Savage utility theory

Often regarded as "the crowning glory of choice theory" (Kreps, 1988, page 120), Leonard J. Savage's theory of expected utility both unites and refines ideas put forth by earlier theorists. Coming off of von Neumann and Morgenstern's work, Savage was concerned with the authors' assumption that probabilities are objective and known to the agent. At the time, the philosophical interpretation of probability — situated at the center of the conflict between frequentist and Bayesian probabilists — was a heavily contested topic.

Unsatisfied with the premise of VNM's theory, Savage proposed an expanded set of axioms that simultaneously explains for both (subjective) probabilities and (expected) utilities. We will return to this topic later in this section. For now, it suffices to intuit that the agent's willingness to enter into different wagers evidences their belief about whether one event is more likely than another or vice versa and that a quantitative probability measure may be derived on the basis of these comparisons.

Seeing as we may no longer start by defining preferences over lotteries (i.e. random outcomes adhering to known distributions), we require a slightly more nuanced model of decision-making. Let *states* $\omega \in \Omega$ be defined as the collection of covariates that determine the outcome $\gamma = f(\omega)$ of an action $f \in \mathcal{F}$, where $\mathcal{F} \subseteq \mathcal{Y}^{\Omega}$. From here, Savage proceeds to define a preference relation $\succ$ on $\mathcal{F}$. Paralleling the degenerate lotteries δ_γ from the previous section, Savage assumes that for any outcome $\gamma \in \mathcal{Y}$ there exists a constant action $f_\gamma(\omega) = \gamma$ for all $\omega \in \Omega$. As before, we will simply write γ in place of f_γ where possible without introducing ambiguity.

Two additional constructions will prove to be immediately useful. First, reminiscent of the compound lotteries seen in Section 1.1, define the *compound action*

$$\mathbf{x}_A(f, g) = \begin{cases} f(\omega) & \text{if}\, \omega \in A \\ g(\omega) & \text{otherwise} \end{cases} \tag{1.18}$$

as the action which yields $f(\omega)$ for all states $\omega \in A$ and $g(\omega)$ otherwise. Second, given an event $A \subseteq \Omega$ and actions $f, g \in \mathcal{F}$, define the *conditional preference* relation

$$f \succ g \mid A \iff \mathbf{x}_A(f, h) \succ \mathbf{x}_A(g, h), \text{ for all } h \in \mathcal{F}. \tag{1.19}$$

In words, f is preferred to g given A if and only if, when both actions are modified to yield the same results for all $\omega \notin A$, the compound of f is preferred to that of g. We shall say that an event $A \subseteq \Omega$ is *null* if $f \sim g \mid A$ for all $f, g \in \mathcal{F}$.[1]

[1] Null events correspond to measure zero sets under a yet-to-be-introduced distribution $\dot{p}$.

Now is a good time for us to properly introduce an algebra of sets $\mathcal{A}$ for Ω. Following de Finetti (1937), Savage exclusively focuses on finitely additive probability measures. Seeing as the purpose of this chapter is one of exposition, we will largely stick to his original arguments. However, Villegas (1964) would later show that this position can be generalized to the countably additive case under the mild assumption of monotone continuity. For this reason, we deem it appropriate to refer to $\mathcal{A}$ as a σ-algebra. Further, since Savage's approach is specifically tailored to cases where Ω is infinite and where non-null events $A \in \mathcal{A}$ can be partitioned into arbitrarily fine, non-null subsets, we define $\mathcal{A}$ as the set of all subsets of Ω and will sometimes write $A \subseteq \Omega$ for $A \in \mathcal{A}$. Those curious about finite Ω should see Kraft et al. (1959), while readers interested in overviewing these topics should see Fishburn (1986) and references contained therein. These definitions in place, we are ready to overview Savage's theory of expected utility.

Theorem 1.5 (Savage expected utility)**.** *Let $f(\omega) = \gamma \in \mathcal{Y}$ be the result of carrying out an action $f \in \mathcal{F}$ in a state $\omega \in \Omega$. Suppose that $\succ$ on $\mathcal{F}$ satisfies:*

P1. *$\succ$ on $\mathcal{F}$ is a preference relation.*

P2. *For all events $A \subseteq \Omega$ and actions $f, f', g, h \in \mathcal{F}$,*

$$\mathbf{x}_A(f, g) \succ \mathbf{x}_A(f', g) \iff \mathbf{x}_A(f, h) \succ \mathbf{x}_A(f', h).$$

P3. *For all non-null events $A \subseteq \Omega$, actions $f \in \mathcal{F}$, and outcomes $\mu, \lambda \in \mathcal{Y}$*

$$\mu \succ \lambda \mid A \iff \mu \succ \lambda.$$

P4. *For all events A and B and for all pairs of outcome $\mu \succ \lambda$ and $\mu' \succ \lambda'$*

$$\mathbf{x}_A(\mu, \lambda) \succ \mathbf{x}_B(\mu, \lambda) \iff \mathbf{x}_A(\mu', \lambda') \succ \mathbf{x}_B(\mu', \lambda').$$

P5. *There exist outcomes $\mu, \lambda \in \mathcal{Y}$ such that $\mu \succ \lambda$.*

P6. *If $f \succ g$, it follows for any outcome $\gamma \in \mathcal{Y}$ that there exists a finite partition $\{\Omega_1, \ldots, \Omega_n\}$ of Ω so that*

$$f \succ \mathbf{x}_{\Omega_i}(\gamma, g) \text{ and } \mathbf{x}_{\Omega_i}(\gamma, f) \succ g, \text{ for all } i = 1, \ldots, n.$$

P7. *For all events $A \subseteq \Omega$ and actions $f, g \in \mathcal{F}$,*

$$f \succ \gamma \mid A, \forall \gamma \in g(A) \implies f \succeq g \mid A.$$

Then, it follows that:

a) *There is one and only one probability measure $\dot{p}$ on $\mathcal{A}$ that agrees with the more-likely-than binary relation $\dot{\succ}$, defined for all $A, B \subseteq \Omega$ and $\mu, \lambda \in \mathcal{Y}$ with $\mu \succ \lambda$ as $A \dot{\succ} B \iff \mathbf{x}_A(\mu, \lambda) \succ \mathbf{x}_B(\mu, \lambda)$, in the sense that*

$$A \dot{\succ} B \iff \dot{p}(A) > \dot{p}(B), \text{ for all } A, B \in \Omega.$$

Further, for all $C \subseteq \Omega$ and $c \in [0,1]$, there exists a $D \subseteq C$ so that $\dot{p}(C) = c\dot{p}(D)$.

b) *There exists a unique*[2] *utility function $u : \mathcal{Y} \to \mathbb{R}$ for which*

$$f \succ g \iff \mathbb{E}_{\dot{p}}[u(f(\omega))] > \mathbb{E}_{\dot{p}}[u(g(\omega))], \text{ for all } f, g \in \mathcal{F}.$$

As before, let us begin by unpacking the axioms shown above. Note that many of these axioms tacitly assert that (the distribution of) ω is independent of the chosen $f \in \mathcal{F}$. Axiom P2, which closely resembles the independence axiom (A2), dictates that: the agent's preference for f over g should only depend on those states $\omega \in \Omega$ for which $f(\omega) \neq g(\omega)$. In light of this axiom, the conditional preference $f \succ g \mid A$ can now be interpreted to mean that f is preferred to g when Ω is restricted to A. Together with P2, P3 realizes a core part of Savage's model: the *sure-thing principle.*

Definition 1.6 (Sure-thing principle)**.** *Given a pair of actions $f, g \in \mathcal{F}$ and a non-null event $A \subseteq \Omega$ with complement $A^c = \Omega \setminus A$,*

$$f \succ g \mid A \text{ and } f \succeq g \mid A^c \implies f \succ g. \tag{1.20}$$

In digesting this principle, Savage's original parable proves wonderfully insightful:

> A businessman contemplates buying a certain piece of property. He considers the outcome of the next presidential election relevant. So, to clarify the matter to himself, he asks whether he would buy if he knew that the Democratic candidate were going to win, and decides that he would. Similarly, he considers whether he would buy if he knew that the Republican candidate were going to win, and again finds that he would. Seeing that he would buy in either event, he decides that he should buy, even though he does not know which event obtains, or will obtain, as we would ordinarily say. It is all too seldom that a decision can be arrived at on the basis of this principle, but except possibly for the assumption of simple ordering, I know of no other extralogical principle governing decisions that finds such ready acceptance. Savage (1954, page 21)

Proceeding with the next axiom, P4 ensures that the preference $\mathbf{x}_A(\mu, \lambda) \succ \mathbf{x}_B(\mu, \lambda)$ used to define the more-likely-than relation $\dot{\succ}$ in Theorem 1.5a is independent of the choice of outcomes $\mu, \lambda \in \mathcal{Y}$ so long as $\mu \succ \lambda$. As we will see, this axiom plays a key role in the derivation of a probability measures that agrees with $\succ$. Axiom P5 asserts that the agent is not indifferent on all of $\mathcal{F}$, which helps guarantee that $\dot{p}$ is unique.

P6 is similar, in spirit, to the Archimedean axiom A3: it states that there is no outcome $\gamma \in \mathcal{Y}$ such that we are unable to non-trivially[3] combine an action f with the constant action f_γ without reversing the agent's preference $f \succ g$ (or $g \succ f$). Additional implications of P6, evident in part (ii) of Theorem 1.5a, are as follows: Ω

[2] Up to positive affine transformations.

[3] By *non-trivially*, we mean that $\mathbf{x}_A(\gamma, f)$ is predicated upon a non-null event $A \subseteq \Omega$.

must be uncountable; for all $\omega \in \Omega$, $\dot{p}(\omega) = 0$; and, for every $n \in \mathbb{N}$, there must be an n-event, uniform partition $\{\Omega_1, \ldots, \Omega_n\}$ of Ω such that $\Omega_i \dot{\sim} \Omega_j$ for all $i, j \in [1, n]$.

Finally, P7 together with the previous axioms ensures that utilities are bounded, which enables us to extend arguments from Section 1.1 to cases where the distributions $p \circ f^{-1}$ generated by actions $f \in \mathcal{F}$ have support over an infinite number of outcomes $\gamma \in \mathcal{Y}$. Equipped with these axioms, let us now explore the arguments that lead to Theorem 1.5. Having explored Savage's seven axioms, we now move on to investigate his claims.

1.2.1 Proof of Theorem 1.5: Subjective probability

One of the distinguishing features of Savage's approach to decision theory was his treatment of probability. In discussing VNM, we implicitly assumed that each of the lotteries available to the agent corresponds with a particular ground truth distribution over outcomes that is known a priori by the agent. This view is largely at odds with the subjective interpretation of probability inherent to the Bayesian paradigm. Earlier works, most notably Ramsey (1926) and de Finetti (1937), had already begun to axiomatize subjective probability as a consequence of revealed preference. It was Savage, however, who would ultimately refine these ideas and integrate them into a unified model of choice. To get a feel for what these developments look like, let us revisit the more-likely-than relation previously given in Theorem 1.5a.

Definition 1.7 (More likely than)**.** *Let $\dot{\succ}$ be a binary relation on a σ-algebra $\mathcal{A}$ for Ω. Then, $\dot{\succ}$ is a more-likely-than relation if and only if, for any two events $A, B \in \mathcal{A}$ and outcomes $\mu, \lambda \in \mathcal{Y}$ such that $\mu \succ \lambda$, it follows that*

$$A \dot{\succ} B \iff \mathbf{x}_A(\mu, \lambda) \succ \mathbf{x}_B(\mu, \lambda). \tag{1.21}$$

The intuition here is straightforward: since $\mathbf{x}_A(\mu, \lambda)$ is preferred to $\mathbf{x}_B(\mu, \lambda)$ despite the fact that both actions generate the same outcomes, it follows that the perceived chance of obtaining the favored outcome μ must be greater under $\mathbf{x}_A(\mu, \lambda)$ than under $\mathbf{x}_B(\mu, \lambda)$. Notice that $\dot{\succ}$ is simply a consequence of our original preference relation $\succ$. As we will see, the fact that $\dot{\succ}$ ensues from $\succ$ is critical because $\dot{\succ}$ will serve as the cornerstone for a comparison-based system of subjective probability.

Definition 1.8 (Qualitative probability)**.** *A binary relation $\dot{\succ}$ on a σ-algebra $\mathcal{A}$ for Ω is said to be a qualitative probability if:*

F1. *For all $A \in \mathcal{A}$, $A \dot{\succeq} \emptyset$.*

F2. $\Omega \dot{\succ} \emptyset$.

F3. *$\dot{\succ}$ is a weak order on $\mathcal{A}$.*

F4. *For all $A, B, C \in \mathcal{A}$,*

$$(A \cap C = B \cap C = \emptyset) \implies (A \dot{\succ} B \iff A \cup C \dot{\succ} B \cup C).$$

Properties F1–F4 are necessary conditions for there to exist a probability measure $\dot{p}$ on $\mathcal{A}$ satisfying Kolmogorov's axioms that agrees with $\dot{\succ}$ in the sense that

$$A \dot{\succ} B \iff \dot{p}(A) > \dot{p}(B). \tag{1.22}$$

Suppose $\dot{p}$ exists. Then, F1 and F2 impress upon $\dot{\succ}$ the fact that $\dot{p}(\emptyset) = 0$, $\dot{p}(\Omega) = 1$, and $\dot{p}(A) \geq 0$ for all $A \in \mathcal{A}$. Further, since $\dot{p}$ induces a numerical ordering on $\mathcal{A}$, $\dot{\succ}$ must similarly order $\mathcal{A}$ if (1.22) is to have any hope of holding. The final property, F4, ensures that $\dot{\succ}$ respects the finite additivity of $\dot{p}$.

Perhaps not surprisingly, F1–F4 are consequences of P1–P6. To these, Savage adds the following condition (also due to P1–P6), which helps guarantee that there is one and only probability measure $\dot{p}$ that agrees with $\dot{\succ}$ on $\mathcal{A}$.

F5. If $A, B \subseteq \Omega$ satisfy $A \dot{\succ} B$, then there exists a finite partition $(\Omega_1, \ldots, \Omega_m)$ of Ω such that $A \dot{\succ} B \cup \Omega_i$ for all $i = 1, \ldots, m$.

We will briefly demonstrate how P1, P4, and P5 imply that $\dot{\succ}$ is a weak order on $\mathcal{A}$; derivation of the remaining properties can be found in Fishburn (1970, page 200). P5 guarantees that $\mu \succ \lambda$ for some $\mu, \lambda \in \mathcal{Y}$; hence, the definition of $\dot{\succ}$ in (1.21) and P1 together imply that $\dot{\succ}$ is asymmetric. To see that $\dot{\succ}$ is negatively transitive, suppose $A \dot{\preceq} B$ and $B \dot{\preceq} C$. Still taking $\mu \succ \lambda$, P4 and (1.21) followed by P1 give

$$\mathbf{x}_A(\mu, \lambda) \preceq \mathbf{x}_B(\mu, \lambda) \text{ and } \mathbf{x}_B(\mu, \lambda) \preceq \mathbf{x}_C(\mu, \lambda) \implies \mathbf{x}_A(\mu, \lambda) \preceq \mathbf{x}_C(\mu, \lambda). \tag{1.23}$$

Since (1.23) holds so long as $\mu \succ \lambda$, it follows by (1.21) that

$$A \dot{\preceq} B \text{ and } B \dot{\preceq} C \implies A \dot{\preceq} C. \tag{1.24}$$

Theorem 1.5a. *If $\succ$ satisfies P1–P6 on $\mathcal{F}$, then $\dot{\succ}$ admits F1–F5 on $\mathcal{A}$. In turn, there is one and only one probability measure $\dot{p}$ on $\mathcal{A}$ such that:*

i. *For all $A, B \subseteq \Omega$, $A \dot{\succ} B$ if and only if $\dot{p}(A) > \dot{p}(B)$.*

ii. *For all $A \subseteq \Omega$ and $\varepsilon \in [0, 1]$, there exists a $B \subseteq A$ for which $\dot{p}(B) = \varepsilon \dot{p}(A)$.*

Proof We sketch the proof for the first part of Theorem 1.5a; for details regarding (ii), see Fishburn (1970, page 199). Let $Z(i, 2^n)$ be the set of all possible unions of i distinct components from a uniform partition $\{\Omega_1, \ldots, \Omega_{2^n}\}$ of Ω. For convenience, we will denote a generic part of $Z(i, 2^n)$ as $z(i, 2^n)$. Conceptually, it may be useful to think of $\{\Omega_1, \ldots, \Omega_{2^n}\}$ as the possible sequences produced by n tosses of a fair coin. Statements such as $A \dot{\succ} z(i, 2^n)$ can then be interpreted to mean that the agent prefers a binary lottery predicated on A to its equivalent predicated on $z(i, 2^n)$. The general proof strategy is as follows.

First, use F1–F5 to show that for all $n, m \in \mathbb{W}$, $i \in \{0, \ldots, 2^n\}$, and $j \in \{0, \ldots, 2^m\}$,

$$C \dot{\sim} D \text{ for all } C, D \in Z(i, 2^n) \text{ and } z(i, 2^n) \dot{\sim} z(i2^m, 2^{n+m}). \tag{1.25}$$

Now, let $\kappa : \mathcal{A} \times \mathbb{N} \to \mathbb{W}$ produce the largest integer i for which $A \dot{\succeq} z(i, 2^n)$, i.e.

$$\kappa(A, 2^n) = \sup\Big\{0 \leq i \leq 2^n : A \dot{\succeq} z(i, 2^n)\Big\}. \tag{1.26}$$

Similarly, define $\dot{p}$ as the function

$$\dot{p}(A) = \sup\left\{\frac{\kappa(A, 2^n)}{2^n} : n \in \mathbb{W}\right\}. \tag{1.27}$$

Immediately, we have $\dot{p}(\emptyset) = 0$, $\dot{p}(\Omega) = 1$, $\dot{p}(A) \geq 0$, $\forall A \subseteq \Omega$, and $p(z(i, 2^n)) \geq {}^{i}\!/_{2^n}$. Prior to continuing, let us strengthen this last statement to show that

$$p(z(i, 2^n)) = \frac{i}{2^n}. \tag{1.28}$$

First, observer that (1.25) and transitivity imply

$$z(i, 2^n) \dot{\succeq} z(j, 2^m) \iff z(i2^m, 2^{n+m}) \dot{\succeq} z(j2^n, 2^{n+m}) \iff \frac{i}{2^n} \geq \frac{j}{2^m}. \tag{1.29}$$

Suppose that $\dot{p}(z(i, 2^n)) > {}^{i}\!/_{2^n}$. Then, for some $m \in \mathbb{W}$ and $0 \leq j \leq 2^m$, it would follow that $\dot{p}(z(k, 2^n)) \geq {}^{j}\!/_{2^m} > {}^{i}\!/_{2^n}$, whereupon (1.26) would imply $z(i, 2^n) \dot{\succeq} z(j, 2^m)$. Since this hypothesis directly contradicts (1.29), we conclude that $p(z(i, 2^n)) = {}^{i}\!/_{2^n}$.

Next, it follows by (1.26) and (1.29) that, for all $n \in \mathbb{W}$,

$$A \dot{\succeq} B \implies z(\kappa(A, 2^n), 2^n) \dot{\succeq} z(\kappa(B, 2^n), 2^n) \iff \kappa(A, 2^n) \geq \kappa(B, 2^n). \tag{1.30}$$

Consequently, the definition of $\dot{p}$ (1.27) implies

$$A \dot{\succeq} B \implies p(A) \geq p(B). \tag{1.31}$$

Finally, use F1–F5 to show that $\dot{p}$ is finitely additive and, then, obtain

$$A \dot{\succ} B \implies p(A) > p(B) \tag{1.32}$$

by refining (1.30). ■

We are now in roughly the same position as we were when starting the proof of Theorem 1.3. However, rather than assuming an exogenous probability measure, we have shown that the agent's preference $\succ$ on $\mathcal{F}$ give rise to a unique, subjective probability measure $\dot{p}$ on $\mathcal{A}$. We now turn our attention to the matter of expected utilities.

1.2.2 Proof of Theorem 1.5: Expected utility representation

This section demonstrates the latter half of Theorem 1.5, namely:

Theorem 1.5b. *P1–P7 imply that there exists a function $u : \mathcal{Y} \to \mathbb{R}$ satisfying*

$$f \succ g \iff \mathbb{E}_{\dot{p}}[u(f(\omega))] > \mathbb{E}_{\dot{p}}[u(g(\omega))], \text{ for all } f, g \in \mathcal{F}. \tag{1.33}$$

Per the previous section, let $u : \mathcal{Y} \to \mathbb{R}$ be the unique function for which

$$\gamma \sim u(\gamma)\gamma_{\text{sup}} + (1 - u(\gamma))\gamma_{\text{inf}}, \tag{1.34}$$

where $u(\gamma_{\text{sup}}) = 1$ and $u(\gamma_{\text{inf}}) = 0$. As discussed so far, then, u satisfies Theorem 1.5b on the set of all actions that generate simple outcome distributions, i.e.

$$\mathcal{F}_s = \{f \in \mathcal{F} : \dot{p} \circ f^{-1} \in \mathcal{P}_s\}. \tag{1.35}$$

To help move things along, we take for granted that u is a bounded, $\mathcal{A}$-measurable function and that $\dot{p} \circ f^{-1} = \dot{p} \circ g^{-1} \implies f \sim g$ for all $f, g \in \mathcal{F}_s$. Lastly, we will assume that $\gamma_{\text{sup}} \succeq f \succeq \gamma_{\text{inf}}$ for all actions $f \in \mathcal{F}$. Similar to the first part of the proof for Theorem 1.3, we require a few key lemmas to demonstrate Theorem 1.5b.

Lemma 1.9. *Let $\succ$ satisfy P1–P7 on $\mathcal{F}$. Then, the following statements hold for all actions $f \in \mathcal{F}$, non-empty events $A \subseteq \Omega$, outcomes $\mu \in \mathcal{Y}$, and constants $c \in \mathbb{R}$.*

a. *If simple distributions $p, q \in \mathcal{P}_s$ satisfy $p \succ q$, then*

$$p \succeq f \succeq q \implies f \sim \alpha p + (1-\alpha)q, \text{ for one and only one } 0 \leq \alpha \leq 1. \tag{1.36}$$

b. *There exists a simple distribution $p \in \mathcal{P}_s$ for which*

$$\mu \succeq f \mid A \text{ and } c > \sup_{\omega \in A} u(f(\omega)) \implies p \succeq f \mid A \text{ and } c \geq \mathbb{E}_p[u(\gamma)]. \tag{1.37}$$

c. *Given an n-event partition $\{A_1, \ldots, A_n\}$ of Ω and a simple distribution $p \in \mathcal{P}_s$ such that $f \succeq p$, it follows that*

$$\left[\exists c_i : c_i > \sup_{\omega \in A_i} u(f(\omega)),\ 1 \leq i \leq n\right] \implies \sum_{i=1}^{n} \dot{p}(A_i)c_i \geq \mathbb{E}_p[u(\gamma)]. \tag{1.38}$$

Proof These are lemmas 14.4, 14.6, and 14.7 from Fishburn (1970, Chapter 14). We sketch proofs of the first two parts. Starting with Lemma 1.9a: recall that $\mathcal{P}_s$ is a mixture set and suppose $f \sim \alpha p + (1 - \alpha)q$. It follows by Lemma 1.4 that, $\forall \beta \in [0, 1]$,

$$\beta > \alpha \implies \beta p + (1 - \beta)q \succ f \qquad \alpha > \beta \implies f \succ \beta p + (1 - \beta)q. \tag{1.39}$$

Consequently, there is at most one $\alpha \in [0, 1]$ that satisfies the claim. From here, repeated use of P6 to modify refinements of p and q yield the following contradictions:

$$\begin{aligned} f \succ \alpha p + (1 - \alpha)q &\implies f \succ \beta p + (1 - \beta)q, \text{ for } \beta = \alpha + \varepsilon; \\ \alpha p + (1 - \alpha)q \succ f &\implies \beta p + (1 - \beta)q \succ f, \text{ for } \beta = \alpha - \varepsilon, \end{aligned} \tag{1.40}$$

where $\varepsilon > 0$ is a small positive constant.[4] Hence, $f \sim \alpha p + (1-\alpha)q$.

Regarding Lemma 1.9b: if A is null or $c \geq u(\mu)$, then the claim is trivially satisfied by any p or δ_μ, respectively; henceforth, suppose otherwise. Since $c > u(f(\omega))$ for all $\omega \in A$, there exists an outcome λ so that $c \geq u(\lambda)$. Accordingly, let $p = \alpha\delta_\mu + (1-\alpha)\delta_\lambda$ be the unique mixture of δ_μ and δ_λ satisfying $\mathbb{E}_p[u(\gamma)] = c$.

For an arbitrary choice of $\omega_i \in A$, let $f_i = \mathbf{x}_A(p, f(\omega_i))$ be the compound action that yields a random outcome distributed according to p when $\omega \in A$ and $f(\omega_i)$ otherwise. Further, denote $p_i = p \circ f_i^{-1}$. Since $\mathbb{E}_p[u(\gamma)] = c > \sup_{\omega \in A} u(f(\omega))$, we have

$$\begin{aligned} u(f(\omega_i)) &= \dot{p}(A)u(f(\omega_i)) + (1-\dot{p}(A))u(f(\omega_i)) \\ &< \dot{p}(A)\,\mathbb{E}_p[u(\gamma)] + (1-\dot{p}(A))u(f(\omega_i)) = \mathbb{E}_{p_i}[u(\gamma)]. \end{aligned} \tag{1.41}$$

Consequently, Theorem 1.3 implies $f_i \succ f(\omega_i)$. By construction, then, $f_i \succ f(\omega_i) \mid A$. Finally, since $f_i \succeq f(\omega_i) \mid A$ and $p_i = p$ on A for all $\omega_i \in A$, P7 implies $p \succeq f \mid A$. ■

These lemmas in hand, we are ready to prove the second half of Theorem 1.5. The general strategy will be to show that there exists a $p \in \mathcal{P}_s$ satisfying $p \sim f$. Doing so will enable us to translate statements such as $f \succeq g$ into the language of preferences over simple distributions (e.g. $p \succeq p'$) and so recycle Theorem 1.3.

Theorem 1.5b. *P1–P7 imply that there exists a function $u : \mathcal{Y} \to \mathbb{R}$ such that*

$$f \succ g \iff \mathbb{E}_{\dot{p}}[u(f(\omega))] > \mathbb{E}_{\dot{p}}[u(g(\omega))], \text{ for all } f, g \in \mathcal{F}. \tag{1.42}$$

Proof Due to earlier assumption that $\gamma_{\sup} \succeq f \succeq \gamma_{\inf}$, Lemma 1.9a implies that there is a simple distribution $p \in \mathcal{P}_s$ for which $p \sim f$. Now, consider the n-event partition of Ω with elements

$$\Omega_i^n = \left\{\omega : \tfrac{i-1}{n} < u(f(\omega)) \leq \tfrac{i}{n}\right\}, \text{ for } i = 1, \ldots, n. \tag{1.43}$$

On this partition, define the function

$$u_n(\omega) = \tfrac{i-1}{n} \text{ for all } \omega \in \Omega_i^n,\ i = 1, \ldots, n, \tag{1.44}$$

such that

$$\mathbb{E}_{\dot{p}}[u(f(\omega))] \geq \mathbb{E}_{\dot{p}}[u_n(f(\omega))] = \sum_{i=1}^{n} \dot{p}(\Omega_i^n)\tfrac{i-1}{n}. \tag{1.45}$$

Here, n denotes the cardinality of the partition, not the number of distinct outcomes with support under p. Consequently, some Ω_i^n may be empty. Since u is bounded and $\mathcal{A}$-measurable, it follows that u_n converges uniformly from below to u as $n \to \infty$. In much the same way, define

$$u_n'(\omega) = \tfrac{i-1-\varepsilon}{n} \text{ for all } \omega \in \Omega_i^n,\ i = 1, \ldots, n, \tag{1.46}$$

[4]Note that $q \succ f$ and $f \succ \alpha p + (1-\alpha)q$ imply that $\alpha < 1$, so $\alpha + \varepsilon$ is valid; and so for $\alpha - \varepsilon$.

where $\varepsilon > 0$. By Lemma 1.9c it follows that

$$\mathbb{E}_p[u(\gamma)] \geq \mathbb{E}_{\dot{p}}[u'_n(f(\omega))] = \sum_{i=1}^{n} \dot{p}(\Omega_i^n) \tfrac{i-1-\varepsilon}{n}. \tag{1.47}$$

Since u_n and u'_n both converge to u as $n \to \infty$, the definition of expectation implies

$$\mathbb{E}_p[u(f(\omega))] = \lim_{n\to\infty} \mathbb{E}_p[u_n(f(\omega))] = \lim_{n\to\infty} \mathbb{E}_p[u'_n(f(\omega))] = \mathbb{E}_q[u(\gamma)]. \tag{1.48}$$

Analogous to Section 1.1.2, we may then show that

$$f \sim p \in \mathcal{P}_s \iff \mathbb{E}_p[u(f(\omega))] = \mathbb{E}_p[u(\gamma)]. \tag{1.49}$$

Finally, it follows by transitivity that

$$f \succ g \iff \mathbb{E}_{\dot{p}}[u(f(\omega))] > \mathbb{E}_{\dot{p}}[u(g(\omega))], \text{ for all } f, g \in \mathcal{F}. \tag{1.50}$$

Hence, the claim follows. ■

1.3 Bayesian decision theory

Having shown how basic assumptions regarding preferences can be used to construct a framework for rational decision-making, let us now consider the case of a stateful agent whose preferences change as new information arrives. This will be important in later sections when considering sequential decision-making problems. The purpose of this section is to show that conditionally most-preferred actions maximize conditional expected utility functions.

Originally given by (1.19), the conditional preference relation is defined as

$$f \succ g \mid A \iff \mathbf{x}_A(f,h) \succ \mathbf{x}_A(g,h), \text{ for all } h \in \mathcal{F}, \tag{1.51}$$

where, e.g., $\mathbf{x}_A(f,h)$ is a compound action such that f is carried out if $\theta \in A$ else h. Recall that, by P2, $f \succ g \mid A$ may be interpreted to mean that f is preferred to g when the set of possible states is restricted to $A \subseteq \Omega$. Below, we will write $S \subseteq \Omega$ for the agent's knowledge state and assume it to be a "$\succ$"-nonnull event such that $\dot{p}(S) > 0$. Similarly, define the conditional more-likely-than relation as

$$A \dot{\succ} B \mid S \iff \mathbf{x}_A(\mu,\lambda) \succ \mathbf{x}_B(\mu,\lambda) \mid S, \quad \forall \mu, \lambda \in \mathcal{Y} \text{ such that } \mu \succ \lambda. \tag{1.52}$$

In words: A is conditionally more likely than B if, upon restricting Ω to S, the agent prefers a binary lottery predicated on A to its equivalent lottery predicated on B. This is definition has the benefit of being constructive, but does not necessarily lend itself to intuition. Fortunately, refining (1.52) into a more palatable form proves to be straightforward.

Using (1.51), rewrite the right-hand side of (1.52) in terms of doubly compounded actions with $h = g$ and simplify as

$$\begin{aligned} A \dot{\succ} B \mid S &\iff \mathbf{x}_S(\mathbf{x}_A(f,g),g) \succ \mathbf{x}_S(\mathbf{x}_B(f,g),g) \\ &\iff \mathbf{x}_{A\cap S}(f,g) \succ \mathbf{x}_{B\cap S}(f,g) \\ &\iff (A\cap S) \dot{\succ} (B \cap S). \end{aligned} \tag{1.53}$$

Hence, the more-likely-than relation and its conditional counterpart have the familiar property that an event A is more likely than another B given a third event S if and only if S is more likely to jointly manifest with A than with B.

Translating (1.51) in terms of the (unconditional) more-likely-than relation $\dot{\succ}$ allows us to inherit results discussed early in the text. In particular, the definition of qualitative probability (Definition 1.8) implies that

$$A \dot{\succ} B \mid S \iff \dot{p}(A\cap S) > \dot{p}(B \cap S). \tag{1.54}$$

This implication in mind, define a probability measure $\dot{p}(\ \cdot\ \mid S)$ as[5]

$$\dot{p}(A \mid S) = \frac{\dot{p}(A\cap S)}{\dot{p}(S)}, \text{ for all } A \subseteq \Omega. \tag{1.55}$$

Since $\dot{p}(S) > 0$ by assumption, $\dot{p}(\ \cdot\ \mid S)$ satisfies the first part of Theorem 1.5, namely

$$A \dot{\succ} B \mid S \iff \dot{p}(A \mid S) > \dot{p}(B \mid S), \text{ for all } A, B \subseteq \Omega. \tag{1.56}$$

More generally, one may use the fact that the conditional preference-indifference relation obeys P1–P7 to show that the conditional more-likely-than relation admits F1–F5 and, hence, that $\dot{p}(\ \cdot\ \mid S)$ is the only probability measure which satisfies Theorem 1.5a. For more details, see Kreps (1988, Chapter 10).

Turning our attention to the second half of Theorem 1.5, let us show that conditional preferences indeed correspond with conditional expected utilities. To help ease notation, let $f' = \mathbf{x}_S(f,h)$ and $g' = \mathbf{x}_S(g,h)$ so that $f \succ g \mid S \iff f' \succ g'$ for an arbitrary choice of $h \in \mathcal{F}$. Theorem 1.5 implies that

$$\begin{aligned} f \succ g \mid S &\iff f' \succ g' \iff \mathbb{E}_{\dot{p}}[u(f'(\omega))] > \mathbb{E}_{\dot{p}}[u(g'(\omega))] \\ &\iff \int_{\mathcal{Y}} u(\gamma)\dot{p}(f'^{-1}(\gamma))d\gamma > \int_{\mathcal{Y}} u(\gamma)\dot{p}(g'^{-1}(\gamma))d\gamma. \end{aligned} \tag{1.57}$$

By finite additivity of $\dot{p}$, we have

$$\dot{p}(A) = \dot{p}(A\cap S) + \dot{p}(A \cap S^c), \text{ for any } A \subseteq \Omega. \tag{1.58}$$

Moreover, since $f' = g'$ on S^c, it follows that

$$\dot{p}(f'^{-1}(\gamma) \cap S^c) = \dot{p}(g'^{-1}(\gamma) \cap S^c), \text{ for all } \gamma \in \mathcal{Y}. \tag{1.59}$$

[5]This is not a rigorous definition of conditional probability, but suffices when $\dot{p}$ admits representation in terms of a probability density (mass) function.

Decomposing both $\dot{p}(f'^{-1}(\gamma))$ and $\dot{p}(g'^{-1}(\gamma))$ on the right-hand side of (1.57) with the help of (1.58) and, subsequently, eliminating like terms gives

$$\begin{aligned} f \succ g \mid S &\iff \int_{\mathcal{Y}} u(\gamma)\dot{p}(f'^{-1}(\gamma) \cap S)d\gamma > \int_{\mathcal{Y}} u(\gamma)\dot{p}(g'^{-1}(\gamma) \cap S)d\gamma \\ &\iff \int_{\mathcal{Y}} u(\gamma)\dot{p}(f^{-1}(\gamma) \cap S)d\gamma > \int_{\mathcal{Y}} u(\gamma)\dot{p}(g^{-1}(\gamma) \cap S)d\gamma, \end{aligned} \tag{1.60}$$

where, in the second line, we have used the fact that $f' = f$ and $g' = g$ on S. Finally, dividing through by $\dot{p}(S)$ confirms that conditional preferences obey Theorem 1.5b:

$$f \succ g \mid S \iff \mathbb{E}_{\dot{p}(\omega|S)}[u(f(\omega))] > \mathbb{E}_{\dot{p}(\omega|S)}[u(g(\omega))]. \tag{1.61}$$

Under Savage's model, then, a rational agent incorporates information about the world by updating their beliefs about the probabilities of different events in accordance with Bayes' rule. Putting things together gives the following definition of optimal decision-making, the implications of which will be discussed below.

Definition 1.10 (Bayes optimal strategy)**.** *Let $\mathcal{F}$ be the set of actions mapping from states Ω to outcomes $\mathcal{Y}$ and let $\mathcal{A}$ be the largest σ-algebra for Ω. Supposing the agent's preference $\succ$ on $\mathcal{F}$ obeys P1–P7, a strategy $\pi : \mathcal{A} \to \mathcal{F}$ is Bayes optimal if and only if, for all "$\succ$"-nonnull events $S \in \mathcal{A}$, it follows that*

$$\pi(S) \in \arg\max_{f \in \mathcal{F}} \mathbb{E}_{\dot{p}(\omega|S)}[u(f(\omega))], \tag{1.62}$$

where $\dot{p}(\ \cdot\ \mid S)$ is the unique, finitely additive probability measure on $\mathcal{A}$ that agrees with the conditional more-likely-than relation (1.52) *induced by $\succ$ in the sense that*

$$A \dot{\succ} B \mid S \iff \dot{p}(A \mid S) > \dot{p}(B \mid S), \text{ for all } A, B \in \mathcal{A}; \tag{1.63}$$

and, where u is the unique utility function (up to a positive affine transform) satisfying

$$f \succ g \mid S \iff \mathbb{E}_{\dot{p}(\omega|S)}[u(f(\omega))] > \mathbb{E}_{\dot{p}(\omega|S)}[u(g(\omega))], \text{ for all } f, g \in \mathcal{F}. \tag{1.64}$$

1.4 Discussion

This chapter surveyed the foundations of Bayesian decision theory with an emphasis on the key technical arguments behind its development. Throughout, our primary aim has been to show why a rational agent (as prescribed by Savage's seven axioms) should maximize the conditional expected utility of their actions.

This framework serves as the basis for popular algorithms that seek to enact or emulate Bayes optimal strategies (Definition 1.10). Often, the agent's utility function is explicitly given in the form of an objective or reward signal, which greatly simplifies the decision-making process. In these cases, one typically constructs a model that encodes the agent's prior beliefs and proceeds by selecting actions that maximize the

given utility function's conditional expectation under the model. Algorithms of this sort are easy to understand, but can be difficult to execute.

Throughout the remainder of this work, we examine these algorithms and the hurdles they face. Chapter 2 focuses on Bayesian optimization. In it, we first show how ideas discussed in the present chapter can be used to derive well-known *acquisition functions* from a decision theoretic perspective. We then study the problem of maximizing acquisition functions, which corresponds to the process of finding optimal actions. We investigate the mathematical properties of these acquisition functions (and their estimators) and uses these to devise efficient maximization procedures.

In Chapter 3, we then restrict our attention to a particular class of models known as Gaussian processes. These models have a number of desirable properties — such as being highly interpretable and making well-calibrated predictions — that make them the go-to choice for many decision-making algorithms. Unfortunately, Gaussian processes have typically struggled in cases where estimating an action's expected utility requires us to simulate a large numbers of terms under the model. Seeking to address this issue, we will show how these simulations can be cheaply generated by taking advantage of a lesser-known formulation of Gaussian process posteriors.

CHAPTER 2

Bayesian Optimization

In this chapter, we investigate the application of Bayesian decision theory to global optimization problems

$$\boldsymbol{x}^* \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}, \omega^*), \tag{2.1}$$

where $f : \mathcal{X} \times \Omega \to \mathbb{R}$ is an objective function mapping designs $\boldsymbol{x} \in \mathcal{X}$ and states $\omega \in \Omega$ to outcomes $y \in \mathbb{R}$, while $\omega^* \in \Omega$ is an unknown ground-truth state.

We focus on the "black-box function" setting, where f is seen as a stochastic process and ω^* as an abstract random number. For simplicity, let us assume that f is an infinite collection of continuous random variables $y = f(\boldsymbol{x})$, $\forall \boldsymbol{x} \in \mathcal{X}$, whose probability measure uniquely extends a family of (consistent) distributions over its finite-dimensional subsets $\mathrm{Fin}(f) = \{f(\mathbf{X}_n) : \forall \mathbf{X}_n \in \mathcal{X}^n, \forall n \in \mathbb{N}\}$.[1]

Since the elements of $\mathrm{Fin}(f)$ are assumed continuous, it follows that there exist probability density functions that agree with each of the aforementioned finite-dimensional distributions in the usual way. Moreover, for any two finite collections $\boldsymbol{y}, \boldsymbol{y}' \in \mathrm{Fin}(f)$, the random variable $\boldsymbol{y} \mid \boldsymbol{y}' = \boldsymbol{\gamma}'$ will follow the conditional distribution $p(\boldsymbol{y} \mid \boldsymbol{\gamma}') = p(\boldsymbol{y}, \boldsymbol{\gamma}')p(\boldsymbol{\gamma}')^{-1}$.

The general structure of this chapter is as follows. We first derive a Bayesian decision-theoretic approach to optimization, during the course of which a number of key definitions will be made precise. We, then, take a step back and discuss Bayesian optimization in a broader sense before focusing in on techniques for efficiently maximizing acquisition functions.

[1] Where possible without introducing ambiguity, we denote the pointwise evaluation of a function on a set as, e.g., $f(\mathbf{X}_n) = \{f(\boldsymbol{x}) : \forall \boldsymbol{x} \in \mathbf{X}_n\}$.

2.1 The Bayesian approach to optimization

Continuing from the previous chapter, we begin by exploring the strategy suggested by Savage's model of decision-making. To this end, we interpret the aforementioned global optimization problem as a statement about preference. Specifically, we interpret (2.1) to mean that the agent's utility function $u : \mathbb{R} \to \mathbb{R}$ is identity so that, for all $\omega \in \Omega$ and $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$,

$$\boldsymbol{x} \succeq \boldsymbol{x}' \mid \omega \iff f(\boldsymbol{x}, \omega) \geq f(\boldsymbol{x}', \omega), \tag{2.2}$$

where conditional preference $\boldsymbol{x} \succeq \boldsymbol{x}' \mid \omega$ is defined as in (1.19). Recasting the original problem in this way will enable us to naturally accommodate uncertainty for ω^*. In most cases, however, it will be more convenient for us to express this uncertainty in terms of conditional distributions on $\mathrm{Fin}(f)$ given observations $D_t = (\boldsymbol{x}_i, \gamma_i)_{i=1}^t$, where $\gamma_i = f(\boldsymbol{x}_i, \omega^*)$ is the ground-truth value of f at $\boldsymbol{x}_i \in \mathcal{X}$.

Returning to the optimization problem (2.1), we know from Section 1.3 that any most-preferred design — henceforth referred to as an *incumbent* and denoted by $\boldsymbol{\chi} \in \mathcal{X}$ — must maximize the agent's conditional expected utility, i.e.

$$\boldsymbol{\chi} \in \operatorname*{arg\,max}_{\boldsymbol{x}_{t+1} \in \mathcal{X}} \mathbb{E}_{p(y_{t+1} \mid D_t)}[y_{t+1}]. \tag{2.3}$$

As such, define an *incumbent rule* $\chi : \mathcal{D} \to \mathcal{X}$ as any function mapping datasets in $\mathcal{D} = \mathrm{Fin}(\mathcal{X} \times \mathbb{R})$ to conditionally most-preferred designs so that

$$\chi(D) \succeq \boldsymbol{x} \mid D \text{ for all } \boldsymbol{x} \in \mathcal{X} \text{ and } D \in \mathcal{D}. \tag{2.4}$$

Denoting $\boldsymbol{\chi}_t = \chi(D_t)$, we will write

$$U(D_t) = \mathbb{E}_{p(f(\boldsymbol{\chi}_t) \mid D_t)}[f(\boldsymbol{\chi}_t)] \tag{2.5}$$

for the conditional expected utility of the incumbent under rule χ. The remainder of this section focuses on the most widely used incumbent rule, namely the *best-seen rule*

$$\chi_{\text{BS}}(D_t) \in \{\boldsymbol{x}_i \in \mathcal{X} : \exists \gamma_i \in \mathbb{R}, (\boldsymbol{x}_i, \gamma_i) \in D_t \wedge \gamma_i = \gamma_t^*\} \tag{2.6}$$

where $\boldsymbol{\gamma}_t = (\gamma_1, \ldots, \gamma_t)$ is the set of observed outcomes and $\gamma_t^* = \max \boldsymbol{\gamma}_t$. Since $p(y \mid D_t) = \delta_\gamma$ for all observed designs $\boldsymbol{x} \in (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t)$, it follows that

$$U_{\text{BS}}(D_t) = \max_{i=1,\ldots,t} \mathbb{E}[f(\boldsymbol{x}_i) \mid D_t] = \gamma_t^*. \tag{2.7}$$

Hence, the best-seen rule is Bayes-optimal (Definition 1.10) when incumbents must be chosen from the set of previously queried designs. Below, we extend this line of reasoning to derive Bayes-optimal strategies for querying f.

Consider the simple case where an agent is given a dataset D_{T-1} and allowed to evaluate single design $\boldsymbol{x}_T \in \mathcal{X}$ before nominating a final incumbent $\boldsymbol{\chi}_T = \chi_{\text{BS}}(D_T)$.

Since (2.7) implies the agent's utility is unchanged when re-evaluating a previously queried design, we focus on the case of novel queries $\boldsymbol{x}_T \notin (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T-1})$. In this setting, uncertainty over outcomes $y_T = f(\boldsymbol{x}_T)$ typically leads to uncertainty for the choice of incumbent $\boldsymbol{\chi}_T$. Consequently, queries $\boldsymbol{x}_T$ can be seen as compound actions (see Section 1.1.2) that stipulate a choice of $\boldsymbol{\chi}_T$ for each possible value of $y_T = f(\boldsymbol{x}_T)$. For the best-seen rule (2.6), this *incumbent plan* is defined as

$$\chi_{\text{BS}}(\boldsymbol{x} \mid D_{T-1}) = \begin{cases} \boldsymbol{x} & f(\boldsymbol{x}) > \gamma^*_{T-1} \\ \boldsymbol{\chi}_{T-1} & \text{otherwise,} \end{cases} \tag{2.8}$$

and its expected utility is given by

$$\begin{aligned} U_{\text{BS}}(\boldsymbol{x}_T \mid D_{T-1}) &= U_{\text{BS}}(D_{T-1}) + \int_{\gamma^*_{T-1}}^{\infty} \Big(y_T - \gamma^*_{T-1}\Big) p(y_T \mid D_{T-1}) \, \mathrm{d}y_T \\ &= \underbrace{U_{\text{BS}}(D_{T-1})}_{\text{utility at } t} + \underbrace{\mathbb{E}_{p(y_T \mid D_{T-1})}\Big[\max\Big\{0, y_T - \gamma^*_{T-1}\Big\}\Big]}_{\text{expected improvement at } t+1}. \end{aligned} \tag{2.9}$$

These equations are simple and intuitive: (2.8) tells us that the agent intends to keep incumbent $\boldsymbol{\chi}_{T-1}$ unless query $\boldsymbol{x}_T$ outperforms it (and so takes its place); similarly, (2.9) says that the expected utility for querying $\boldsymbol{x}_T$ is simply the agent's current expect utility incremented by the *expected improvement* of $\boldsymbol{x}_T$ over $\boldsymbol{\chi}_{T-1}$ (more on this later).

Under the best-seen rule, querying by maximizing (2.9) is *one-step optimal* in the sense that there is no design whose evaluation would generate a more favorable incumbent plan. More formally,

$$\boldsymbol{x} \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} U_{\text{BS}}(\boldsymbol{x} \mid D_t) \iff \chi_{\text{BS}}(\boldsymbol{x} \mid D_t) \succeq \chi_{\text{BS}}(\boldsymbol{x}' \mid D_t) \mid D_t, \ \forall \boldsymbol{x}' \in \mathcal{X}. \tag{2.10}$$

To see this more clearly, recall $U_{\text{BS}}(D_{T-1}) = \max\{\gamma_1, \ldots, \gamma_{T-1}\} = \gamma^*_{T-1}$ and write

$$\begin{aligned} U_{\text{BS}}(\boldsymbol{x}_T \mid D_{T-1}) &= \mathbb{E}_{p(y_T \mid D_{T-1})}[\max\{\gamma_1, \ldots, \gamma_{T-1}, y_T\}] \\ &= \mathbb{E}_{p(y_T \mid D_{T-1})}[U_{\text{BS}}(D_{T-1} \cup (\boldsymbol{x}_T, y_T))]. \end{aligned} \tag{2.11}$$

Now, suppose the agent is tasked with selecting a penultimate query $\boldsymbol{x}_{T-1} \in \mathcal{X}$, again with the goal of maximizing the expected utility of an incumbent chosen at time T. We have already that one-step optimal strategies proceed by maximizing the expected utility of incumbent plans that peer one step into the future. By backward induction, it follows that the expected utility of penultimate query $\boldsymbol{x}_{T-1}$ is given by

$$U_{\text{2-step}}(\boldsymbol{x}_{T-1} \mid D_{T-2}) = \mathbb{E}_{p(y_{T-1} \mid D_{T-2})}\Big[\max_{\boldsymbol{x}_T \in \mathcal{X}} U\Big(\boldsymbol{x}_T \mid D_{T-2} \cup (\boldsymbol{x}_{T-1}, y_{T-1})\Big)\Big] \tag{2.12}$$

and that $\boldsymbol{x}_{T-1} \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} U_{\text{2-step}}(\boldsymbol{x} \mid D_{T-2})$ is a Bayes optimal query. Along the same lines, the expected utility of a query made $\tau = T - t$ steps prior to terminating

may be written as

$$U_{\tau\text{-step}}(\boldsymbol{x}_t \mid D_t) = \mathbb{E}\Big[\max_{\boldsymbol{x}_{t+1}\in\mathcal{X}} \ldots \mathbb{E}\Big[\max_{\boldsymbol{x}_{T-1}\in\mathcal{X}} \mathbb{E}\Big[\max_{\boldsymbol{x}_T\in\mathcal{X}} U\Big(\boldsymbol{x}_T \mid D_t \cup (\boldsymbol{x}_i, y_i)_{i=t}^{T-1}\Big)\Big]\Big]\Big]. \quad (2.13)$$

Note that, in the preceding equations, we have assumed that maximums exist[2] and that lookahead utilities are measurable.

In a narrow capacity, we have now "solved" the problem of Bayesian optimization in that we have obtained a family of Bayes optimal strategies for querying f. To see how this approach falls short, consider the simple case where each of $m = |\mathcal{X}|$ available queries has $n = |\mathcal{Y}|$ potential outcomes for some $m, n \in \mathbb{N}$. At step $T-1$, an optimal query is found by solving

$$\boldsymbol{x}_{T-1} \in \arg\max_{\boldsymbol{x}\in\mathcal{X}} \sum_{\gamma\in\mathcal{Y}} p(\gamma \mid D_{T-2}) \max_{\boldsymbol{x}'\in\mathcal{X}} U(\boldsymbol{x}' \mid D_{T-2} \cup (\boldsymbol{x}, \gamma)). \quad (2.14)$$

Since the term being summed on the right has $\mathcal{O}(mn)$ time complexity, it would then take us $\mathcal{O}(m^2n^2)$ time to obtain $\boldsymbol{x}_{T-1}$. For longer planning horizons, this trend continues such that Bayes optimal querying quickly becomes prohibitively expensive. This tension between theory and practice is at the heart of this chapter and, arguably, of Bayesian optimization as a field. In the next section, we will explore a variety of acquisition functions and see how they balance these agendas.

2.2 Acquisition functions

So far, we have pursued Bayesian optimization (BO) from a formal perspective as a natural extension of Bayesian decision theory. In this section, we will broaden this definition and discuss BO in the sense of optimization strategies based on conditional expectations of *value functions* $v : \mathcal{D} \to \mathbb{R}$, where $\mathcal{D} = \mathrm{Fin}(\mathcal{X} \times \mathbb{R})$. Specifically, we will focus on algorithms that given an arbitrary dataset $D \in \mathcal{D}$ select a set of queries by maximizing an acquisition function

$$V(\mathbf{X} \mid D) = \mathbb{E}_{p(\boldsymbol{y}|D)}[v(\mathbf{X}, \boldsymbol{y} \mid D)], \quad (2.15)$$

where $v(\mathbf{X}, \boldsymbol{y} \mid D)$ can loosely be seen as the extent to which observing $\boldsymbol{y} = f(\mathbf{X})$ helps the agent identify an optimal design. We will say that an acquisition function is *myopic* if the value it assigns to a set of queries is solely determined by the joint distribution of the corresponding outcomes, i.e. if it holds for all $D \in \mathcal{D}$ and $\mathbf{X}, \mathbf{X}' \subseteq \mathcal{X}$ that

$$p(f(\mathbf{X}) \mid D) = p(f(\mathbf{X}') \mid D) \implies V(\mathbf{X} \mid D) = V(\mathbf{X}' \mid D). \quad (2.16)$$

In these cases, we will omit $\mathbf{X}$ from the right-hand side and write $V(\mathbf{X} \mid D) = \mathbb{E}_{p(\boldsymbol{y}|D)}[v(\boldsymbol{y} \mid D)]$. Notice how this definition implies the value of observing $\boldsymbol{y} = f(\mathbf{X})$ does not reflect said observations' impact on the expected utility of designs $\mathcal{X} \setminus \mathbf{X}$.

[2]This will be the case, e.g., when $\mathcal{X}$ is compact and f is sample-continuous.

Abbr.	**Score Function**	**Reparameterization**	**Myopic**
EI	$\max\{0, \boldsymbol{y} - \alpha\}$	$\max\Big\{0, \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z} - \alpha\Big\}$	Y
EMAX	$\max\{\boldsymbol{y}\}$	$\max\Big\{\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}\Big\}$	Y
KG	$\sup\Big\{\mu \mid \boldsymbol{y}\Big\} - \sup \mu$	$\sup\Big\{\mu + k(\,\cdot\,, \mathbf{X})\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\Big\} - \sup \mu$	N
UCB	$\max\Big\{\boldsymbol{\mu} + \sqrt{\beta\pi/2}\,\mathrm{Abs}(\boldsymbol{y} - \boldsymbol{\mu})\Big\}$	$\max\Big\{\boldsymbol{\mu} + \sqrt{\beta\pi/2}\,\mathrm{Abs}\Big(\boldsymbol{\Sigma}^{1/2}\boldsymbol{z}\Big)\Big\}$	Y
PI	$\max\Big\{\mathbb{1}_{y_1>\alpha}, \ldots, \mathbb{1}_{y_q>\alpha}\Big\}$	$\max\Big\{\sigma_{\tau,\epsilon}\Big(\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{z}\Big)\Big\}$	Y

Table 2.1: Overview of acquisitions functions discussed in this chapter, each of which is obtained by integrating out $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ from the second column or $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ from the third. *Glossary:* query locations $\mathbf{X} \in \mathcal{X}^q$, outcomes $\boldsymbol{y} = f(\mathbf{X})$, improvement threshold α, mean function $\mu : \mathcal{X} \to \mathbb{R}$ and vector $\boldsymbol{\mu} = \mu(\mathbf{X})$, covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and matrix $\boldsymbol{\Sigma} = k(\mathbf{X}, \mathbf{X})$, confidence parameter $\beta \in \mathbb{R}_+$, absolute value function $\mathrm{Abs} : \mathbb{R} \to \mathbb{R}_+$, and (inverse) sigmoid function $\sigma_{\tau,\epsilon}(y)^{-1} = 1 + \exp(-\frac{y+\epsilon}{\tau})$ with temperature parameter $\tau \in \mathbb{R}_+$ and offset $\epsilon \in \mathbb{R}_+$ so that $\sigma_{\tau,\epsilon}(y)$ converges to $\mathbb{1}_{y>0}$ as τ and ϵ both tend to zero.

Figure 2.1 sketches a prototypical Bayesian optimization algorithm. At each iteration, a model is fit to a set of observations and an acquisition function is defined in terms of the resulting predictive posterior. A set of queries is obtained by maximizing this acquisition function; new observations are made by evaluating these queries; and the cycle repeats. In the remainder of this section, our goal will be to see how the machinery developed in the previous section translates to well-known acquisition functions.

Towards this end, we define the *marginal expected utility* of a query $\boldsymbol{x}$ given dataset D as the difference in expected utilities of incumbents chosen before and after observing $y = f(\boldsymbol{x})$. Various acquisition functions manifest as marginal expected utilities; the most popular of which is undoubtedly the Expected Improvement acquisition function (Jones et al., 1998). In the previous section, Expected Improvement (EI) was implicitly given as the (one-step) marginal expected utility of a query $\boldsymbol{x}$ under the best-seen incumbent rule:

$$\mathrm{EI}(\boldsymbol{x} \mid D) = U_{\mathrm{BS}}(\boldsymbol{x} \mid D) - U_{\mathrm{BS}}(D) = \mathbb{E}_{p(y|D)}[\max\{0, y - \alpha\}], \tag{2.17}$$

where U_{bs} is defined in (2.9) and where $\alpha = \max\{\gamma \in \mathcal{Y} : \exists \boldsymbol{x} \in \mathcal{X}, (\boldsymbol{x}, \gamma) \in D\}$ denotes the "best-seen" outcome prior to querying $\boldsymbol{x}$. Due to their clear theoretic origins, ease-of-use, and strong empirical performance, EI and its many variants are often regarded as go-to choices of acquisition functions. Table 2.1 overviews the various acquisition functions discussed throughout this chapter. Prior to discussing these alternatives however, we briefly detour to introduce batch querying strategies. For brevity, we focus on the fully synchronous case.

In many real-world scenarios, the agent would like to simultaneously evaluate $q > 1$ queries. We will not dwell on their motives for doing so; however, it is worth noting that purely sequential querying strategies dominate parallel ones in the absence of factors such as time constraints and shared costs. When it comes to incumbent-based

acquisition functions, there is nothing particularly obscure about the definition of the marginal expected utility for querying a batch of designs $\mathbf{X} \in \mathcal{X}^q$, given here as

$$V(\mathbf{X} \mid D) = U(\mathbf{X} \mid D) - U(D). \tag{2.18}$$

The term $U(D)$ averages the utility of the incumbent $\chi(D) \in \mathcal{X}$ chosen by the rule χ; and, $U(\mathbf{X} \mid D)$ does the same, while further accounting for the impact observing $\boldsymbol{y} = f(\mathbf{X})$ will have on the resulting incumbent. As a concrete example, the Expected Improvement of a batch $\mathbf{X}$ may be written as

$$\mathrm{EI}(\mathbf{X} \mid D) = \mathbb{E}_{p(\boldsymbol{y}|D)}[\max\{0, y_1 - \alpha, \ldots, y_q - \alpha\}] \tag{2.19}$$

and corresponds to the increase in the incumbent's expected utility under the plan

$$\chi_{\text{BS}}(\mathbf{X} \mid D) = \begin{cases} \boldsymbol{x}_1 & y_1 > \max\{\alpha, y_2, \ldots, y_q\}, \\ \vdots & \\ \boldsymbol{x}_q & y_q > \max\{\alpha, y_1, \ldots, y_{q-1}\}, \\ \chi_{\text{BS}}(D) & \text{otherwise}, \end{cases} \tag{2.20}$$

where $\alpha = \max\{\gamma \in \mathcal{Y} : \exists \boldsymbol{x} \in \mathcal{X}, (\boldsymbol{x}, \gamma) \in D\}$ again denotes the utility of the current, best-seen incumbent. Parallel querying and, specifically, maximization of batch acquisition functions plays a major role in the second half of this chapter. For this reason, we focus on acquisition functions in their generalized batch forms henceforth.

Moving on from Expected Improvement, any number of acquisition functions may be defined by designating different incumbent rules. Strategies based on maximizing these acquisition functions all share the property of being (one-step) optimal in some capacity. Two acquisition functions in particular may be obtained by minor modification of the best-seen rule.

The first is found by taking the best-seen rule and further restricting the set of valid incumbent according to how recently a design was queried. For an arbitrary choice of window $\tau \geq 0$, this gives the *best-recent rule*

$$\chi_{\text{BR}}(D_t) = \begin{cases} \boldsymbol{x}_1 & t_1 \geq t - \tau, \text{and } y_1 \geq \max\{y_2, \ldots, y_q\}, \\ \vdots & \\ \boldsymbol{x}_q & t_q \geq t - \tau \text{ and } y_q > \max\{y_1, \ldots, y_{q-1}\}, \end{cases} \tag{2.21}$$

where, by minor abuse of notation, t_i denotes the arrival time of the i-th outcome. When $\tau = 0$, we recover the EMAX acquisition function (Azimi et al., 2010)

$$\mathrm{EMAX}(\mathbf{X} \mid D_t) = \mathbb{E}_{p(\boldsymbol{y}|D_t)}[\max \boldsymbol{y}], \tag{2.22}$$

which can be viewed as selecting an incumbent from the most recent batch $\mathbf{X}$. Especially when $q = 1$ (or when batches are constructed greedily as discussed in Section 2.5), maximizing EMAX often results in repeated queries of the best-seen design $\boldsymbol{\chi} = \chi_{\text{BS}}(D)$. In these cases, it follows that the optimal size q batch under

EMAX is no better than the optimal size $q-1$ batch under EI

$$\mathrm{EMAX}(\mathbf{X} \mid D) = \mathrm{EI}(\mathbf{X} \setminus \boldsymbol{\chi} \mid D) + \alpha = U_{\mathrm{BS}}(\mathbf{X} \setminus \boldsymbol{\chi} \mid D). \tag{2.23}$$

When all previous queries are valid incumbents, we therefore prefer EI to EMAX.

Rather than restricting incumbents to the set of previously evaluated designs (or some subset thereof), suppose we allow them to be chosen at will on the entire domain $\mathcal{X}$. Generalizing the best-seen rule in this way gives the *best-expected rule*

$$\chi_{\mathrm{BE}}(D) \in \arg\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}\Big[f(\boldsymbol{x}) \mid D\Big] \tag{2.24}$$

and, similarly, the best-expected plan

$$\chi_{\mathrm{BE}}\Big(\mathbf{X} \mid D_t\Big) = \chi_{\mathrm{BE}}\Big(D_t \cup (\boldsymbol{x}_j, \gamma_j)_{j=1}^q\Big), \text{ for all } \boldsymbol{\gamma} \in \mathcal{Y}^q. \tag{2.25}$$

Together, they give the Knowledge Gradient acquisition function (Gupta and Miescke, 1996; Frazier et al., 2008)

$$\mathrm{KG}(\mathbf{X} \mid D) = \mathbb{E}_{p(\boldsymbol{y}|D)}\Big[\max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}\Big[f(\boldsymbol{x}) \mid D \cup (\boldsymbol{x}_j, y_j)_{j=1}^q\Big]\Big] - \max_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}\Big[f(\boldsymbol{x}) \mid D\Big]. \tag{2.26}$$

Notice that KG is non-myopic, since acquisition values (2.26) depend on both the distribution of outcomes $\boldsymbol{y} = f(\mathbf{X})$ and the location of the batch $\mathbf{X} \in \mathcal{X}^q$.

Comparing EI and KG, two primary considerations are as follows. First, one typically uses a model to represent their belief about a black-box function f. Generally speaking, the agent's faith in the chosen prior for f and, hence, the model will vary by case. In cases where the model is deemed sufficiently trustworthy, the best-expected rule is typically preferred to the best-seen rule. Second, EI is easier to use and less computationally demanding than KG, which may bias practitioners in its favor. Recent works such as Wu and Frazier (2016), Wu et al. (2017), and (Balandat et al., 2020), however, have attacked this latter problem head on and demonstrated significant speed ups for KG acquisition functions. Overall, in cases where models are trustworthy and acquisition functions may be thoroughly optimized, KG is likely to outperform EI.

While each of the acquisition functions discussed so far have clear decision theoretic origins, this is not always the case. Rather than terminating an optimal lookahead strategy after, e.g., a single step, many acquisition functions attempt to distill the high-level behavior of optimal strategies into simpler ones. A common paradigm for intuiting these behaviors is the *explore-exploit tradeoff*. On the one hand, the agent must explore the domain $\mathcal{X}$ in order to learn about the global trends exhibited by a black-box f; on the other, they must exploit what is already known about f in order to identify local optima in the first place. Exploration helps ensure that local optima are global optima, but is unlikely to immediately yield better incumbents. Conversely, exploitation frequently leads to better incumbents, but typically conveys little additional information about f. While Bayes optimal strategies implicitly balance between these considerations, popular acquisition functions such as Upper

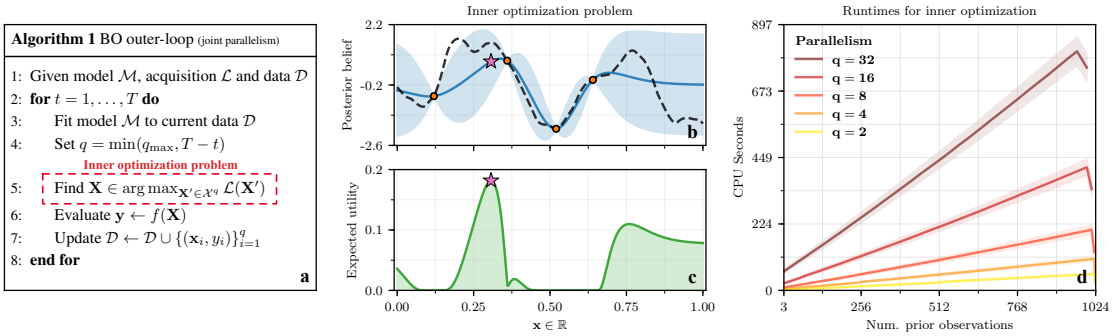

Figure 2.1: Overview of Bayesian optimization. *Left:* Pseudo code for a generic BO algorithm. *Middle:* A prototypical model-based posterior, with observations denoted by orange circles; and, a corresponding acquisition surface. In both plots, the next query location is indicated by a pink star. *Right:* Time to compute 2^{14} acquisition values given varying amounts of data and batch-sizes. At the final step, runtimes fall of because batch-sizes $q = \min(q_{\max}, T - t)$ diminish to satisfy an evaluation budget constraint $T = 1024$.

Confidence Bounds (Srinivas et al., 2010a) do so explicitly.

Over and beyond mere intuition, these heuristics (incl. one-step optimal acquisition functions) are often justified through rigorous analysis of their asymptotic properties. It is easy to see that maximizing the utility of a (dynamically chosen) incumbent $\boldsymbol{\chi}$ is equivalent to minimizing the expected value of its associated *regret*

$$r(\boldsymbol{\chi}) = \max_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) - f(\boldsymbol{\chi}). \tag{2.27}$$

Grünewälder et al. (2010) bound the simple regret incurred by a best-seen incumbent at time T under the optimal strategy (2.13) for Gaussian process priors (with Hölder continuous kernels and known hyperparameters) on f, while Vazquez and Bect (2010) and Bull (2011) bound the regret incurred by Expected Improvement in similar settings. Again for GP priors on f, Srinivas et al. (2010a) derive regret bounds for the Upper Confidence Bound strategy as a function of the mutual information between black-box f and observations D. For the special case of noise-free observations y, De Freitas et al. (2012) modify this UCB algorithm to improve these bounds and connect them with bounds on the cumulative regret $\sum_{t=1}^{T} r(\boldsymbol{x}_t)$.

2.3 Inner optimization problems

All of the theory discussed so far builds on the premise that queries are obtained by globally maximizing acquisition functions, often referred to as solving the *inner optimization problem* (Gelbart et al., 2014; Martinez-Cantin, 2014; Wang et al., 2016; Wilson et al., 2018). However, this seemingly innocent assumption presents a major challenge to its real-world applications. In practice, a modest amount of resources (usually, time or compute power) are allocated to decision-making itself. One often spends minutes choosing designs that take days to evaluate. Much of

this can be attributed to the fact that some inner optimization problems are (much) more expensive than others.

This state of affairs is partially attributable to the fact that, in general, models become increasingly costly and acquisition functions become increasingly multimodal as more and more data is collected. Even when dealing with a specific querying strategy, it can therefore be difficult to prescribe a "one-size-fits-all" approach to solving inner optimization problems. This problem is exacerbated by the fact that different querying strategies often incur dramatically different decision-making costs.

To help see this, consider two outer-loops based on the same of acquisition function: one in which a single query is chosen in each of T iterations and one in which T queries are made in the first (and only) iteration. Model fitting overheads notwithstanding, decision-making costs will typically be higher in the latter scenario. In addition the size of the search space simply being bigger when dealing with batches of queries, popular acquisition functions are frequently analytic for individual queries, but intractable for batch-sizes $q > 1$. Consequently, batch selection problems often rely on unbiased (but comparatively expensive) estimators of acquisition values. These issues are compounded by the fact that batched strategies essentially trade sample efficiency for wall-time efficiency, which further increases decision-making costs since it implies that they must collect more queries in order to achieve the same level of performance.

The remainder of this chapter investigates techniques for efficiently maximizing acquisition functions, with an emphasis on batch selection problems. We begin by developing Monte Carlo gradient estimators for acquisition functions, which allow the same powerful gradient-based methods to be used in both sequential and batched BO settings. When then demonstrate that a family of batch acquisition functions (incl. Expected Improvement) are submodular. This property provides strong justification for the use of greedy maximization techniques, which enables us to avoid some of the issues mentioned above by converting the batch selection problem into a sequence of subproblems in which $q = 1$. Lastly, we show how to combat the loss of analytic expression when moving from purely sequential to batch selection problems by using Rao-Blackwellization to constructed better estimators.

2.4 Pathwise gradient estimators

Derivatives are one of the most valuable sources of information when seeking to optimize a function, since they tell us whether moving its arguments ever so slightly in a particular direction will improve its value. This information powers a wide variety of efficient local optimization algorithms; and, this efficiency is vital when the cost of evaluating the objective is high, relative to the size of the search space and particulars of the task at hand. Here, we detail conditions under which averaging gradients obtained by differentiating through each sample in a Monte Carlo integral produces an unbiased gradient estimator.

Let $V(\ \cdot\ \mid D) : \mathrm{Fin}(\mathcal{X}) \to \mathbb{R}$ be an acquisition function defined per (2.15) as the conditional expectation of a value function $v : \mathcal{D} \to \mathbb{R}$ given observations $D \in \mathcal{D}$,

where $\mathcal{D} = \mathrm{Fin}(\mathcal{X} \times \mathbb{R})$. In many cases, exact computation of $V(\mathbf{X} \mid D)$ is infeasible due to the involvement of one or more intractable integrals. A popular option is therefore to use a Monte Carlo estimator as a proxy for the original function.

Consider a generic Monte Carlo estimator of an acquisition function V, namely

$$\widetilde{V}(\mathbf{X} \mid D) = \frac{1}{m} \sum_{i=1}^{m} v\Big(\mathbf{X}, \boldsymbol{\gamma}^{(i)} \mid D\Big), \tag{2.28}$$

where $\boldsymbol{\gamma}^{(i)}$ denotes the i-th realization of $\boldsymbol{y} = f(\mathbf{X})$. Supposing this estimator is unbiased, we would like to verify whether the same can be said of the corresponding gradient estimator

$$\widetilde{V}(\mathbf{X} \mid D) = \frac{1}{m} \sum_{i=1}^{m} \nabla_{\mathbf{X}} v\Big(\mathbf{X}, \boldsymbol{\gamma}^{(i)} \mid D\Big). \tag{2.29}$$

Validating this gradient estimator requires us to show that: (i) the derivative of $\boldsymbol{\gamma}^{(i)}$ with respect to $\mathbf{X}$ is well-defined and (ii) the order of integration and differentiation way be swapped. We first explore these concept under the assumption that draws of $\boldsymbol{y}$ are obtained by evaluating random functions $f(\,\cdot\,, \omega)$, before turning our attention to cases where $\boldsymbol{y}$ is generated by sampling from a distribution whose parameters depend on $\mathbf{X}$. Throughout, we assume an arbitrary but fixed choice of $D \in \mathcal{D}$.

For convenience, define $v_f(\mathbf{X}, \omega) = v(\mathbf{X}, (f \mid D)(\mathbf{X}, \omega) \mid D)$. Assuming it exists, the *pathwise derivative* of v_f with respect to the (i, j)-th site parameter $X_{ij} = [\mathbf{X}]_{ij}$ is given by

$$\frac{dv_f}{dX_{ij}}(\mathbf{X}, \omega) = \lim_{\lambda \to 0} \frac{v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega) - v_f(\mathbf{X}, \omega)}{\lambda}, \tag{2.30}$$

where $[\boldsymbol{e}_{ij}]_{kl} = \mathbb{1}_{ij=kl}$ is the basis vector associated with X_{ij}. Here, the term "pathwise" emphasizes the fact that (2.30) is the derivative of v_f along the path specified by $\omega \in \Omega$. Regarding the first of the two questions raised above: when draws of $\boldsymbol{y}$ are obtained as $\boldsymbol{\gamma}(\omega) = f(\mathbf{X}, \omega)$, it is clear that the pathwise derivative of v_f with respect to X_{ij} will exist so long as $v(\,\cdot\,\mid D)$ is differentiable with respect to $f(\mathbf{X}, \omega)$ and both functions are differentiable with respect to $\mathbf{X}$.

Now that we have defined the derivative of $\boldsymbol{\gamma}(\omega)$ with respect to $\mathbf{X}$, let us investigate the question of whether or not the expectation of v's derivative is equivalent to the derivative of v's expectation. Expanding both quantities and exploiting linearity of expectation, it is immediately clear that the central question is whether the limit may be brought inside the integral so that

$$\begin{aligned} \nabla_{\mathbf{X}}\, \mathbb{E}[v(\mathbf{X}, \omega)] &= \lim_{\lambda \to 0} \mathbb{E}\left[\frac{v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega) - v_f(\mathbf{X}, \omega)}{\lambda}\right] \\ &= \mathbb{E}\left[\lim_{\lambda \to 0} \frac{v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega) - v_f(\mathbf{X}, \omega)}{\lambda}\right] = \mathbb{E}[\nabla_{\mathbf{X}} v_f(\mathbf{X}, \omega)]. \end{aligned} \tag{2.31}$$

Interchanges of this sort are typically validated by appealing to Lebesgue's dominated convergence theorem (Glasserman, 1988; Mohamed et al., 2020). By definition, we know that the finite difference term converges pointwise to the pathwise derivative

(if it exists)

$$\frac{v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega) - v_f(\mathbf{X}, \omega)}{\lambda} \xrightarrow{\lambda \to 0} \frac{dv_f}{dX_{ij}}(\mathbf{X}, \omega). \tag{2.32}$$

Consequently, if there exists an integrable function $\psi : \mathcal{X}^q \times \Omega \to \mathbb{R}$ that almost surely dominates this term in the sense that, for all $\lambda \in \mathbb{R}$ and for all $\omega \in \Omega$ a.s.,

$$\psi(\mathbf{X}, \omega) \geq \left| \frac{v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega) - v_f(\mathbf{X}, \omega)}{\lambda} \right|, \tag{2.33}$$

then one may show that

$$\lim_{\lambda \to 0} \mathbb{E} \left| \frac{dv_f}{dX_{ij}}(\mathbf{X}, \omega) - \frac{v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega) - v_f(\mathbf{X}, \omega)}{\lambda} \right| = 0, \tag{2.34}$$

and, hence, that (2.31) holds. Necessary and sufficient conditions for the existence of a dominating function ψ are that v_f is sample-continuous and $\frac{dv_f}{dX_{ij}}$ exists and is integrable (Cao, 1985; Glasserman, 1988). When π is atomless, these conditions are satisfied by most continuous, piecewise differentiable value functions v. In practice, then, "the most important condition is the continuity of $[v]$ across points where it fails to be differentiable" Glasserman (1988, page 2).

In Chapter 3, we will develop techniques for efficiently sampling (approximate) function draws $f(\,\cdot\,, \omega)$ and show that they offer significant advantages when $q = |\mathbf{X}|$ is large. In many practical settings, however, q will be small and it will be more convenient for us to generate random vectors $\boldsymbol{\gamma}$ by another means.

Recall from the beginning of the chapter that f is assumed to uniquely extend a family of finite-dimensional distributions over elements of $\mathrm{Fin}(f)$. For a fixed $q \in \mathbb{N}$, suppose we are handed a function $\theta : \mathcal{X}^q \to \Theta$ and told that θ parameterizes the q-dimensional distributions of $f \mid D$ in the sense that there exists an independent random variable $\boldsymbol{z} \sim \pi$ and a continuously differentiable function $g : \Theta \times \mathcal{Z} \to \mathbb{R}^q$ satisfying[3]

$$(f \mid D)(\mathbf{X}) \stackrel{\mathrm{d}}{=} g(\theta(\mathbf{X}), \boldsymbol{z}), \text{ for all } \mathbf{X} \in \mathcal{X}^q. \tag{2.35}$$

For convenience, define $v_g(\mathbf{X}, \boldsymbol{z}) = v(\mathbf{X}, g(\theta(\mathbf{X}), \boldsymbol{z}))$. Noting our earlier conditions for differentiation under the integral sign, let us further assume that v, θ, and g are all continuously differentiable so that $\frac{dv_g}{dX_{ij}}$ is integrable. Now, since $v_f(\mathbf{X}, \omega) \stackrel{\mathrm{d}}{=} v_g(\mathbf{X}, \boldsymbol{z})$ for all $\mathbf{X} \in \mathcal{X}^q$, it follows that

$$\begin{aligned} \frac{d\,\mathbb{E}[v_f]}{dX_{ij}}(\mathbf{X}) &= \lim_{\lambda \to 0} \frac{\mathbb{E}[v_f(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \omega)] - \mathbb{E}[v_f(\mathbf{X}, \omega)]}{\lambda} \\ &= \lim_{\lambda \to 0} \frac{\mathbb{E}[v_g(\mathbf{X} + \lambda \boldsymbol{e}_{ij}, \boldsymbol{z})] - \mathbb{E}[v_g(\mathbf{X}, \boldsymbol{z})]}{\lambda} = \frac{d\,\mathbb{E}[v_g]}{dX_{ij}}(\mathbf{X}), \end{aligned} \tag{2.36}$$

Like $\frac{dv_f}{dX_{ij}}(\mathbf{X}, \omega)$ before it, $\frac{dv_g}{dX_{ij}}(\mathbf{X}, \boldsymbol{z})$ may be used to construct a pathwise gradient

[3] Continuously differentiability of g helps to ensure that v_g is integrable and, hence, that we may differentiate under the integral sign.

estimator (2.29). Rather than generating f all at once, however, paths $g(\theta(\ \cdot\ ), \boldsymbol{z})$ now realize its q-dimensional subsets.

In the machine learning literature, the act of differentiating through samples obtained by pushing an independent random variable $\boldsymbol{z}$ forward through a function $g : \Theta \times \mathcal{Z} \to \mathbb{R}^q$ parameterized by a vector $\theta(\mathbf{X})$ is often referred to as the *reparameterization trick* (Kingma and Welling, 2014) or stochastic backpropagation (Rezende et al., 2014). Elsewhere, it is sometimes referred to as infinitesimal perturbation analysis (Cao, 1985; Glasserman, 1988) or, simply, pathwise differentiation (Glasserman, 2013). For a recent survey of these techniques and more, see (Mohamed et al., 2020).

The preceding definition of g as a (continuously) differentiable function parameterized by $\theta(\mathbf{X})$ is broad enough to encompass most reparameterizations of $\boldsymbol{y}$, but does not necessarily lend itself to insight. Supposing $g(\theta(\mathbf{X}),\ \cdot\ )$ is invertible, an alternative point of view is to think of $g(\theta(\mathbf{X}),\ \cdot\ )^{-1}$ as a "standardization function" that removes $\boldsymbol{y}$'s dependence on $\theta(\mathbf{X})$ (Figurnov et al., 2018). This is important because this additional dependence, left unaccounted for, would otherwise bias our gradient estimates. Thinking in terms of standardization functions $g(\theta(\mathbf{X}),\ \cdot\ )^{-1}$ is particularly natural when dealing with continuous random variables $y \in \mathbb{R}$. In these cases, a reasonable choice is to let $z \sim \mathcal{U}(0, 1)$ be uniformly distributed and define g as the inverse of y's cumulative distribution function (CDF). Despite their intuitive appeal, however, inverse transform sampling methods of this sort are often computationally demanding and may not extend well to the multivariate setting $\boldsymbol{y} \in \mathbb{R}^q$. Readers interested in learning more about strategies for generating random variables should consult (Devroye, 2006).

For now, let us suppose that $f \sim \mathcal{GP}(\mu, k)$ is a Gaussian process—i.e. a stochastic process whose finite-dimensional subsets $\mathrm{Fin}(f)$ are all multivariate normally distributed—given in terms of a mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. These functions have the requisite property that, for all $\mathbf{X} \in \mathrm{Fin}(\mathcal{X})$,

$$f(\mathbf{X}) \sim \mathcal{N}(\mu(\mathbf{X}), k(\mathbf{X}, \mathbf{X})). \tag{2.37}$$

When it comes to drawing random vectors $f(\mathbf{X})$ with covariance $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$, the typical approach is to linearly transform a base random vector $\boldsymbol{z}$ so that

$$\mathrm{Cov}(\mathbf{A}\boldsymbol{z}) = \mathbf{A}\,\mathrm{Cov}(\boldsymbol{z})\mathbf{A}^\top = \mathbf{K}, \tag{2.38}$$

where $\mathbf{A}$ is the matrix representation of the aforementioned transform (Devroye, 2006, Section 2.2). Since the family of Gaussian random variables is closed under affine transformations, this procedure typically manifests as a location scale transform

$$f(\mathbf{X}) \overset{\mathrm{d}}{=} \mu(\mathbf{X}) + k(\mathbf{X}, \mathbf{X})^{1/2}\boldsymbol{z} \qquad\qquad \boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2.39}$$

where $k(\mathbf{X}, \mathbf{X})^{1/2}$ is a matrix square root of $k(\mathbf{X}, \mathbf{X})$. Supposing (2.39) is used to generate $f(\mathbf{X})$, it follows that $\mu(\mathbf{X})$ and $k(\mathbf{X}, \mathbf{X})^{1/2}$ must be continuously differentiable with respect to $\mathbf{X}$ in order for the pathwise gradient estimator (2.29) to be valid. Wang et al. (2016) show that the latter condition is met by GPs with twice differentiable kernels k, so long as the elements of $\mathbf{X}$ are unique. The authors then go on to show

that (2.29) is an unbiased gradient estimator for Expected Improvement. Subsequent works would go on to show that this claim also holds for Knowledge Gradient acquisition functions (Wu and Frazier, 2016; Wu et al., 2017).

In the remainder of this section, we explore the practical implications of gradient-based versus gradient-free approaches to maximizing acquisition functions through a series of experiments.

2.4.1 Overview of experiments

This section outlines the general setup of the empirical studies presented in this chapter. Throughout, experiments were designed to isolate the impact different approaches to solving the inner optimization problem have on outer-loop performance. To help streamline discussion, we focus on results when queries are made by maximizing Expected Improvement.

On the whole, we investigated performance in two distinct scenarios: *synthetic tasks* where the ground-truth function f was drawn from a known GP prior; and, *black-box tasks* where the nature of f is unknown at the start of optimization. Dividing our experiments this way enables us to better understand the inner optimization problem's impact by isolating the effects of model mismatch. In both cases, we employ a GP prior $f \sim \mathcal{GP}(\mu, k)$ with a constant mean function $\mu(\,\cdot\,) = c \in \mathbb{R}$ and an anisotropic Matérn-$5/2$ kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_k^2 \Big(1 + \sqrt{5}r + \tfrac{5}{3}r^2\Big) \exp(-\sqrt{5}r), \tag{2.40}$$

where $r^2 = (\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Lambda}^{-1} (\boldsymbol{x} - \boldsymbol{x}')$ for some diagonal matrix $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \dots \lambda_d)$ with $\lambda_i > 0$. When optimizing functions $f : \mathbb{R}^d \to \mathbb{R}$ drawn from GP priors, we set $c = 0$, $\sigma_k = 1$, $\lambda_i = \frac{d}{16}$. In the black-box setting, these hyperparameters were estimated online at the start of each outer-loop iteration. In all cases, trials began with three randomly chosen queries and proceeded until a total of T designs had been evaluated. Finally, while the general notation of this chapter has assumed noise-free observations, all experiments were run with Gaussian observations $y \mid f(\boldsymbol{x}) \sim \mathcal{N}(f(\boldsymbol{x}), 10^{-3})$.

We considered a range of (acquisition) maximizers, ultimately settling on stochastic gradient ascent using Adam (Kingma and Ba, 2015), Covariance Matrix Adaptation Evolution Strategy CMA-ES, (Hansen, 2006), and Random Search RS (Bergstra and Bengio, 2012). To ensure fairness, maximizers were constrained by CPU runtime. At each outer-loop iteration, a runtime budget was established by measuring the average amount of time required to evaluate N batch acquisition values (carried out in parallel). For the greedy strategies introduced in Section 2.5, this budget was split evenly among each of q rounds. To characterize performance as a function of allocated runtime, experiments were run using inner budgets $N \in \{2^{12}, 2^{14}, 2^{16}\}$. Lastly, we note that the strategy used to initialize the various optimizers will be introduced and motivated in Section 2.5.1.

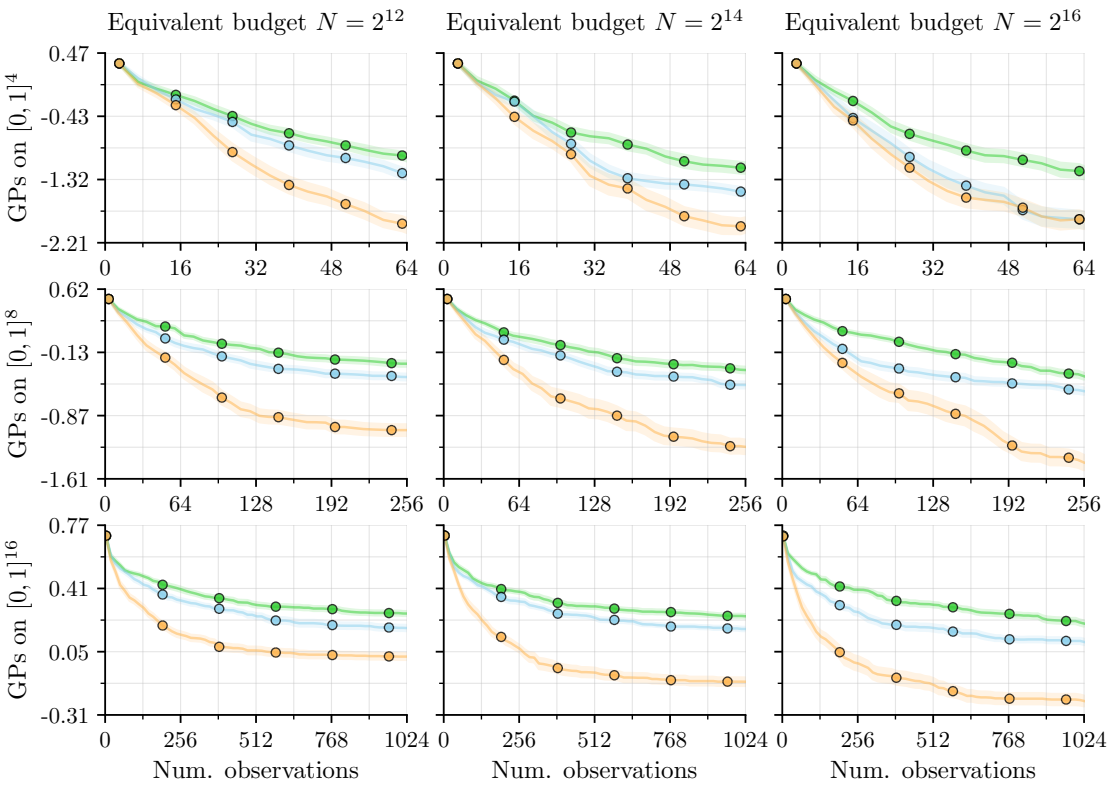

Figure 2.2: Average performance—means and standard errors of $\log_{10}$ immediate regrets incurred by best-seen incumbents over 32 independent trials—when optimizing functions drawn from known GP priors using different approaches to maximizing Monte Carlo EI. Random Search is shown in green, CMA-ES in blue, and stochastic gradient ascent in yellow. The dimensionality of the search space d increases from top to bottom rows (each time with batch-size $q = d$), while the time spent solving the inner optimization problem increases from left to right columns.

2.4.2 Results

Before diving into the results shown in Figures 2.2 and 2.3, let us first clarify how the Monte Carlo (gradient) estimators discussed in this section were used. When such an estimator is constructed from reparameterized samples $\boldsymbol{\gamma} = g(\boldsymbol{\theta}, \boldsymbol{\zeta})$, we may either optimize it stochastically (by resampling $\boldsymbol{\zeta}$) or deterministically (by holding $\boldsymbol{\zeta}$ fixed). Together with a choice of sample count m, this decision reflects a well-known tradeoff of approximation-based, estimation-based, and optimization-based sources of error (Bousquet and Bottou, 2008). Here, approximation error reflects the potential and likely mismatch between the idealized prior for f and the one we employ; estimation error conveys the difference between the Monte Carlo estimator and the true acquisition function (or gradients); and, optimization error communicates how much worse the batches we obtain are from globally optimal ones.

Resampling $\boldsymbol{\zeta}$ typically eliminates estimation error but increases optimization error. Conversely, recycling $\boldsymbol{\zeta}$ helps cut down on optimization error but introduces a bias. In our experiments, we found that stochastic gradient methods—specifically Adam

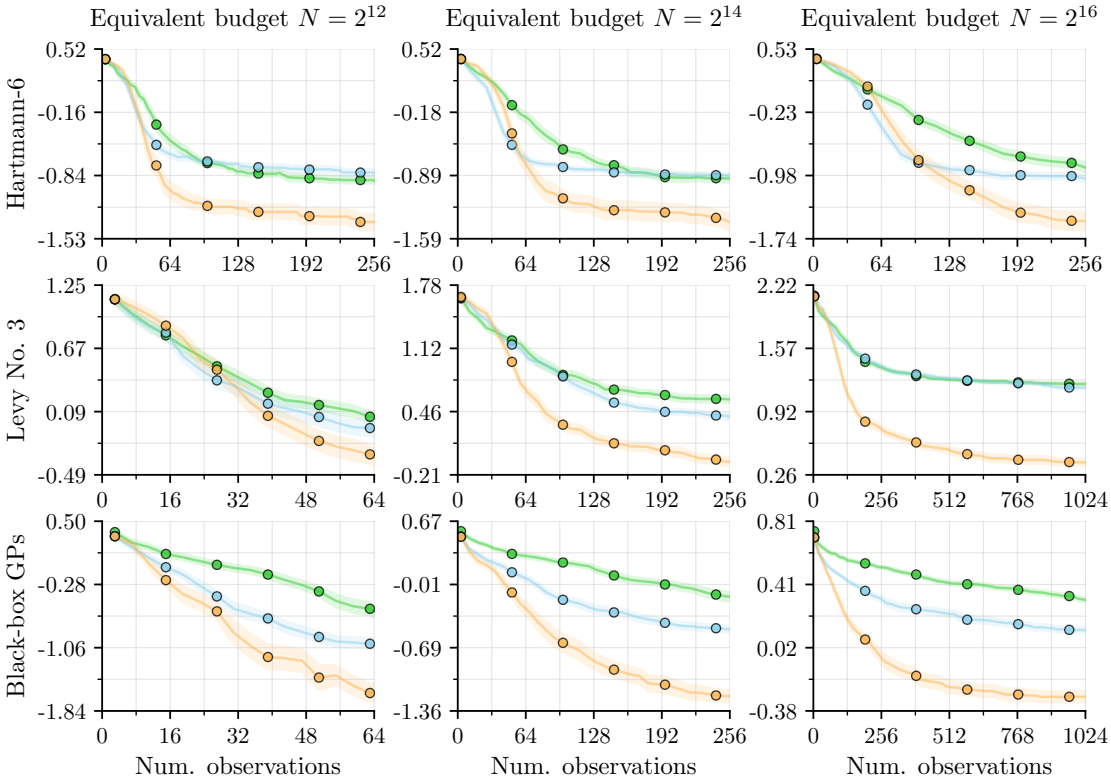

Figure 2.3: Comparison of BO performance when optimizing black-box objectives using different approaches to maximizing Monte Carlo EI; analogous to Fig. 2.2. Random Search is shown in green, CMA-ES in blue, and stochastic gradient ascent in yellow. When optimizing Levy No.3 and draws from unknown GP priors, design dimensionality d and batch-size q increase as columns go from left to right as $d = q \in (4, 8, 16)$. For Hartmann-6, $d = 6$ remains unchanged but q increases according to the aforementioned schedule.

(Kingma and Ba, 2015) with an initial learning rate of $\frac{1}{40}$ and $m = 128$ samples—consistently matched or outperformed deterministic ones with varying sample counts. Similar trends were observed for the gradient-free CMA-ES optimizer, results for which are shown using stochastic evaluations of Monte Carlo acquisition functions. Since the time of original publication, subsequent works have demonstrated benefits for using deterministic optimizers with quasi-random draws $\boldsymbol{\zeta}$ (Balandat et al., 2020).

Figures 2.2 paints a clear picture of the inner optimization problem's impact on outer-loop performance by investigating cases where model-based errors have been eliminated. On these synthetic tasks, the benefits of gradient-based approaches to maximizing Monte Carlo estimators are on full display. By exploiting gradient information, we are able to obtain better batches in less time.

Aside from gradient-based methods (yellow) consistently matching or surpassing gradient-free alternatives (green and blue), two additional trends bear immediate mention. First (as columns go from left to right), we see how increasing the amount of time spent maximizing acquisition functions improves performance. Second (as d increases from top to bottom rows), the performance of all methods deteriorates, both because the number of allowed queries only increases linearly and because the

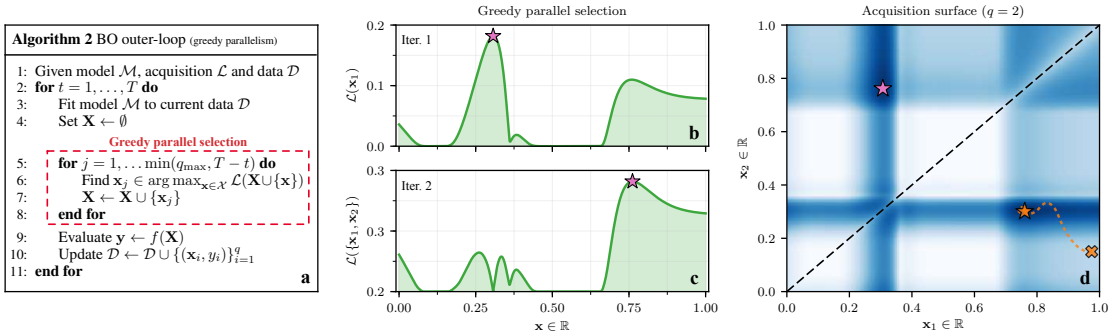

Figure 2.4: *Left:* Pseudo-code for BO outer-loop with greedy parallelism, the inner optimization problem is boxed in red. *Middle:* Successive iterations of greedy maximization, starting from the posterior shown in Figure 1b. *Right:* On the upper left, greedily selected query '$\star$'; on the lower right and from '$\times$' to '$\star$', trajectory when jointly optimizing parallel queries $\mathbf{X}_1$ and $\mathbf{X}_2$ via stochastic gradient ascent. Darker colors correspond with larger acquisitions.

time allocated to the inner optimization problem remains unchanged. Nevertheless, gradient-based approaches are able to better utilize this time and quickly pull ahead of their gradient-free competitors.

The same general trends appear in the black-box task setting (shown in Figure 2.3), with gradient-based approaches consistently yielding the best performance. The final row of Figure 2.3 reproduces the experiments shown on the main diagonal of Figure 2.2, but replaces known priors with Type-II maximum likelihood estimated hyperparameters fit at the start of each outer loop iteration. As the amount of data collected within each trial increases (from left to right columns), these estimates become increasingly accurate and we recover performance nearly identical to that observed on synthetic tasks. Finally, performance on Hartmann-6 (top row) serves as a clear indicator for the importance of thoroughly solving the inner optimization problem. In these experiments, performance improved despite mounting batch-sizes due to a corresponding increase in the inner budget.

Overall, these results clearly demonstrate that gradient-based approaches to maximizing acquisition functions improve outer-loop performance. Furthermore, these gains become more pronounced as the batch dimensionality qd increases.

2.5 Greedy batch selection

Optimally selecting a batch is considerably more difficult than choosing a single query. Per the previous section, this difficulty is partially due to the prevalence of intractable integrals. A second, equally troublesome issue is that finding an optimal batch $\mathbf{X}_j^* \in \arg\max_{\mathbf{X} \in \mathcal{X}^j} V(\mathbf{X} \mid D)$ is typically a high-dimensional, global optimization problem unto itself. This section investigates the mathematical justification for

greedy batch selection whereby, for all $i = 1, \ldots, q$, we have

$$\bar{\mathbf{X}}_i = \bar{\mathbf{X}}_{i-1} \cup \left\{ \arg\max_{\boldsymbol{x} \in \mathcal{X}} V\Big(\bar{\mathbf{X}}_{i-1} \cup \{\boldsymbol{x}\} \mid D\Big) \right\} \qquad \bar{\mathbf{X}}_0 = \emptyset. \tag{2.41}$$

Greedy approaches to constructing a batch decompose the original qd-dimensional problem into a sequence of q, d-dimensional subproblems. By the curse of dimensionality, the cost of solving these (sub)problems increases superlinearly (often exponentially) in its dimensionality. Hence, greedy strategies are far easier to carry out. A number of prior works have, therefore, proposed greedy batch selection as a practical way of tackling real-world factors such as time constraints on decision-making per se (Azimi et al., 2010; Chen and Krause, 2013; Contal et al., 2013; Desautels et al., 2014; Shah and Ghahramani, 2015; Kathuria et al., 2016). Here, we will show that many common batch acquisition functions are *submodular*. Of acquisition functions shown in Table 2.1, this includes EI, EMAX, UCB, and PI.

Definition 2.11 (Submodularity)**.** *Let $V : 2^{\mathcal{X}} \to \mathbb{R}$ be a function on the set of all subsets of a finite collection $\mathcal{X}$. Then, V is said to be submodular if it satisfies either of the following equivalent conditions for all $\mathbf{X}, \mathbf{X}' \subseteq \mathcal{X}$*

a. $V(\mathbf{X}) + V(\mathbf{X}') \geq V(\mathbf{X} \cup \mathbf{X}') + V(\mathbf{X} \cap \mathbf{X}')$.

b. *If $\mathbf{X} \subseteq \mathbf{X}'$, then $V(\mathbf{X} \cup \{\boldsymbol{x}\}) - V(\mathbf{X}) \geq V(\mathbf{X}' \cup \{\boldsymbol{x}\}) - V(\mathbf{X}')$, $\forall \boldsymbol{x} \in \mathcal{X} \setminus \mathbf{X}'$.*

According to Definition 2.11b, then, a batch acquisition function is submodular (SM) if the marginal value for querying $\boldsymbol{x} \notin \mathbf{X}$, i.e. $V(\mathbf{X} \cup \{\boldsymbol{x}\}) - V(\mathbf{X})$, does not increase as additional designs $\boldsymbol{x}' \notin \mathbf{X}$ are added to $\mathbf{X}$. Note that, for the remainder of this section we assume that the domain $\mathcal{X}$ is finite. This assumption has little bearing in practical settings, it is necessary in order for V to satisfy Definition 2.11. Readers interested in learning more about submodularity and its applications should see (Bach, 2013; Krause and Golovin, 2014). Below, we will justify greedy batch selection by appealing to a well-known result regarding greedy maximization of submodular functions (Nemhauser et al., 1978; Krause and Golovin, 2014).

Theorem 2.12. *If $V : 2^{\mathcal{X}} \to \mathbb{R}_+$ is a nonnegative monotone submodular function, then*

$$V\Big(\bar{\mathbf{X}}_i\Big) \geq \Big(1 - e^{-i/j}\Big) \max_{\mathbf{X} \in \mathcal{X}^j} V(\mathbf{X}), \text{ for all } i, j \in \mathbb{N},$$

where $\bar{\mathbf{X}}_i$ is the set obtained after i rounds of greedy selection.

Proof See Krause and Golovin (2014, page 7). ■

Several works on Gaussian-process-based optimization have previously exploited submodularity (Srinivas et al., 2010a; Contal et al., 2013; Desautels et al., 2014). These works typically leverage the submodularity of an auxiliary quantity — such as the mutual information between a Gaussian process f and observations D — to bound the idealized performance of a particular querying strategy. In contrast, we

will show that many batch acquisition functions are submodular and use this fact to bound the local error introduced at each step by greedily solving for optimal batches.

We begin by simplifying some of the machinery at hand. Suppose that V is a myopic acquisition function such that, for some value function $v : 2^{\mathcal{Y}} \to \mathbb{R}$, we may write

$$V(\mathbf{X} \mid D) = \mathbb{E}_{p(\boldsymbol{y}|D)}[v(\boldsymbol{y} \mid D)] = \mathbb{E}_{p(f|D)}[v(f(\mathbf{X}) \mid D)], \tag{2.42}$$

where $p(f \mid D)$ is valid since $\mathcal{X}$ is assumed finite. Suppressing D to ease notation, it follows that V is submodular if and only if the same can be said of v, since

$$\begin{aligned} V(\mathbf{X} \cup \mathbf{X}') + V(\mathbf{X} \cap \mathbf{X}') &= \int \Big[v(f(\mathbf{X} \cup \mathbf{X}')) + v(f(\mathbf{X} \cap \mathbf{X}')) \Big] \, \mathrm{d}p(f) \\ &\leq \int \Big[v(f(\mathbf{X})) + v(f(\mathbf{X}')) \Big] \, \mathrm{d}p(f) \\ &= V(\mathbf{X}) + V(\mathbf{X}'), \end{aligned} \tag{2.43}$$

where the second line follows by Definition 2.11a. When V admits (2.42), we therefore only need to determine whether the corresponding value function v is submodular.

As has been a recurring theme in this chapter, acquisition functions can often be viewed in terms of incumbent rules. Specifically, the acquisition value of a batch $\mathbf{X}$ communicates how the incumbent's utility is expected to change. In myopic cases (where the value of querying $\mathbf{X}$ does not account for its influences over our understanding of f on $\mathcal{X} \setminus \mathbf{X}$), such changes are only possible when an element of $\mathbf{X}$ is taken as the incumbent. Myopic batch acquisition functions, therefore, often manifest as

$$V(\mathbf{X} \mid D) = \mathbb{E}_{p(\boldsymbol{y}|D)} \max\{v(y_1 \mid D), \ldots, v(y_q \mid D)\}. \tag{2.44}$$

In these cases, it suffices to show that the maximum is a submodular set function. Let $\mathcal{V}$ be a finite ground set and define $\max(\emptyset) = \inf \mathcal{V}$. Without loss of generality, suppose $\boldsymbol{u}, \boldsymbol{v} \subseteq \mathcal{Y}$ satisfy $\max(\boldsymbol{u}) \geq \max(\boldsymbol{v})$, such that $\max(\boldsymbol{u}) = \max(\boldsymbol{u} \cup \boldsymbol{v})$. Since $\max(\boldsymbol{v}) \geq \max(\boldsymbol{w})$ for all $\boldsymbol{w} \subseteq \boldsymbol{v}$, we have $\max(\boldsymbol{v}) \geq \max(\boldsymbol{u} \cap \boldsymbol{v})$. It follows that the maximum, defined as such, is submodular by Definition 2.11a:

$$\max(\boldsymbol{u}) + \max(\boldsymbol{v}) \geq \max(\boldsymbol{u} \cup \boldsymbol{v}) + \max(\boldsymbol{u} \cap \boldsymbol{v}). \tag{2.45}$$

We are not quite done yet, however. In order for Theorem 2.12 to bound the inner-loop regret of greedy batch selection, $\max \circ v \circ f$ must almost surely satisfy, $\forall \mathbf{X}, \mathbf{X}' \in \mathcal{X}$,

i. *Monotonic*: $(\max \circ v \circ f)(\mathbf{X} \cup \mathbf{X}') \geq (\max \circ v \circ f)(\mathbf{X})$.

ii. *Nonnegative*: $(\max \circ v \circ f)(\mathbf{X}) \geq 0$.

Seeing as monotonicity is guaranteed by the max operation, the challenge here is to show that $v \circ f$ is nonnegative. Note that, in practice, it suffices for $v \circ f$ to be bounded from below. In some cases, nonnegativity is implied by v. In others, it suffices to bound the conditional expectation of f. More generally, however, lower bounding $v \circ f$ may require us to show that $\inf f$ is finite. For a centered Gaussian

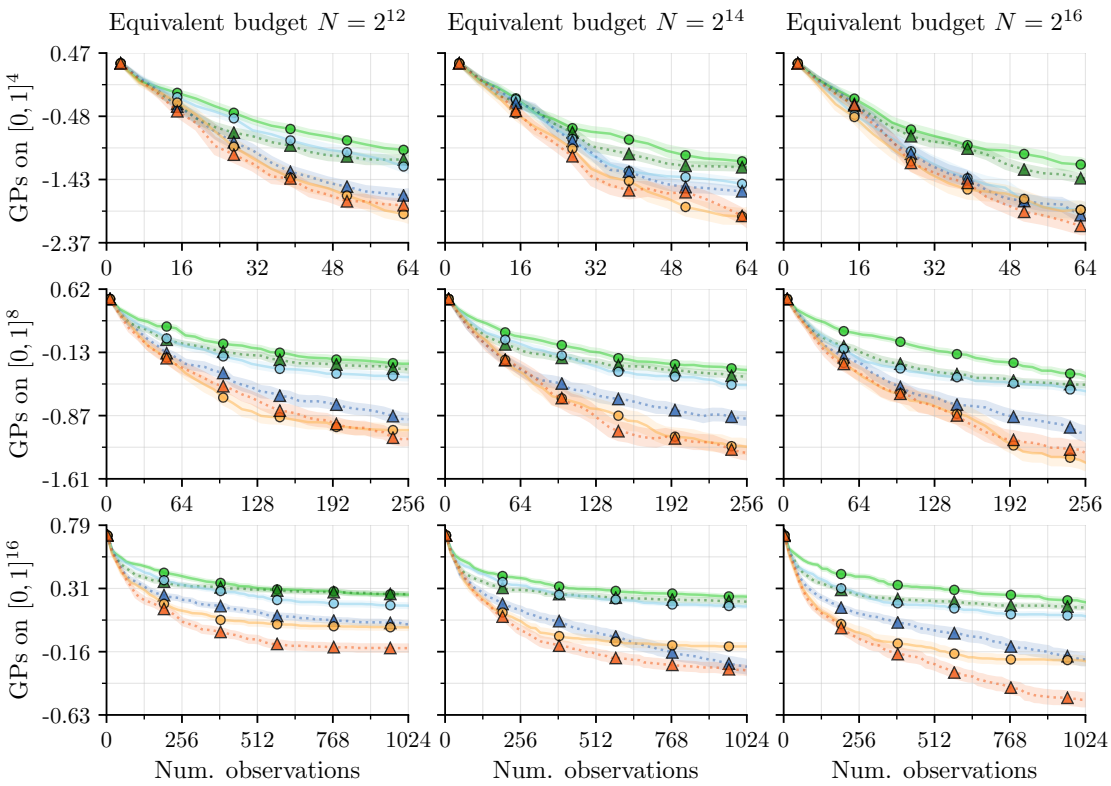

Figure 2.5: Performance comparison between joint (lighter colors) and greedy (darker colors) approaches to batch selection when using Monte Carlo EI to optimize functions drawn from known GP priors; these results continue from Figure 2.3. The performance of Random Search (as an acquisition function maximizer) is shown in greens, that of CMA-ES is shown in blues, and that of stochastic gradient ascent is shown in orange and yellow.

process f on a compact domain $\mathcal{X}$, asking whether f is almost surely bounded is equivalent to asking whether it is sample-continuous. Alternatively, it sometimes happens that f is known a priori to be bounded from below. Azimi et al. (2010), for example, focus on maximizing nonnegative functions f and exploit the ensuing submodularity of the EMAX acquisition function (2.22). Finally, we note that an alternative proof of submodularity for the Probability of Improvement acquisition function can be found in the supplementary material of Tallorin et al. (2018).

In summary, batches $\bar{\mathbf{X}}_i \in \mathcal{X}^i$ obtained by greedily maximizing a submodular acquisition function are near-optimal in the sense of the inner-loop regret bound

$$V\Big(\mathbf{X}_j^* \mid D\Big) - V\Big(\bar{\mathbf{X}}_i \mid D\Big) \leq e^{-i/j} V\Big(\mathbf{X}_j^* \mid D\Big), \text{ for all } i, j \in \mathbb{N}, \tag{2.46}$$

where $\mathbf{X}_j^* \in \arg\max_{\mathbf{X} \in \mathcal{X}^j} V(\mathbf{X})$ denotes the optimal size j batch. This theoretical justification in place, we now proceed to examine the practical impact of greedy batch selection.

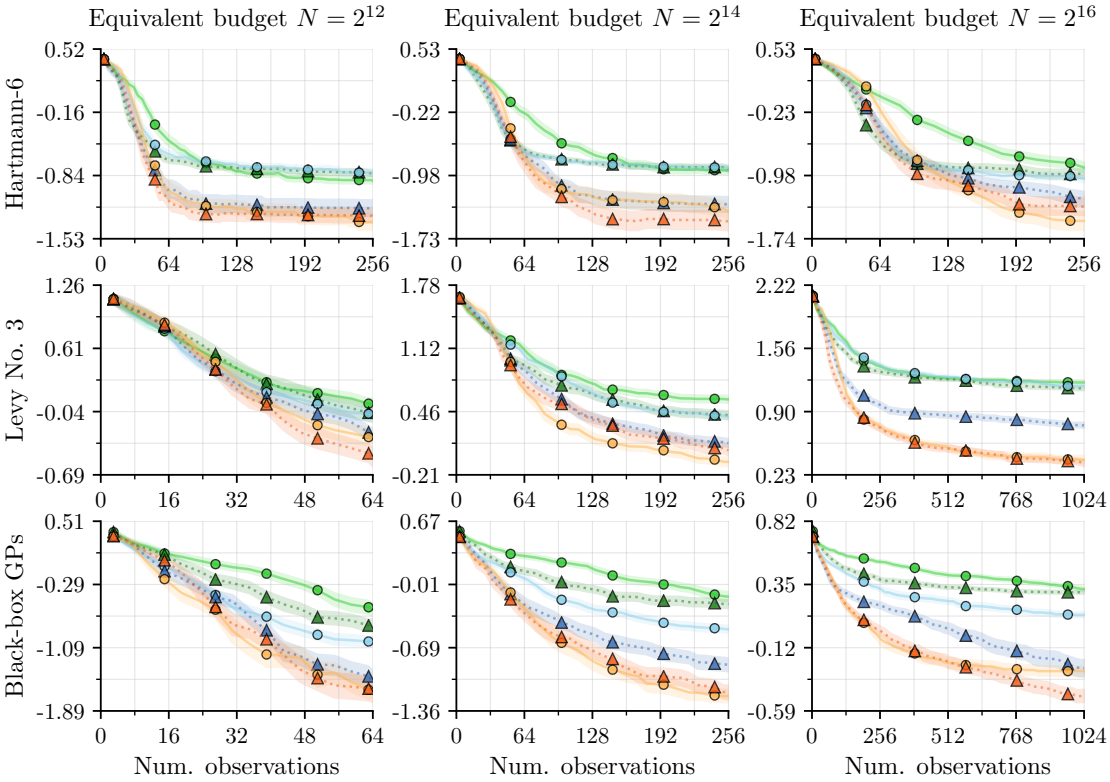

Figure 2.6: Analogue of Figure 2.5 when optimizing black-box functions; these results continue from Figure 2.2.

2.5.1 Results

We validated the greedy strategies motivated above by extending the study outlined at the end of the previous section. Each of the trials discussed there was repeated, this time using greedy batching. For simplicity, time allocated to solving the inner optimization problem was split evenly across rounds of greedy selection.

Prior to discussing the results for this section, we pause to motivate the strategy used to initialize the inner optimization problems faced in experiments presented in this chapter. As noted by Wang et al. (2016), local optimizers of acquisition functions are typically sensitive to the choice of starting positions. The primary reason for this issue is that acquisition surfaces are often highly non-convex, causing many would-be queries to get stuck in local regions of the design space $\mathcal{X}$.

Where applicable, we propose to combat these issues by appealing to submodularity. For submodular acquisition functions V, it follows that $V(\mathbf{X} \cup \{\boldsymbol{x}\}) \leq V(\boldsymbol{x})$ for all finite sets $\mathbf{X} \subseteq \mathcal{X}$ and designs $\boldsymbol{x} \in \mathcal{X}$. In the case of EI, this implies that any design for which $V(\boldsymbol{x}) \approx 0$ is incapable of substantially contributing toward the quality of an overarching batch. We therefore propose to initialize batches by sampling designs proportional to $V(\boldsymbol{x})$. Specifically, we draw initial batches by sampling q times without replacement from $V(\boldsymbol{x}) / \sum_{\boldsymbol{x}' \in \mathcal{X}_n} V(\boldsymbol{x}')$, where $\mathcal{X}_n$ is a size n discretization of $\mathcal{X}$ and $\boldsymbol{x} \in \mathcal{X}_n$. This initialization strategy can be executed efficiently and enables

us to avoid "inactive" regions of $\mathcal{X}$ by prioritizing more promising designs. Across our experiments, using this heuristic consistently led to improved performance.

Results shown in Figure 2.5 and Figure 2.6 demonstrate the benefits of greedy batching. Even in low-dimensional settings where inner optimization problems can be solved with ease, greedily selected batches perform on par with jointly selected ones. As batch dimensionality increases, however, greedy methods pull ahead due to their ability to decompose the original qd-dimensional problem into a sequence of q, d-dimensional subproblems. Crucially, the fact that these benefits are enjoyed by all of the tested acquisition function optimizers indicates that greedy batch selection is flexible and robust.

2.6 Rao-Blackwellization

In Section 2.4, we discussed how Monte Carlo gradient estimators can be used to efficiently maximize popular acquisition functions; and, in Section 2.5, we used submodularity to motivate greedy approaches for constructing near-optimal batches of queries. Both of these approaches are quite general and work well when combined. Nevertheless, we sometimes run into trouble when sampled quantities are sparse. This pathology is most commonly observed during the latter parts of optimizations, where much of the search space has been deemed uninteresting save for in case of rare events. To combat this issue, this section develops *Rao-Blackwellized* estimators for batch acquisition functions. For expediency, we again focus on batch acquisition functions defined as the expected maximum of an underling value function, leading to the naïve estimators

$$\widetilde{V}(\mathbf{X}) = \frac{1}{m}\sum_{i=1}^{m} \max v\Big(\boldsymbol{\gamma}^{(i)}\Big) \qquad \nabla_{\mathbf{X}}\widetilde{V}(\mathbf{X}) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\mathbf{X}} \max v\Big(\boldsymbol{\gamma}^{(i)}\Big). \tag{2.47}$$

Theorem 2.13 (Rao-Blackwell)**.** *Let $u : \mathcal{Y} \to \mathbb{R}$ be an estimator of an unknown quantity $\mu \in \mathbb{R}$ with a finite second moment for all μ. If θ is a sufficient statistic for μ, then*

$$\mathbb{E}\Big[(u^*(\theta) - \mu)^2\Big] \leq \mathbb{E}\Big[(u(y) - \mu)^2\Big], \text{ where } u^*(\theta) = \mathbb{E}[u(y) \mid \theta].$$

Proof $\mathbb{E}\Big[(u^*(\theta) - \mu)^2\Big] = \mathbb{E}\Big[(\mathbb{E}[u(y) - \mu \mid \theta])^2\Big] \leq \mathbb{E}\Big[\mathbb{E}\Big[(u(y) - \mu)^2 \mid \theta\Big]\Big] = \mathbb{E}\Big[(u(y) - \mu)^2\Big]$. ■

Proven independently by Rao (1945) and Blackwell (1947), the Rao-Blackwell theorem is a standard result regarding crude estimators and their optimal counterparts[4]. For our immediate purposes, this theorem tells us that we may improve upon the Monte Carlo estimators (2.47) by analytically integrating out one or more unknowns as in

$$V(\mathbf{X}) \approx \frac{1}{m}\sum\nolimits_{i=1}^{m} \mathbb{E}\Big[v(\boldsymbol{y}) \mid y_j = \gamma_j^{(i)} \text{ for all } j = \{\ldots\}\Big]. \tag{2.48}$$

[4]Here, optimality is typically defined with respect to mean squared error.

Below, we will show how (estimators of) acquisition functions discussed earlier in the text can be Rao-Blackwellized to enhance estimated acquisition values and gradients thereof. To this end, let us begin by introducing some additional notation.

Denote by $\boldsymbol{v}_q = v(\boldsymbol{y}_q) \in \mathbb{R}^q$ the result of pushing each element of a random vector $\boldsymbol{y}_q = f(\mathbf{X}_q)$ forward through a value function $v : \mathcal{Y} \to \mathbb{R}$. Likewise, write $\boldsymbol{\nu}_q = v(\boldsymbol{\gamma}_q)$ for an arbitrary realization $\boldsymbol{\gamma}_q$ of $\boldsymbol{y}_q$. Now, let the *marginal acquisition value* for adding a query $\boldsymbol{x} \in \mathcal{X}$ to a batch $\mathbf{X}_q \in \mathcal{X}^q$ be defined as

$$G(\boldsymbol{x} \mid \mathbf{X}_q) = V(\mathbf{X}_{q+1}) - V(\mathbf{X}_q) = \mathbb{E}_{p(\boldsymbol{y}_{q+1})}[\max \boldsymbol{v}_{q+1} - \max \boldsymbol{v}_q], \tag{2.49}$$

where $\mathbf{X}_{q+1} = \mathbf{X}_q \cup \{\boldsymbol{x}\}$ and $\boldsymbol{v}_{q+1} = v(f(\mathbf{X}_{q+1}))$. Setting $\mathbf{X}_0 = \emptyset$ and $V(\mathbf{X}_0) = 0$, it follows that the acquisition value of a batch $\mathbf{X}_q$ is given by the telescoping sum

$$V(\mathbf{X}_q) = \sum_{i=1}^{q} G(\boldsymbol{x}_i \mid \mathbf{X}_{i-1}), \text{ for all } \mathbf{X}_q \in \mathcal{X}^q \text{ and } q \in \mathbb{N}. \tag{2.50}$$

These quantities are particularly important when batches are constructed iteratively, as they constitute the maximization objectives at each round of greedy selection.[5] By law of total expectation, it follows that $G(\boldsymbol{x} \mid \mathbf{X}_q) = \mathbb{E}_{p(\boldsymbol{y}_q)}\Big[g(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q)\Big]$, where

$$g\Big(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q\Big) = \mathbb{E}_{p(y \mid \boldsymbol{y}_q)}\Big[\max \boldsymbol{v}_{q+1} - \max \boldsymbol{v}_q\Big]. \tag{2.51}$$

Since $p(\boldsymbol{y}_{q+1}) = p(y \mid \boldsymbol{y}_q)p(\boldsymbol{y}_q)$, we may view the naïve estimator

$$\widetilde{G}(\boldsymbol{x} \mid \mathbf{X}_q) = \frac{1}{m}\sum_{i=1}^{m} \max \boldsymbol{\nu}_{q+1}^{(i)} - \max \boldsymbol{\nu}_q^{(i)}. \tag{2.52}$$

as the average of nested estimators

$$g\Big(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q = \boldsymbol{\gamma}_q\Big) \approx \frac{1}{m^*}\sum_{j=1}^{m^*} \max\Big\{\nu_1, \ldots, \nu_q, \nu^{(j)}\Big\} - \max \boldsymbol{\nu}_q, \tag{2.53}$$

where $m^* = 1$ and $\nu^{(j)}$ denotes the j-th realization of $y \mid \boldsymbol{y}_q = \boldsymbol{\gamma}_q$. By Theorem 2.13, it is clear that (2.52) may be improved by taking the limit where $m^* \to \infty$, namely

$$\widetilde{G}^*(\boldsymbol{x} \mid \mathbf{X}_q) = \frac{1}{m}\sum_{i=1}^{m} g\Big(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q = \boldsymbol{\gamma}_q^{(i)}\Big). \tag{2.54}$$

Supposing $\widetilde{G}$ is unbiased, it follows that $\widetilde{G}^*$ will be a reduced variance estimator of marginal acquisition function G and so for the Rao-Blackwellized gradient estimator

$$\nabla_{\mathbf{X}}\widetilde{G}^*(\boldsymbol{x} \mid \mathbf{X}_q) = \frac{1}{m}\sum_{i=1}^{m} \nabla_{\mathbf{X}} g\Big(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q = \boldsymbol{\gamma}_q^{(i)}\Big). \tag{2.55}$$

Of course, all of this hinges upon our ability to evaluate g. In what cases will g be

[5] Note that similar statements hold in parallel asynchronous cases.

analytic though? In answering this question, a trivial but useful identity is

$$\max(\boldsymbol{v}_{q+1}) - \max(\boldsymbol{v}_q) = \max\{0, v - \max \boldsymbol{v}_q\}, \tag{2.56}$$

which holds because $v - \max \boldsymbol{v}_q < 0$ implies $\max \boldsymbol{v}_{q+1} = \max \boldsymbol{v}_q$. Hence, we have

$$g\Big(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q = \boldsymbol{\gamma}_q\Big) = \mathbb{E}_{p(y \mid \boldsymbol{y}_q = \boldsymbol{\gamma}_q)}\Big[\max\Big\{0, v(y) - \max v(\boldsymbol{\gamma}_q)\Big\}\Big]. \tag{2.57}$$

If (2.57) looks familiar, that is because it is simply the expected improvement of the batch acquisition value itself, given $\boldsymbol{y}_q = \boldsymbol{\gamma}_q$. This connection (and others like it) exists because of the shared use of the max to measure a set's value.

As a specific example, consider the case of marginal Expected Improvement. Denoting the improvement threshold by α, it is easy to show that

$$\max\Big\{0, v_{\text{EI}}(y) - \max v_{\text{EI}}(\boldsymbol{\gamma}_q)\Big\} = \max\Big\{0, y - \max\{\alpha, \gamma_1, \ldots, \gamma_q\}\Big\}. \tag{2.58}$$

and, consequently, that

$$g_{\text{EI}}\Big(\boldsymbol{x} \mid \mathbf{X}_q, \boldsymbol{y}_q = \boldsymbol{\gamma}_q\Big) = \text{EI}\Big(\boldsymbol{x} \mid (\boldsymbol{x}_i, \gamma_i)_{i=1}^q\Big). \tag{2.59}$$

Hence, the change in Expected Improvement when adding a design $\boldsymbol{x}$ to a batch $\mathbf{X}_q$ is obtained by "fantasizing" what the Expected Improvement for querying $\boldsymbol{x}$ would be if we observing $\boldsymbol{y}_q = \boldsymbol{\gamma}_q$ and, then, integrating out $\boldsymbol{y}_q$.

When $y \mid \boldsymbol{y}_q$ is Gaussian, (2.57) admits a closed-form solution for each of the myopic, batch acquisition functions discussed in this chapter. In these cases, the analytic nature of g typically results in hybrid estimators whose gradients are significantly more robust to cases where sampled utility values are sparse.

2.6.1 Results

Results for this section stem from two separate sets of experiments: initial experiments following the setup described in Section 2.4.1 and a follow-up one tailored to highlight and clarify the benefits of Rao-Blackwellization (RB). In both cases, RB was used together with greedy batch selection as follows. After choosing an initial batch element $\boldsymbol{x}_1$, we generated m samples of the unknown outcome $y_1 = f(\boldsymbol{x}_1)$ and used these samples to construct an enhanced estimator (2.55). At each subsequent round of greedy maximization $j > 1$, a single realization of corresponding outcome y_j was drawn from each of the m distributions formed by conditioning on the different sample vectors.

Figure 2.7 presents the results for the first wave of experiments. These results show that RB estimators are at least as good as their naïve counterparts and, sometimes, significantly better. A seeming counterexample to this statement occurs when dealing with low-dimensional batches (top row of Figure 2.7). In subsequent experiments, however, we found that this issue reflected the small number of samples afforded to RB estimators $m = 16$. Moreover, we found that these experiment do not adequately

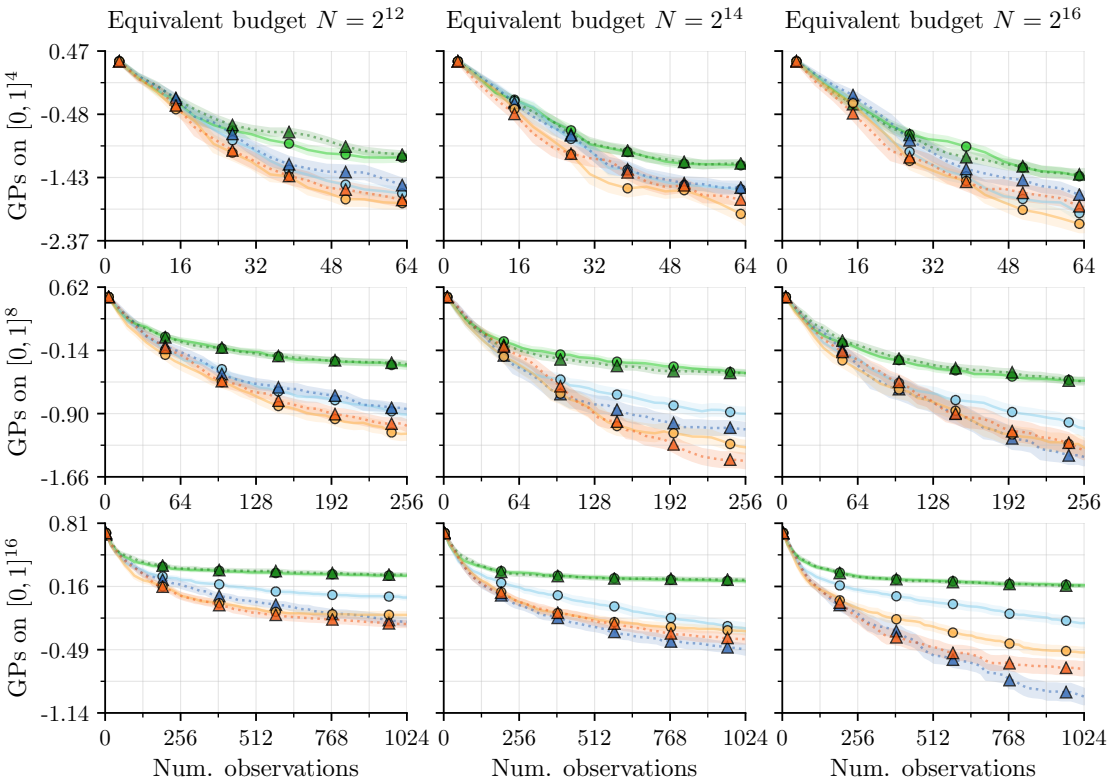

Figure 2.7: Comparison of BO performance (means and standard errors of log immediate regret over 32 trials) when greedily maximizing naïve and Rao-Blackwellized Monte Carlo estimators for EI. In all cases, batch-sizes were chosen to match with the design dimensionalies, i.e. $q = d$. Lighter shades denote the use of naïve estimators, while darker ones represent their Rao-Blackwellized counterparts. Colors indicate which algorithm was used to solve each round of greedy selection with Random Search in greens, CMA-ES in blues, and stochastic gradient ascent in orange and yellow.

characterize the advantages of Rao-Blackwellization for the simple reason that trials were not run long enough for sparse rewards to become an issue.

Figure 2.8, therefore, presents extended findings. We report performance on the popular Hartmann-6 test function with size $q = 8$ batches of synchronously evaluated queries, chosen using estimators consisting of $m = 64$ samples. As is clear from the plot, both methods perform virtually identically during the early phases of optimization. Towards the end, however, when rewards are sparse, RB variants' are seen to strongly outperform their basic counterparts. In additional experiments (not shown), these trends were found to be consistent across different optimizations problems and batch-sizes q, as well when switching from synchronous to asynchronous evaluations.

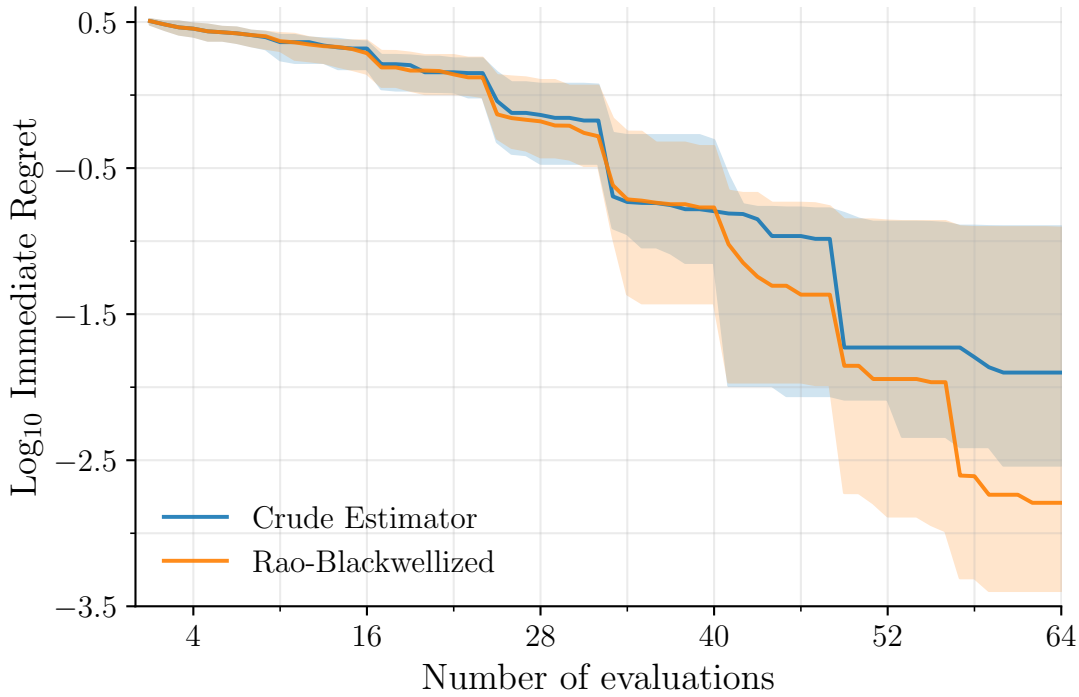

Figure 2.8: Extended results for BO performance on Hartmannn-6 (medians and interquartiles ranges of log immediate regret over 32 trials) when greedily maximizing crude and Rao-Blackwellized estimators for EI with $q = 8$. As seen on the right, RB approaches perform markedly better than the naïve baseline when dealing with (pathwise) sparse acquisition values.

2.7 Discussion

The purpose of this chapter has been to recount the decision-theoretic origins of Bayesian optimization and to discuss methods for maximizing acquisition functions. We began by seeing how the decision-making framework presented in Chapter 1 leads to a version of Bayesian optimization in which acquisition functions directly measure changes in the agent's expected utility. An intuitive recipe for this construction is as follows: (i) use a model to simulate outcomes, (ii) determine the agent's choice of incumbent given the simulated outcomes, (iii) compute the difference in expected utility between new and old incumbents, (iv) repeat the first three steps many times to integrate over possible outcomes. Querying strategies based on this process are often highly performant and can easily be adapted to different settings though appropriate choice of incumbent rule.

Of course, all of this hinges upon our ability to maximize these signals sufficiently well. Much of this chapter therefore focused on techniques for solving inner optimization problems. We first showed that Monte Carlo estimators of popular acquisition functions are typically pathwise differentiable. The core arguments behind these findings were previously known (Wang et al., 2016). Our role has simply been to demonstrate greater generality and to help motivate widespread adoption (Balandat

et al., 2020).

In practice, one often expedites the process of finding desirable solutions by exploiting parallel evaluations. Hence, we also investigated the matter of batch selection. Our contribution here has been to show that well-known acquisition functions (such as Expected Improvement and Probability of Improvement) are submodular. This result justifies the use of greedy algorithms, which drastically reduce the amount of work required to obtain high quality batches of queries. Further, we showed how Monte Carlo estimators for many of the same acquisition functions can be Rao-Blackwellized to enhance the performance of gradient-based, greedy batch selection. Overall, we hope to have equipped practitioners with the right tools to tackle commonly occurring types of inner optimization problems.

CHAPTER 3

Pathwise Conditioning of Gaussian Processes

In this final chapter, we focus on the driving force behind many practical applications of Bayesian decision-making, namely Gaussian processes (GPs). In machine learning, the narrative of GPs is dominated by talk of distributions (Rasmussen and Williams, 2006). This view is often helpful and convenient: a Gaussian process is a random function; however, seeing as we may trivially marginalize out arbitrary subsets of this function, we can simply focus on its behavior at a finite number of input locations. When dealing with regression and classification problems, this reduction simplifies discourse and expedites implementation by allowing us to work with joint distributions at training and test locations instead of random functions.

Model-based learning and prediction generally service broader goals. For example, when making decisions in the face of uncertainty, models enable us to simulate the consequences of our actions. Decision-making, then, amounts to optimizing the expectation of a simulated quantity of interest, such as a measure of utility. Be it for purposes of safety or for balancing trade-offs between long-term and short-term goals, it is crucial that these simulations faithfully portray both knowledge and uncertainty. Gaussian processes are known to make accurate, well-calibrated predictions and, therefore, stand as the model-of-choice in fields such as Bayesian optimization (Shahriari et al., 2015), uncertainty quantification (Bect et al., 2012), and model-based reinforcement learning (Deisenroth et al., 2015).

Unfortunately, marginal distributions and simulations do not always go hand in hand. When the quantity of interest is a function of a process value $f(\boldsymbol{x}_*)$ at an individual input location $\boldsymbol{x}_*$, its expectation can sometimes be obtained analytically. Conversely, when this quantity is a function of process values $\boldsymbol{f}_* = f(\mathbf{X}_*)$ at multiple

locations $\mathbf{X}_*$, its expectation is generally intractable. Rather than solving these integrals directly in terms of marginal distributions $p(\boldsymbol{f}_*)$, we therefore estimate them by averaging over many simulations of $\boldsymbol{f}_*$. Drawing $\boldsymbol{f}_*$ from $p(\boldsymbol{f}_*)$ takes $\mathcal{O}(*^3)$ time, where $* = |\mathbf{X}_*|$ is the number of input locations. Hence, distribution-based approaches to sampling $\boldsymbol{f}_*$ quickly become untenable as this number increases. In these cases, we may be better off thinking about GPs from a perspective that naturally lends itself to sampling

In the early 1970s, one such view surfaced in the then nascent field of geostatistics (Journel and Huijbregts, 1978; Chilès and Delfiner, 2012). Instead of emphasizing the statistical properties of Gaussian random variables, "conditioning by Kriging" encourages us to think in terms of the variables themselves. This chapter studies the broader implications of this paradigm shift to develop a general framework for conditioning Gaussian processes at the level of random functions. Formulating conditioning in terms of sample paths, rather than distributions, allows us to separate out the effect of the prior from that of the data. By leveraging this property, we can use *pathwise conditioning* to efficiently approximate function draws from GP posteriors. As we will see, working with sample paths enables us to simulate process values $\boldsymbol{f}_*$ in $\mathcal{O}(*)$ time and brings with it a host of additional benefits.

This chapter is organized as follows. Section 3.1 and Section 3.2 introduce pathwise conditioning of Gaussian random vectors and processes, respectively. Section 3.3 surveys strategies for approximating function draws from GP priors, while Section 3.4 discusses methods for mapping from prior to posterior random variables. Section 3.5 studies the behavior of errors introduced by different approximation techniques, and Section 3.6 complements this theory by exploring several applications.

Notation By way of example, we denote matrices as $\mathbf{A}$ and vectors as $\boldsymbol{a}$. We write $\boldsymbol{x} = \boldsymbol{a} \oplus \boldsymbol{b}$ for the concatenation of vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. Throughout, we use $|\cdot|$ to denote the cardinality of sets and dimensionality of vectors. When dealing with covariance matrices $\boldsymbol{\Sigma} = \mathrm{Cov}(\boldsymbol{x}, \boldsymbol{x})$, we use subscripts to identify corresponding blocks. For example, $\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} = \mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b})$. As shorthand, we denote the evaluation of a function $f : \mathcal{X} \to \mathbb{R}$ at a finite set of locations $\mathbf{X}_* \subset \mathcal{X}$ by the vector $\boldsymbol{f}_*$. Putting these together, when dealing with random variables $\boldsymbol{f}_* = f(\mathbf{X}_*)$ and $\boldsymbol{f}_n = f(\mathbf{X}_n)$, we write $\mathbf{K}_{*,n} = \mathrm{Cov}(\boldsymbol{f}_*, \boldsymbol{f}_n)$.

3.1 Gaussian distributions and random vectors

A random vector $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ is said to be Gaussian if there exists a matrix $\mathbf{L}$ and vector $\boldsymbol{\mu}$ for which

$$\boldsymbol{x} \overset{\mathrm{d}}{=} \boldsymbol{\mu} + \mathbf{L}\boldsymbol{\zeta} \qquad\qquad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3.1}$$

where $\mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard (multivariate) normal distribution, whose probability density function is given below. Each such distribution is uniquely identified by its

first two moments: its mean $\boldsymbol{\mu} = \mathbb{E}(\boldsymbol{x})$ and its covariance $\boldsymbol{\Sigma} = \mathbb{E}[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top]$. Assuming it exists, the corresponding density function is defined as

$$p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right). \tag{3.2}$$

The representation of $\boldsymbol{x}$ given by (3.1) is commonly referred to as its *location-scale* form and stands as the most widely used method for generating Gaussian random vectors. Since $\boldsymbol{\zeta}$ has identity covariance, any matrix square root of $\boldsymbol{\Sigma}$, such as its Cholesky factor $\mathbf{L}$ with $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$, may be used to draw $\boldsymbol{x}$ as prescribed by (3.1).

Here, we focus on multivariate cases $n > 1$ and investigate different ways of reasoning about random variables $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$ for non-trivial partitions $\boldsymbol{x} = \boldsymbol{a} \oplus \boldsymbol{b}$.

3.1.1 Distributional conditioning

The quintessential approach to deriving the distribution of $\boldsymbol{a}$ subject to the condition $\boldsymbol{b} = \boldsymbol{\beta}$ begins by employing the usual set of matrix identities to factor $p(\boldsymbol{b})$ from $p(\boldsymbol{a}, \boldsymbol{b})$. Applying Bayes' rule, $p(\boldsymbol{b})$ then cancels out and $p(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta})$ is identified as the remaining term—namely, the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{a}|\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}|\boldsymbol{\beta}})$ with moments

$$\boldsymbol{\mu}_{\boldsymbol{a}|\boldsymbol{\beta}} = \boldsymbol{\mu}_{\boldsymbol{a}} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}(\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{b}}) \qquad \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}|\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}} - \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{a}}. \tag{3.3}$$

Having obtained this conditional distribution, we can now generate $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$ by computing a matrix square root of $\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}|\boldsymbol{\beta}}$ and constructing a location-scale transform (3.1).

Due to their emphasis of conditional distributions, we refer to methods that represent or generate a random variable $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$ by way of $p(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta})$ as being *distributional* in kind. This approach to conditioning is not only standard, but particularly natural when quantities of interest may be derived analytically from $p(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta})$. Many quantities, such as expectations of nonlinear functions, cannot be deduced analytically from $p(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta})$ alone, however. In these cases we must instead work with realizations of $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$. Since the cost of obtaining a matrix square root of $\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}|\boldsymbol{\beta}}$ scales cubically in $|\boldsymbol{a}|$, distributional approaches to evaluating these quantities struggle to accommodate high-dimensional random vectors. To address this issue, we now consider Gaussian conditioning in another light.

3.1.2 Pathwise conditioning

Instead of taking a *distribution-first* stance on Gaussian conditionals, we may think of conditioning directly in terms of random variables. In this *variable-first* paradigm, we will explicitly map samples from the prior to draws from a posterior and let the corresponding relationship between distributions follow implicitly. Throughout this work, we investigate this notion of *pathwise conditioning* through the lens of the following result.

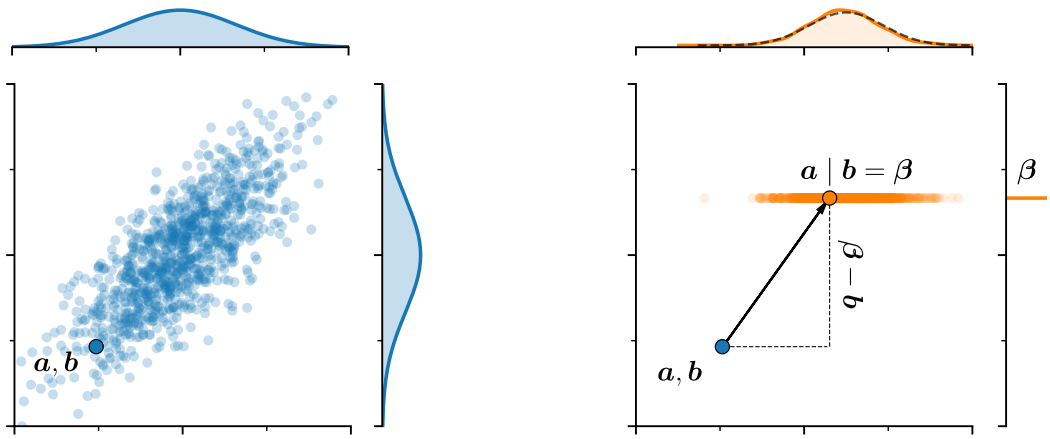

Figure 3.1: Visualization of Matheron's update rule for a bivariate normal distribution with correlation coefficient $\rho = 0.75$. *Left:* Draws from $p(\boldsymbol{a}, \boldsymbol{b})$ are shown alongside the marginal distributions of $\boldsymbol{a}$ and $\boldsymbol{b}$. *Right:* Theorem 3.14 is used to update samples shown on the left subject to the condition $\boldsymbol{b} = \boldsymbol{\beta}$. This process is illustrated in full for a particular draw. *Top right:* the empirical distribution of the update samples is compared with $p(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta})$.

Theorem 3.14 (Matheron's Update Rule)**.** *Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be jointly Gaussian, centered random variables. Then, the random variable $\boldsymbol{a}$ conditional on $\boldsymbol{b} = \boldsymbol{\beta}$ may be expressed as*

$$(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) \stackrel{\mathrm{d}}{=} \boldsymbol{a} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} (\boldsymbol{\beta} - \boldsymbol{b}). \tag{3.4}$$

Proof Comparing the mean and covariance on both sides immediately affirms the result

$$\begin{aligned} \mathbb{E}\Big(\boldsymbol{a} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} (\boldsymbol{\beta} - \boldsymbol{b})\Big) &= \boldsymbol{\mu}_{\boldsymbol{a}} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} (\boldsymbol{\beta} - \boldsymbol{\mu}_{\boldsymbol{b}}) = \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) \\ \mathrm{Cov}\Big(\boldsymbol{a} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} (\boldsymbol{\beta} - \boldsymbol{b})\Big) &= \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{a}} - 2\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{a}} \\ &= \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{a}} - \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1} \boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{a}} = \mathrm{Cov}(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) \end{aligned} \tag{3.5}$$

■

This observation leads to a straightforward, alternative recipe for generating $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$: first, draw $\boldsymbol{a}, \boldsymbol{b} \sim p(\boldsymbol{a}, \boldsymbol{b})$; then, update this sample according to (3.4). Compared to the location-scale approach discussed in Section 3.1.1, a key difference is that we now sample *before* conditioning, rather than after. Figure 3.1 visualizes the deterministic process of updating previously generated draws from the prior subject to the condition $\boldsymbol{b} = \boldsymbol{\beta}$.

At first glance, Matheron's update rule may seem more like an interesting footnote than a valuable tool. Indeed, the conventional strategy for sampling $\boldsymbol{a}, \boldsymbol{b}$ (which requires us to take a matrix square root of $\boldsymbol{\Sigma}$) is more expensive than that for generating $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$. We will discuss this matter in detail in the later sections. For now, however, let us strengthen our intuition by delving deeper into this theorem's

function-analytic origins.

3.1.3 Deriving pathwise conditioning via conditional expectations

Here, we overview the precise formalism that gives rise to the pathwise approach to conditioning Gaussian random variables and show how to *derive* this result from first principles. Throughout this section, we take $\boldsymbol{a} \in \mathbb{R}^m$ and $\boldsymbol{b} \in \mathbb{R}^n$ to be centered random vectors defined on the same probability space.

The core idea is to decompose $\boldsymbol{a}$ as the sum of two independent terms—one that depends on $\boldsymbol{b}$ and one that does not—and represent $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$ by conditioning both terms on $\boldsymbol{b} = \boldsymbol{\beta}$. We first prove that conditioning this additive decomposition of $\boldsymbol{a}$ is simple and intuitive.

Lemma 3.15. *Consider three random vectors $\boldsymbol{a} \in \mathbb{R}^m$, $\boldsymbol{b} \in \mathbb{R}^n$, $\boldsymbol{c} \in \mathbb{R}^m$ such that*

$$\boldsymbol{a} \stackrel{\mathrm{d}}{=} f(\boldsymbol{b}) + \boldsymbol{c}, \tag{3.6}$$

where f is a measurable function of $\boldsymbol{b}$ and where $\boldsymbol{b}$ is independent of $\boldsymbol{c}$. Then,

$$\Big(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}\Big) \stackrel{\mathrm{d}}{=} f(\boldsymbol{\beta}) + \boldsymbol{c}. \tag{3.7}$$

Proof Let $\pi_{\boldsymbol{x}}$ denote the distribution of a generic random variable $\boldsymbol{x}$. Further, let $\pi_{\boldsymbol{a}|\boldsymbol{b}}(\ \cdot\ \mid\ \cdot\ )$ be the (regular) conditional probability measure given by disintegration[1] of $(\boldsymbol{a}, \boldsymbol{b})$, such that

$$\int_B \pi_{\boldsymbol{a}|\boldsymbol{b}}(A \mid \boldsymbol{\beta})\, \mathrm{d}\pi_{\boldsymbol{b}}(\boldsymbol{\beta}) = \mathbb{P}(\boldsymbol{a} \in A, \boldsymbol{b} \in B) \tag{3.8}$$

for measurable sets $A \subseteq \mathbb{R}^m$, $B \subseteq \mathbb{R}^n$. When $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$ is represented per (3.7), we have

$$\begin{aligned}
\int_B \mathbb{P}(f(\boldsymbol{\beta}) + \boldsymbol{c} \in A)\, \mathrm{d}\pi_{\boldsymbol{b}}(\boldsymbol{\beta}) &= \int_B \Big(\int_{\mathbb{R}^m} \mathbb{1}_{\{f(\boldsymbol{\beta}) + \boldsymbol{\varsigma} \in A)\}}\, \mathrm{d}\pi_{\boldsymbol{c}}(\boldsymbol{\varsigma}) \Big)\, \mathrm{d}\pi_{\boldsymbol{b}}(\boldsymbol{\beta}) \\
&= \int_{\mathbb{R}^m \times \mathbb{R}^n} \mathbb{1}_{\{f(\boldsymbol{\beta}) + \boldsymbol{\varsigma} \in A, \boldsymbol{\beta} \in B)\}}\, \mathrm{d}\pi_{\boldsymbol{b},\boldsymbol{c}}(\boldsymbol{\beta}, \boldsymbol{\varsigma}) \\
&= \mathbb{P}(f(\boldsymbol{b}) + \boldsymbol{c} \in A, \boldsymbol{b} \in B) = \mathbb{P}(\boldsymbol{a} \in A, \boldsymbol{b} \in B),
\end{aligned} \tag{3.9}$$

where we have begun by expressing probabilities as integrals of indicator functions, before using Tonelli's theorem and independence to express the iterated integral as the double integral over the joint probability measure $\pi_{\boldsymbol{b},\boldsymbol{c}}(\boldsymbol{\beta}, \boldsymbol{\varsigma})$. Comparing the left-hand sides of (3.8) and (3.9) affirms the claim. ■

In words, Lemma 3.15 tells us that for suitably chosen functions f, the act of conditioning $\boldsymbol{a}$ on $\boldsymbol{b} = \boldsymbol{\beta}$ amounts to adding a random variable $\boldsymbol{c}$ to a deterministic

[1] See discussion and details on disintegration by Chang and Pollard (1997) and Kallenberg (2006).

transformation $f(\boldsymbol{\beta})$ of the outcome $\boldsymbol{\beta}$. For this statement to hold, we require the *residual* $\boldsymbol{c} = \boldsymbol{a} - f(\boldsymbol{b})$ induced by f to be statistically independent of $\boldsymbol{b}$. Fortunately, such a function f is well-known in the special case of jointly Gaussian random variables—namely, the *conditional expectation* $f : \boldsymbol{b} \mapsto \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})$.

For square-integrable random variables, the conditional expectation of $\boldsymbol{a}$ given $\boldsymbol{b}$ is defined as the (almost surely) unique solution to the minimization problem

$$\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b}) = \arg\min_{f \in \mathcal{F}} \mathbb{E}\|\boldsymbol{a} - f(\boldsymbol{b})\|^2, \tag{3.10}$$

where $\mathcal{F}$ denotes the set of all Borel-measurable functions $f : \mathbb{R}^n \to \mathbb{R}^m$ (Kallenberg, 2006, Chapter 6). Put simply, $\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})$ is the measurable function of $\boldsymbol{b}$ that best predicts $\boldsymbol{a}$ in the sense of minimizing the mean-square error (3.10). This characterization of the conditional expectation is equivalent to defining it as the orthogonal projection of $\boldsymbol{a}$ onto the σ-algebra generated by $\boldsymbol{b}$, denoted $\sigma(\boldsymbol{b})$. Consequently, a necessary and sufficient condition for $\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b}) \in \mathcal{F}$ to uniquely solve (3.10) is that the residual $\boldsymbol{c} = \boldsymbol{a} - \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})$ be orthogonal to all $\sigma(\boldsymbol{b})$-measurable random variables (Luenberger, 1997, page 50). Here, *orthogonality* can be understood as the absence of correlation, which (for jointly Gaussian random variables) implies independence. As a result, we may satisfy the assumptions of Lemma 3.15 by writing

$$\boldsymbol{a} = \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b}) + \boldsymbol{c}, \tag{3.11}$$

such that $\boldsymbol{a}$ decomposes into a function of $\boldsymbol{b}$ and an independent variable $\boldsymbol{c} = \boldsymbol{a} - \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})$.

As a final remark, we may also use these principles to concisely derive the conditional expectation for jointly Gaussian random variables. For now, suppose that the conditional expectation is a linear function of $\boldsymbol{b}$, i.e. that $\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b}) = \mathbf{S}\boldsymbol{b}$ for some matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$. To satisfy the orthogonality condition of (3.10), we require $\mathrm{Cov}(\boldsymbol{a} - \mathbf{S}\boldsymbol{b}, \boldsymbol{b}) = \mathbf{0}$, implying that $\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}} - \mathbf{S}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}} = \mathbf{0}$. Rearranging terms and solving for $\mathbf{S}$ gives $\mathbf{S} = \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{b}$. With this expression in hand, to show that linearity was assumed without loss of generality, write $\boldsymbol{a} = \mathbf{S}\boldsymbol{b} + \boldsymbol{a} - \mathbf{S}\boldsymbol{b}$, which we may express as as $\boldsymbol{a} = \mathbf{S}\boldsymbol{b} + \boldsymbol{c}$. Taking the conditional expectation of both sides, we may directly calculate $\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})$ by writing

$$\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b}) = \mathbb{E}(\mathbf{S}\boldsymbol{b} + \boldsymbol{c} \mid \boldsymbol{b}) = \underbrace{\mathbb{E}(\mathbf{S}\boldsymbol{b} \mid \boldsymbol{b})}_{\mathbf{S}\boldsymbol{b}} + \underbrace{\mathbb{E}(\boldsymbol{c})}_{\mathbf{0}} = \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{b}, \tag{3.12}$$

where we have used linearity of conditional expectation, followed by independence of $\boldsymbol{c}$ and $\boldsymbol{b}$ to go from the second to the third expression. We now revisit Theorem 3.14.

Theorem 1 (Matheron's Update Rule)**.** *Let $\boldsymbol{a}$ and $\boldsymbol{b}$ be jointly Gaussian, centered random vectors. Then, the random vector $\boldsymbol{a}$ conditional on $\boldsymbol{b} = \boldsymbol{\beta}$ may be expressed as*

$$(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) \overset{\mathrm{d}}{=} \boldsymbol{a} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}(\boldsymbol{\beta} - \boldsymbol{b}). \tag{3.4}$$

Proof With $\boldsymbol{c} = \boldsymbol{a} - \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{b}$, begin by writing

$$\boldsymbol{a} = \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b}) + (\boldsymbol{a} - \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})) = \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{b} + \boldsymbol{c}. \tag{3.13}$$

Since $\boldsymbol{b}$ and $\boldsymbol{c}$ are jointly Gaussian but uncorrelated, it follows that they are independent. Setting $f(\boldsymbol{b}) = \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{b}$ and using Lemma 3.15 to condition both sides on $\boldsymbol{b} = \boldsymbol{\beta}$ gives

$$(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}) \overset{\mathrm{d}}{=} \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{\beta} + \left(\boldsymbol{a} - \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}\boldsymbol{b}\right) = \boldsymbol{a} + \boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}(\boldsymbol{\beta} - \boldsymbol{b}). \tag{3.14}$$

Hence, the claim follows. ■

In summary, we have shown that Matheron's update rule (Theorem 3.14) is a direct consequence of the fact that a Gaussian random variable $\boldsymbol{a}$ conditioned on the outcome $\boldsymbol{\beta}$ of another (jointly) Gaussian random variable $\boldsymbol{b}$ may be expressed as the sum of two independent terms: the conditional expectation $\mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}$ and the residual $\boldsymbol{c} = \boldsymbol{a} - \mathbb{E}(\boldsymbol{a} \mid \boldsymbol{b})$. Rearranging these terms gives (3.4).

With these ideas in mind, we are now ready to explore this work's primary theme: Matheron's update rule enables us to decompose $\boldsymbol{a} \mid \boldsymbol{b} = \boldsymbol{\beta}$ into the prior random variable $\boldsymbol{a}$ and a data-driven update $\boldsymbol{\Sigma}_{\boldsymbol{a},\boldsymbol{b}}\boldsymbol{\Sigma}_{\boldsymbol{b},\boldsymbol{b}}^{-1}(\boldsymbol{\beta} - \boldsymbol{b})$ that explicitly corrects for the error in the coinciding value of $\boldsymbol{b}$ given the condition $\boldsymbol{b} = \boldsymbol{\beta}$. Hence, Theorem 3.14 provides an explicit means of separating out the influence of the prior from that of the data. We now proceed to investigate the implications of pathwise conditioning for Gaussian processes.

3.2 Gaussian processes and random functions

A Gaussian process (GP) is a random function $f : \mathcal{X} \to \mathbb{R}$, such that, for any finite collection of points $\mathbf{X} \subset \mathcal{X}$, the random vector $\boldsymbol{f} = f(\mathbf{X})$ follows a Gaussian distribution. Such a process is uniquely identified by a mean function $\mu : \mathcal{X} \to \mathbb{R}$ and a positive semi-definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Hence, if $f \sim \mathcal{GP}(\mu, k)$, then $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K})$ is multivariate normal with mean $\boldsymbol{\mu} = \mu(\mathbf{X})$ and covariance $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$.

Throughout this section, we investigate different ways of reasoning about the random variable $\boldsymbol{f}_* \mid \boldsymbol{f}_n = \boldsymbol{y}$ for some non-trivial partition $\boldsymbol{f} = \boldsymbol{f}_n \oplus \boldsymbol{f}_*$. Here, $\boldsymbol{f}_n = f(\mathbf{X}_n)$ are process values at a set of training locations $\mathbf{X}_n \subset \mathbf{X}$ where we would like to introduce a condition $\boldsymbol{f}_n = \boldsymbol{y}$, while $\boldsymbol{f}_* = f(\mathbf{X}_*)$ are process values at a set of test locations $\mathbf{X}_* \subset \mathbf{X}$ where we would like to obtain a random variable $\boldsymbol{f}_* \mid \boldsymbol{f}_n = \boldsymbol{y}$. Mirroring Section 3.1, we begin by reviewing distributional conditioning, before examining its pathwise counterpart.

3.2.1 Distributional conditioning

As in finite-dimensional cases, we may obtain $\boldsymbol{f}_* \mid \boldsymbol{y}$ by first finding its conditional distribution. Since process values $(\boldsymbol{f}_n, \boldsymbol{f}_*)$ are defined as jointly Gaussian, this procedure closely resembles that of Section 3.1.1: we factor out the marginal distribution of $\boldsymbol{f}_n$ from the joint distribution $p(\boldsymbol{f}_n, \boldsymbol{f}_*)$ and, upon canceling, identify the remaining distribution as $p(\boldsymbol{f}_* \mid \boldsymbol{y})$. Having done so, we find that the conditional distribution is the Gaussian $\mathcal{N}(\boldsymbol{\mu}_{*|\boldsymbol{y}}, \mathbf{K}_{*,*|\boldsymbol{y}})$ with moments

$$\boldsymbol{\mu}_{*|\boldsymbol{y}} = \boldsymbol{\mu}_* + \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}(\boldsymbol{y} - \boldsymbol{\mu}_n) \qquad \mathbf{K}_{*,*|\boldsymbol{y}} = \mathbf{K}_{*,*} - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{K}_{n,*}. \tag{3.15}$$

As before, we may now generate $\boldsymbol{f}_* \mid \boldsymbol{y}$ via a location-scale transform in $\mathcal{O}(*^3)$ time.

This strategy for sampling Gaussian process posteriors is subtly different from the one given in Section 3.1.1. A Gaussian process is a random function, and conditioning on $\boldsymbol{f}_n = \boldsymbol{y}$ does not change this fact. Unfortunately, (conditional) distributions over infinite-dimensional objects can be difficult to manipulate in practice. Distributional approaches, therefore, focus on finite-dimensional subsets $\boldsymbol{f} = \boldsymbol{f}_n \oplus \boldsymbol{f}_*$, while marginalizing out the remaining process values. Doing so allows them to perfectly describe the random variable $\boldsymbol{f}_* \mid \boldsymbol{y}$ in terms of its mean and covariance (3.15).

When it comes to sampling $\boldsymbol{f}_* \mid \boldsymbol{y}$, however, these approaches have clear limitations. As discussed previously, a key issue is that their $\mathcal{O}(*^3)$ time complexity restricts them to problems that only require us to jointly simulate process values at a manageable number of test locations (up to several thousand). In some senses, this condition is fairly generous. After all, we are often only asked to generate a handful of process values at a time. Still, other problems effectively require us to realize $f \mid \boldsymbol{y}$ in its entirety. Similar issues arise when $\mathbf{X}_*$ is not defined in advance, such as when gradient information is used to adaptively determine the locations at which to jointly sample the posterior. In these cases and more, we would ideally like to sample actual functions that we can efficiently evaluate and automatically differentiate at arbitrary test locations. To this end, we now examine the direct approach to conditioning draws of $f \sim \mathcal{GP}(\mu, k)$.

3.2.2 Pathwise Conditioning

Examining the pathwise update given by Theorem 3.14, it is natural to suspect that an analogous statement holds for Gaussian processes. A quick check confirms this hypothesis.

Corollary 3.17. *For a Gaussian process $f \sim \mathcal{GP}(\mu, k)$ with marginal $\boldsymbol{f}_n = f(\mathbf{X}_n)$, the process conditioned on $\boldsymbol{f}_n = \boldsymbol{y}$ may be expressed as*

$$\underbrace{(f \mid \boldsymbol{y})(\,\cdot\,)}_{\text{conditional}} \overset{\text{d}}{=} \underbrace{f(\,\cdot\,)}_{\text{prior}} + \underbrace{k(\,\cdot\,, \mathbf{X}_n)\mathbf{K}_{n,n}^{-1}(\boldsymbol{y} - \boldsymbol{f}_n)}_{\text{update}}. \tag{3.16}$$

Proof Follows by applying Theorem 3.14 to an arbitrary set of locations. ■

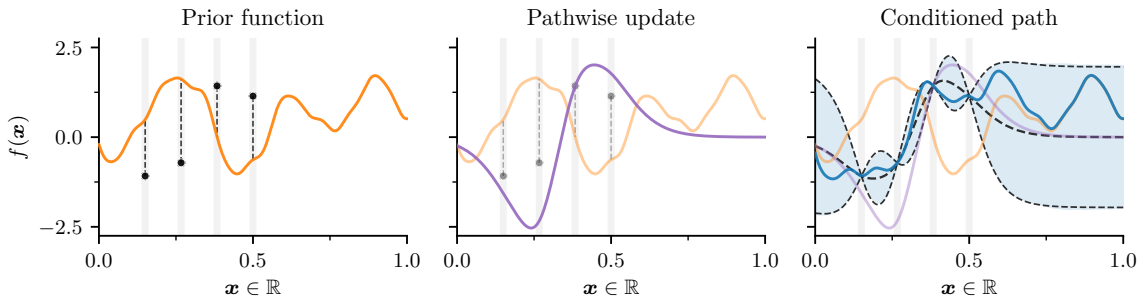

Figure 3.2: Visual guide for pathwise conditioning of Gaussian processes. *Left:* The residual $\boldsymbol{y} - \boldsymbol{f}_n$ (dashed black) of a draw $f \sim \mathcal{GP}(0, k)$, shown in orange, given observations $\boldsymbol{y}$ (black). *Middle:* A pathwise update (purple) is constructed in accordance with Corollary 3.17. *Right:* Prior and update are combined to represent conditional (blue). Empirical moments (light blue) of 10^5 conditioned paths are compared with those of the model (dashed black). The sample average, which matches the posterior mean, has been omitted for clarity.

Figure 3.2 acts a visual guide to Corollary 3.17. From left to right, we begin by generating a realization of $f \sim \mathcal{GP}(\mu, k)$ using methods that will soon be introduced in Section 3.3. Having obtained a sample path, we then use the pathwise update (3.16) to define a function $k(\,\cdot\,, \mathbf{X}_n)\mathbf{K}_{n,n}^{-1}(\boldsymbol{y} - \boldsymbol{f}_n)$ to account for the residual $\boldsymbol{y} - \boldsymbol{f}_n$. Adding these two functions together produces a draw from a GP posterior, the behavior of which is shown on the right. Whereas distributionally conditioning on $\boldsymbol{f}_n = \boldsymbol{y}$ in (3.15) tells us how the GP's statistic properties change, pathwise conditioning (3.16) tells us what happens to individual sample paths. This paradigm shift echoes the running theme: Gaussian (process) conditionals can be directly viewed in terms of random variables. The power of Corollary 3.17 is that it impacts *how* we think about Gaussian process posteriors and, therefore, *what* we do with them.

Having said this, there are several hurdles that we must overcome in order to use the pathwise update (3.16) in the real world. First, we are typically unable to practically sample functions $f \sim \mathcal{GP}(\mu, k)$ from (non-degenerate) Gaussian process priors exactly. A Gaussian process can generally be written as a linear combination of elementary *basis functions*. When the requisite number of basis functions is infinite, however, evaluating this linear combination is usually impossible. In Section 3.3, we will therefore investigate different ways of approximating $f(\,\cdot\,)$ using a finite number of operations.

Second, we incur $\mathcal{O}(n^3)$ time complexity when naïvely carrying out (3.16), due to the need to solve the linear system of equations $\mathbf{K}_{n,n}\boldsymbol{v} = \boldsymbol{y} - \boldsymbol{f}_n$ for a vector $\boldsymbol{v} \in \mathbb{R}^n$, such that

$$(f \mid \boldsymbol{y})(\,\cdot\,) \overset{\mathrm{d}}{=} f(\,\cdot\,) + \underbrace{\sum_{i=1}^{n} v_i k(\,\cdot\,, \boldsymbol{x}_i)}_{n\text{-dimensional basis}}. \tag{3.17}$$

Here, we have re-expressed the matrix-vector product in (3.16) as an expansion with respect to the canonical basis functions $k(\,\cdot\,, \boldsymbol{x}_i)$ centered at training locations $\boldsymbol{x}_i \in \mathbf{X}_n$. For large training sets $(\boldsymbol{x}_i, y_i)_{i=1}^n$, direct application of (3.16) may prove

prohibitively expensive. By the same token, the stated pathwise update does not hold when outcomes $\boldsymbol{y}$ are not defined as realizations of process values $\boldsymbol{f}_n$. In Section 3.4, we will consider various means of resolving these challenges and ones like them.

3.2.3 Historical remarks

Prior to continuing, we pause to reflect on the historical developments that have paved the way for this work. In a 2005 tribute to geostatistics pioneer Georges Matheron, Chilès and Lantuéjoul (2005) comment 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].

As if to echo this very sentiment, Doucet (2010) begins a much appreciated technical note on the subject of Theorem 3.14 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.

The presiding opinion, therefore, seems to be that Matheron's update rule is *too simple* to warrant extended study. Indeed, Theorem 3.14 is exceedingly straightforward to verify. As is often the case, however, this result is harder to discover if one is not already aware of its existence. This dilemma may help to explain why Matheron's update rule is absent from standard machine learning texts. By deriving this result from first principles in Section 3.1.3, we hope to encourage fellow researchers to explore the strengths (and weaknesses) of the pathwise viewpoint espoused here.

We are not the first to have realized the practical implications of pathwise conditioning for GPs. Corollary 3.17 is relatively well-known in geostatistics (Journel and Huijbregts, 1978; de Fouquet, 1994; Emery, 2007; Chilès and Delfiner, 2012). Similarly, Oliver (1996) discusses Matheron's update rule for Gaussian likelihoods (Section 3.4.1). Along the same lines, closely related ideas were rediscovered in the 1990s with applications to astrophysics. In particular, Hoffman and Ribak (1991) propose the use of spectral approximations to stationary priors (Section 3.3.2) in conjunction with canonical pathwise updates (3.17).

Nevertheless, these formulae are seldom seen in machine learning. We hope to systematically organize these findings (along with our own) and communicate them to a general audience of theorists and practitioners alike. The following sections therefore catalog various notable approaches to representing Gaussian process priors and pathwise updates.

3.3 Sampling functions from GP priors

The pathwise representation of GP posteriors described in the Section 3.2.2 allows us to represent $f \mid \boldsymbol{y}$ by transforming a draw of $f \sim \mathcal{GP}(0, k)$. When interpreted as a generative strategy, this approach to sampling can only be deemed *efficient* if the tasks of realizing the prior and performing the update both scale favorably in the total number of locations $|\mathbf{X}| = |\mathbf{X}_n| + |\mathbf{X}_*|$. Half of the battle is, therefore, to obtain faithful but affordable draws of f. Fortunately, GP priors often exhibit convenient mathematical properties not present in their posteriors, which can be utilized to sample them efficiently.

We focus on methods for generating random *functions* that we may evaluate at arbitrary locations $\boldsymbol{x} \in \mathcal{X}$ in $\mathcal{O}(1)$ time and whose marginal distributions approximate those of $f \sim \mathcal{GP}(0, k)$. Conceptually, techniques discussed throughout this section will approximate GP priors as random linear combinations of suitably chosen basis functions $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_\ell)$. Specifically, we will focus on *Bayesian linear models* with Gaussian random weights

$$\tilde{f}(\ \cdot\ ) = \sum_{i=1}^{\ell} w_i \phi_i(\ \cdot\ ) \qquad\qquad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{w}}), \tag{3.18}$$

where the covariance of weights $\boldsymbol{w}$ will vary by case. Notice that, for any finite collection of points $\mathbf{X} \subset \mathcal{X}$, the random vector $\tilde{\boldsymbol{f}} = \tilde{f}(\mathbf{X})$ follows the Gaussian distribution $\mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}\boldsymbol{\Sigma}_{\boldsymbol{w}}\boldsymbol{\Phi}^\top)$, where $\boldsymbol{\Phi} = \boldsymbol{\phi}(\mathbf{X})$ is a $|\mathbf{X}| \times \ell$ matrix of features. By design then, $\tilde{f}$ is a Gaussian process. Rasmussen and Williams (2006) refer to (3.18) as the *weight-space* view of GPs.

From this perspective, the task of efficiently sampling the prior $\tilde{f}$ reduces to one of generating random weights $\boldsymbol{w}$. In practice, $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ is typically diagonal, thereby enabling us to sample $\tilde{f}$ in $\mathcal{O}(\ell)$ time. We stress that, for any draw of $\boldsymbol{w}$, the corresponding realization of $\tilde{f}$ is simply a deterministic function. In particular, we incur $\mathcal{O}(1)$ cost for evaluating $\tilde{f}(\boldsymbol{x})$ and may readily differentiate this term with respect to $\boldsymbol{x}$ (or other parameters of interest).

Below, we review popular strategies for obtaining Bayesian linear models such that $\tilde{f} \overset{\mathrm{d}}{\approx} f$. Our presentation is intended to communicate different angles for attacking this problem and is by no means exhaustive. To set the scene for these approaches, we begin by recounting some properties of the gold standard: location-scale methods.

3.3.1 Location-scale transformations

Location-scale methods (3.1) are the most widely used approach for generating Gaussian random vectors. These generative strategies are *exact* (up to machine precision). Given locations $\mathbf{X}$, we may simulate $\boldsymbol{f} = f(\mathbf{X})$ in location-scale fashion

$$f(\mathbf{X}) \overset{\mathrm{d}}{=} \mathbf{K}^{1/2}\boldsymbol{\zeta} \qquad\qquad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{3.19}$$

by multiplying a square root covariance matrix $\mathbf{K}^{1/2}$ by a standard normal vector $\boldsymbol{\zeta}$.

While (3.19) rightfully stands as the method of choice for many problems, it is not without shortcoming. Chief among these issues is the fact that algorithms for obtaining a matrix square root of $\mathbf{K}$ scale cubically in $|\mathbf{X}|$. In most cases, this limits the use of location-scale approaches to cases where the length of the desired Gaussian random vector is manageable (up to several thousand). This overhead can be interpreted to mean that we incur $\mathcal{O}(i^2)$ cost for realizing the i-th element of $\boldsymbol{f}$, which leads us to our second issue: reusing a draw of $\boldsymbol{f}_n$ to efficiently generate the remainder of $\boldsymbol{f} = \boldsymbol{f}_n \oplus \boldsymbol{f}_*$ requires us to sample from the conditional distribution

$$\boldsymbol{f}_* \mid \boldsymbol{f}_n \sim \mathcal{N}\Big(\boldsymbol{\mu}_* + \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}(\boldsymbol{f}_n - \boldsymbol{\mu}_n), \mathbf{K}_{*,*} - \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-1}\mathbf{K}_{n,*}\Big). \tag{3.20}$$

Despite matching asymptotic costs, iterative approaches to sampling $\boldsymbol{f}$ are substantially slower than simultaneous ones. In applied settings, however, test locations $\mathbf{X}_*$ are often determined adaptively, forcing location-scale-based methods for generating $\boldsymbol{f}$ to repeatedly compute (3.20). Further refining this predicament, we arrive at a final challenge: pathwise derivatives.

Differentiation is a linear operation. The gradient of a Gaussian process f with respect to a location $\boldsymbol{x}$ is, therefore, another Gaussian process f'. By construction, these GPs are correlated. Using gradient information to maneuver along a sample path—for example, to identify its extrema—therefore requires us to re-condition both processes on the realized values of $f(\boldsymbol{x})$ and $f'(\boldsymbol{x})$ at each successive step of gradient descent.

Prior to continuing, it is worth noting that the limitations of location-scale methods can be avoided in certain cases. In particular, the otherwise cubic costs for computing a square root in (3.19) can be dramatically reduced by exploiting structural assumptions regarding covariance matrices $\mathbf{K}$. Well-known examples of structured matrices include banded and sparse ones in the context of one-dimensional Gaussian processes and Gauss-Markov random fields (Rue and Held, 2005; Durrande et al., 2019; Loper et al., 2020), block-Toeplitz Toeplitz-block ones when evaluating stationary product kernels on regularly-spaced grids $\mathbf{X} \subset \mathcal{X}$ (Zimmerman, 1989; Wood and Chan, 1994; Dietrich and Newsam, 1997), and kernel-interpolation-based ones (Wilson and Nickisch, 2015; Pleiss et al., 2018). When the task at hand permits their usage, these methods are highly effective.

The following sections survey different approaches to overcoming the challenges put forth above by approximating Gaussian process priors as finite-dimensional Bayesian linear models.

3.3.2 Stationary covariances

Stationary covariance functions $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x} - \boldsymbol{x}')$, such as the Matérn family's limiting squared exponential kernel, give rise to a significant portion of GP priors in use today. For centered priors $f \sim \mathcal{GP}(0, k)$, stationarity encodes the belief that the relationship between process values $f(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_j)$ is solely determined by the

difference $\boldsymbol{x}_i - \boldsymbol{x}_j$ between locations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Simple but expressive, stationarity is the go-to modeling assumption in many applied settings.

These kernels exhibit a variety of special properties that greatly facilitate the construction of efficient, approximate priors. Here, we restrict attention to kernels admitting a spectral density ρ, and focus on the class of estimators formed by discretizing the spectral representation of k

$$k(\boldsymbol{x} - \boldsymbol{x}') = \int_{\mathbb{R}^d} e^{2\pi i \boldsymbol{\omega}^\top (\boldsymbol{x} - \boldsymbol{x}')} \rho(\boldsymbol{\omega})\, d\boldsymbol{\omega} \qquad \rho(\boldsymbol{\omega}) = \int_{\mathbb{R}^d} e^{-2\pi i \boldsymbol{\omega}^\top \boldsymbol{x}} k(\boldsymbol{x})\, d\boldsymbol{x}\,. \tag{3.21}$$

By the *kernel trick* (Schölkopf and Smola, 2001), a kernel k can be written as the inner product in a corresponding reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ equipped with a feature map $\varphi : \mathcal{X} \to \mathcal{H}_k$. In many cases, this inner product can be approximated by

$$k(\boldsymbol{x}, \boldsymbol{x}') = \langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{x}') \rangle_{\mathcal{H}_k} \approx \boldsymbol{\phi}(\boldsymbol{x})^\top \, \overline{\boldsymbol{\phi}(\boldsymbol{x}')}, \tag{3.22}$$

where $\boldsymbol{\phi} : \mathcal{X} \to \mathbb{C}^\ell$ is some finite-dimensional feature map and $\overline{\boldsymbol{\phi}(\boldsymbol{x}')}$ denotes the complex conjugate. Based on this idea, the method of *random Fourier features* (Rahimi and Recht, 2008) constructs a Monte Carlo estimate to a stationary kernel by representing the right-hand side of (3.22) with ℓ complex exponential basis functions $\phi_j(\boldsymbol{x}) = \ell^{-1/2} \exp(2\pi i \boldsymbol{\omega}_j^\top \boldsymbol{x})$, whose parameters $\boldsymbol{\omega}_j$ are sampled proportional to the corresponding spectral density $\rho(\boldsymbol{\omega}_j)$.[2]

Given an ℓ-dimensional basis $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_\ell)$, we may now proceed to approximate the true prior according to the Bayesian linear model

$$\tilde{f}(\ \cdot\ ) = \sum_{i=1}^{\ell} w_i \phi_i(\ \cdot\ ) \qquad\qquad w_i \sim \mathcal{N}(0, 1). \tag{3.23}$$

Under this approximation, $\tilde{f}$ is a random function satisfying $\tilde{\boldsymbol{f}}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^\top)$, where $\boldsymbol{\Phi}_n = \boldsymbol{\phi}(\mathbf{X}_n)$ is an $n \times \ell$ matrix of features. Per the beginning of this section, then, $\tilde{f}$ is a Gaussian process whose covariance approximates that of f.

Sutherland and Schneider (2015) showed that the worst-case kernel approximation error

$$\max_{\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}} \left| k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}') \right| \tag{3.24}$$

introduced by approximating a stationary kernel by an ℓ-dimensional random Fourier basis $\boldsymbol{\phi}$ decays at a dimension-free rate $\ell^{-1/2}$. In Section 3.5, we will demonstrate that the same is true of worst-case covariance errors between true posteriors and posteriors formed by exactly updating an RFF approximation to the prior in the canonical basis. This property is useful when quantities of interest depend solely on pointwise evaluations $f(\boldsymbol{x})$ since it often simplifies the process of choosing ℓ.

In many practical settings, however, the primary allure of pathwise approaches is their ability jointly evaluate draws of f in a straightforward and scalable way. Much

[2] Using elementary trigonometric identities, we may also derive a related family of basis functions $\phi : \mathcal{X} \to \mathbb{R}^\ell$ with $\phi_j(\boldsymbol{x}) = \sqrt{2/\ell} \cos(2\pi \boldsymbol{\omega}_j^\top \boldsymbol{x} + \tau_j)$, where $\tau_j \sim \mathcal{U}(0, 2\pi)$.

of the analysis provided in Section 3.5 therefore centers on Wasserstein distances between processes, which provide insight for errors defined with respect to functionals acting on f and $\tilde{f}$.

3.3.3 Karhunen–Loève expansions

While exploitation of stationarity is arguably the most common route when constructing approximate priors, it is neither unique nor optimal. A powerful alternative is to utilize the *Karhunen–Loève expansion* of a Gaussian process prior (Castro et al., 1986; Fukunaga, 2013).

We begin by considering the family of ℓ-dimensional Bayesian linear models $\tilde{f}(\,\cdot\,) = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w}$ consisting of orthonormal basis functions $\phi_i : \mathcal{X} \to \mathbb{R}$ on a compact space $\mathcal{X}$. Following standard theory (Fukunaga, 2013), the *optimal* $\tilde{f}$ for approximating a Gaussian process f (in the sense of minimizing mean square error) is found by truncating its Karhunen–Loève expansion

$$f(\,\cdot\,) = \sum_{i=1}^{\infty} w_i \phi_i(\,\cdot\,) \qquad\qquad w_i \sim \mathcal{N}(0, \lambda_i), \tag{3.25}$$

where ϕ_i and λ_i are, respectively, the i-th eigenfunction and eigenvalue of the covariance operator $\psi \mapsto \int_{\mathcal{X}} \psi(\boldsymbol{x}) k(\boldsymbol{x}, \,\cdot\,)\, d\boldsymbol{x}$, written in decreasing order of λ_i.[3] Truncated versions of these expansions are used as both bases for constructing optimal approximate GPs (Zhu et al., 1997; Solin and Särkkä, 2020) and modeling tools in their own right (Krainski et al., 2018). Depending on the case, eigenfunctions ϕ_i are either derived from first principles (Krainski et al., 2018) or obtained by numerical methods (Lindgren et al., 2011; Lord et al., 2014; Solin and Kok, 2019).

In addition to being optimal, Karhunen–Loève expansions are exceedingly general. Even when a covariance function k is non-stationary or the domain $\mathcal{X}$ is non-Euclidean—such as when Gaussian processes are used to represent functions on manifolds (Borovitskiy et al., 2020) and graphs (Borovitskiy et al., 2021)—the Karhunen–Loève expansion often exists.

Widespread use of truncated eigensystems is largely impeded by their frequent lack of convenient, analytic forms. This issue is compounded by the fact that efficient, numerical methods for obtaining (3.25) typically require us to manipulate bespoke mathematical properties of specific kernels. These properties are often closely related to the differential-equation-based perspectives of Gaussian processes introduced in the following section.

3.3.4 Stochastic partial differential equations

Many Gaussian process priors, such as the Matérn family, can be expressed as solutions of *stochastic partial differential equations* (SPDEs). SPDEs are common in

[3] These eigenvalues are well-ordered and countable as a consequence of the compactness of $\mathcal{X}$.

fields such as physics, where they describe natural phenomena (such as diffusion and heat transfer); many of which share a deep connection with the squared exponential kernel (Grigoryan, 2009). Additionally, SPDEs are often the starting point when designing non-stationary GP priors (Krainski et al., 2018). Below, we detail how the *Galerkin finite element method* (Evans, 2010; Lindgren et al., 2011; Lord et al., 2014) can be used to construct Bayesian linear models that approximate GP priors capable of being represented as SPDEs.

Suppose a Gaussian process $f \sim \mathcal{GP}(0, k)$ satisfies $\mathcal{L}f = \mathcal{W}$, where $\mathcal{L}$ is a linear differential operator and $\mathcal{W}$ is a Gaussian white noise process (Lifshits, 2012). Here, we demonstrate how to derive a Gaussian process $\tilde{f}$ that approximately satisfies this SPDE. To begin, we express $\mathcal{L}f = \mathcal{W}$ in its weak form[4]

$$\int_{\mathcal{X}} (\mathcal{L}f)(\boldsymbol{x}) g(\boldsymbol{x})\, d\boldsymbol{x} = \int_{\mathcal{X}} g(\boldsymbol{x})\, d\mathcal{W}(\boldsymbol{x}), \tag{3.26}$$

where g is an arbitrary element of an appropriate class of test functions. Next, we proceed by approximating both the desired solution f and the test function g with respect to a finite-dimensional basis as $\tilde{f}(\,\cdot\,) = \sum_{i=1}^{\ell} w_i \phi_i(\,\cdot\,)$ and $\tilde{g}(\,\cdot\,) = \sum_{j=1}^{\ell} v_j \phi_j(\,\cdot\,)$. Substituting these terms into (3.26) and differentiating both sides with respect to the coefficients of $\tilde{g}$, we obtain the following expression for each $j = 1, \ldots, \ell$:

$$\sum_{i=1}^{\ell} w_i \underbrace{\int_{\mathcal{X}} (\mathcal{L}\phi_i)(\boldsymbol{x}) \phi_j(\boldsymbol{x})\, d\boldsymbol{x}}_{A_{ij}} = \underbrace{\int_{\mathcal{X}} \phi_j(\boldsymbol{x})\, d\mathcal{W}(\boldsymbol{x})}_{b_j}. \tag{3.27}$$

Defining $\mathbf{M} = \mathrm{Cov}(\boldsymbol{b})$, where $\mathrm{Cov}(b_i, b_j) = \langle \phi_i, \phi_j \rangle$ coincides with the finite-element mass matrix, allows us to rearrange this system of random linear equations in matrix-vector form by writing $\mathbf{A}\boldsymbol{w} = \boldsymbol{b}$. The basis coefficients of the random function $\tilde{f}$ are, therefore, distributed as $\boldsymbol{w} \sim \mathcal{N}\big(\mathbf{0}, \mathbf{A}^{-1}\mathbf{M}\mathbf{A}^{-\top}\big)$. As in the previous sections, $\tilde{f}$ can be seen as the weight-space view of a corresponding Gaussian process.

A popular choice is to employ compactly supported basis functions ϕ_i (Lindgren et al., 2011). The matrices $\mathbf{A}$ and $\mathbf{M}$ are then sparse, and the resulting linear systems can be solved efficiently. For example, the family of piecewise linear basis functions is a simple but effective choice for second order differential operators $\mathcal{L}$ (Evans, 2010; Lord et al., 2014).[5]

3.3.5 Discussion

This section has focused on identifying finite-dimensional bases with which to construct Bayesian linear models $\tilde{f}(\,\cdot\,) = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w}$. These models can be seen as *weight-space* interpretations (Rasmussen and Williams, 2006) of corresponding Gaussian process priors $\tilde{f} \sim \mathcal{GP}(0, \tilde{k})$ with covariance functions $\tilde{k}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\Sigma}_{\boldsymbol{w}} \boldsymbol{\phi}(\boldsymbol{x}')$.

[4] One typically integrates $(\mathcal{L}f)(\boldsymbol{x})g(\boldsymbol{x})$ by parts, either by necessity or due to affordances of the basis ϕ_i. We suppress this to ease notation.

[5] A second order differential operator gives rise to a first-order bilinear form when integrated by parts, which matches with piecewise linear basis functions which are once differentiable almost everywhere. For higher-order operators, a piecewise polynomial basis may be used instead.

Since $\boldsymbol{w}$ and $\tilde{\boldsymbol{f}}_n = \boldsymbol{\Phi}_n \boldsymbol{w}$ are jointly normal, Theorem 3.14 implies that we may enforce the condition $\tilde{\boldsymbol{f}}_n = \boldsymbol{y}$ by writing[6]

$$\boldsymbol{\phi}(\,\cdot\,)^\top (\boldsymbol{w} \mid \boldsymbol{y}) \stackrel{\mathrm{d}}{=} \boldsymbol{\phi}(\,\cdot\,)^\top \Big(\boldsymbol{w} + \boldsymbol{\Phi}_n^\top (\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^\top)^{-1} (\boldsymbol{y} - \boldsymbol{\Phi}_n \boldsymbol{w}) \Big). \tag{3.28}$$

This result encourages us to approximate posteriors in much the same way as we have priors. After all, if we have chosen a basis $\boldsymbol{\phi}$ that encodes our prior knowledge for f (such as how smooth we believe this function to be), then it is reasonable to think that $\boldsymbol{\phi}$ will further enable us to efficiently approximate $f \mid \boldsymbol{y}$. To the extent that this approach may seem like the natural evolution of ideas discussed in this section, we argue for the benefits of *decoupling* the representation of the prior from that of the data.

The trouble with using a finite set of homogeneous basis functions $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_\ell)$ to represent both the prior and the data is that these two tasks focus on different things. To accurately approximate a prior is to faithfully describe a random function f on a domain $\mathcal{X}$. Consequently, parsimonious approximations $\tilde{f}$ employ global basis functions that vary non-trivially everywhere on $\mathcal{X}$. This is largely why, e.g., Fourier features are an attractive choice for approximating stationary priors. But what of the data?

Conditioning on observations $\boldsymbol{y}$ requires us to convey how our understanding of f has changed. In most cases, we choose priors (and likelihoods) that reflect the belief that an observation y_i only informs us about the process f in the immediate vicinity of a point $\boldsymbol{x}_i$. Updating f to account for $\boldsymbol{y}$, therefore, typically focuses on process values corresponding to specific regions of $\mathcal{X}$. Rather than global basis functions, the data is best characterized by local ones that have near-zero values outside of the aforementioned regions. Not coincidentally, the canonical basis functions $k(\,\cdot\,, \boldsymbol{x})$ fit this description perfectly when the chosen prior implies that y_i is only locally informative.

A key property of pathwise conditioning is that it not only provides us with a natural decomposition of GP posteriors—as sums of prior random variables and data-driven updates—but enables us to represent these terms in separate bases. Similar ideas can be found in recent works that explore alternative decompositions of Gaussian processes, such as separation of mean and covariance functions (Cheng and Boots, 2017; Salimbeni et al., 2018) or decoupling of RKHS subspaces and their orthogonal complements (Shi et al., 2020). Unlike these works, however, we stress decoupling in the sense of using different classes of basis functions to represent different aspects of GP posteriors. While this type of decoupling is not unique to pathwise approaches (Lázaro-Gredilla and Figueiras-Vidal, 2009; Hensman et al., 2017), they drastically simplify the process by eliminating the need to analytically solve for sufficient statistics.

This line of reasoning also helps to explain why finite-dimensional GPs constructed from homogeneous basis functions often produce poorly-calibrated posteriors. For

[6]Practical variants of (3.28) avoid inverting $\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^\top$ by employing, e.g., Gaussian likelihoods (Section 3.4.1).

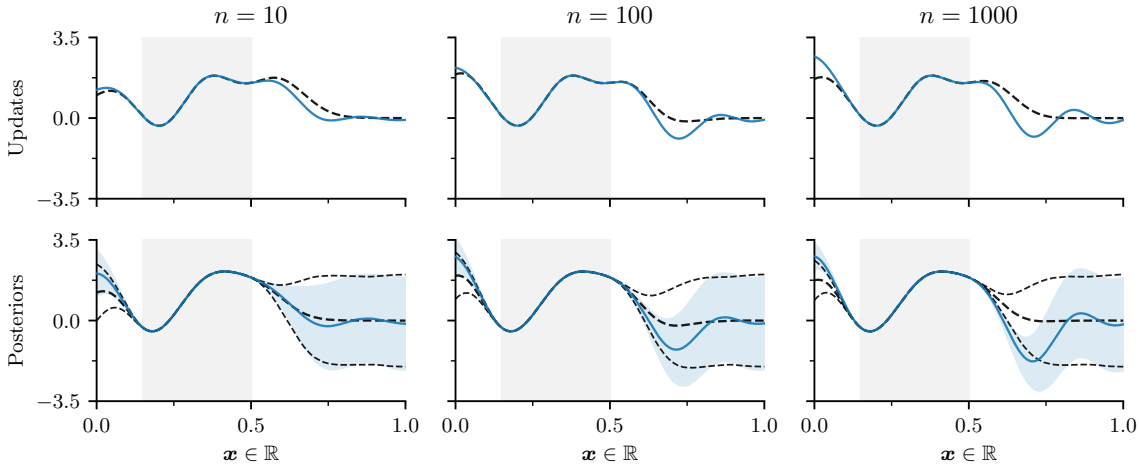

Figure 3.3: Overview of variance starvation when conditioning on $n \in \{10, 100, 1000\}$ observations of the form $y_i \sim \mathcal{N}(f_i, 10^{-5})$ located within the gray shaded region. *Top:* Comparison of pathwise updates to a single draw from an approximate prior $\tilde{f}(\,\cdot\,) = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w}$, constructed using $\ell = 1000$ Fourier features ϕ. Updates defined using the same Fourier basis $\boldsymbol{\phi}(\,\cdot\,)$ and the canonical basis functions $k(\,\cdot\,, \mathbf{X})$ are shown in blue and dashed-black, respectively. *Bottom:* Mean and two standard deviations of the empirical posteriors formed by applying the aforementioned updates to 10^5 draws from the approximate prior.

now, we restrict our attention to the issue of *variance starvation* (Wang et al., 2018; Mutny and Krause, 2018; Calandriello et al., 2019) and return this topic in Section 3.4.5. Figure 3.3 demonstrates what happens as the number of observations $n = |\boldsymbol{y}|$ approaches the number of random Fourier features $\ell = 1000$ used to approximate a squared exponential kernel. In general, the approximate posteriors produce extrapolations which become increasingly erratic. Note that the rate at which these defects materialize depends upon the choice of kernel and likelihood. In the figure, posteriors yielded by pathwise updates in canonical and Fourier bases (all other things being held equal) diverge as the number of observations n approaches the number of random Fourier features ℓ. This pattern emerges because the Fourier basis is better at describing stationary priors than non-stationary posteriors. Fourier features excel at capturing the global properties of the prior, but struggle to portray the localized effects of the data.

Of course, different types of data impose different kinds of conditions on the process f. We now examine various pathwise updates that enforce prominent types of conditions.

3.4 Conditioning via pathwise updates

Building off of the foundation prepared in Section 3.2, we now adapt Corollary 3.17 to accommodate different types of conditions and computational budgets. Throughout this section, we use $\boldsymbol{\gamma}$ to denote the random variable realized by observations $\boldsymbol{y}$ under the chosen likelihood.

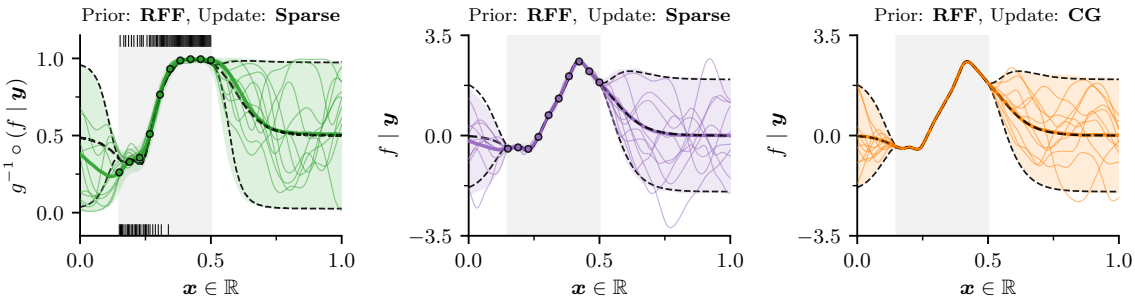

Figure 3.4: Visual comparison of different pathwise updates. *Left and middle:* Variational inference is used to learn sparse updates at $m = 10$ inducing locations $\mathbf{Z}$ (circles). *Right:* preconditioned conjugate gradients is used to iteratively solve for Gaussian updates. In all cases, 1000 observations $\boldsymbol{y}$ are evenly spaced in the shaded region. Dashed lines denote mean and two standard deviations of ground truth posteriors, colored regions and thicker lines denote those of empirical ones. Middle and right plots illustrate regression with a Gaussian likelihood $\mathcal{N}(y_i \mid f_i, 10^{-3})$. The left plot shows binary classification with a Bernoulli likelihood and probit link function g; every tenth label is shown as a small, vertical bar.

3.4.1 Gaussian updates

Corollary 3.17 treats observations $\boldsymbol{y}$ as a realization of process values $\boldsymbol{f}_n = f(\mathbf{X}_n)$. Hence, the conditions it imposes manifest as the equality constraint $\boldsymbol{f}_n = \boldsymbol{y}$. In the real world, however, we seldom observe $\boldsymbol{f}_n$ directly. To account for this nuance, an observation y is modeled by a *likelihood* $p(y \mid f(\boldsymbol{x}))$. Viewed from this perspective, the equality constraint $\boldsymbol{f}_n = \boldsymbol{y}$ correspond to the limit where p contracts to a point mass. Seeing as y usually fails to fully disambiguate the true value of $f(\boldsymbol{x})$, we typically employ likelihoods that induce weaker conditions than strict equalities.

For regression problems, the most common choice is to employ a Gaussian likelihood $p(y \mid f(\boldsymbol{x})) = \mathcal{N}(y \mid f(\boldsymbol{x}), \sigma^2)$, the log of which penalizes the squared Mahalanobis distance of $f(\boldsymbol{x})$ from y. Under the corresponding observation model $\boldsymbol{\gamma} = \boldsymbol{f}_n + \boldsymbol{\varepsilon}$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, f and $\boldsymbol{y}$ are jointly Gaussian. By Corollary 3.17 then, we may condition f on $\boldsymbol{\gamma} = \boldsymbol{y}$ by writing

$$(f \mid \boldsymbol{\gamma} = \boldsymbol{y})(\,\cdot\,) \overset{\mathrm{d}}{=} f(\,\cdot\,) + k(\,\cdot\,, \mathbf{X})(\mathbf{K}_{n,n} + \sigma^2\mathbf{I})^{-1}(\boldsymbol{y} - \boldsymbol{f}_n - \boldsymbol{\varepsilon}). \tag{3.29}$$

Rather than exactly passing through observations $\boldsymbol{y}$, the conditioned path $f \mid \boldsymbol{y}$ now smoothly interpolates between them. In cases where $\boldsymbol{\gamma}$ is not a Gaussian random variable, additional tools are needed.

3.4.2 Non-Gaussian updates

In the general setting, where the random variable $\boldsymbol{\gamma}$ is arbitrarily distributed under the chosen likelihood, $\boldsymbol{\gamma}$ relates to process values $\boldsymbol{f}$ by way of the non-conjugate prior

$$p(\boldsymbol{\gamma}, \boldsymbol{f}) = p\left(\boldsymbol{\gamma} \mid g^{-1}(\boldsymbol{f})\right) \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{\mu}, \mathbf{K}), \tag{3.30}$$

where the *link function* $g : \mathcal{Y} \to \mathbb{R}$ maps from the space of predictions $\mathcal{Y} \subset \mathbb{R}$ to the range of f. For binary classification problems, popular choices for $g : [0, 1] \to \mathbb{R}$ include logit and probit functions (Rasmussen and Williams, 2006). The left column of Figure 3.4 illustrates this scenario using methods described below.

Even under a non-conjugate prior (3.30), the conditional expectation $\mathbb{E}(f \mid \boldsymbol{\gamma})$ and the residual $f - \mathbb{E}(f \mid \boldsymbol{\gamma})$ it induces are uncorrelated (see Section 3.1.3). Since $p(\boldsymbol{f}, \boldsymbol{\gamma})$ may not be Gaussian, however, it no longer follows that this lack of correlation implies independence—hence, the pathwise update (3.16) may not hold.

Exact Bayesian inference and prediction are typically intractable when dealing with non-conjugate priors. Strategies for circumventing this issue generally approximate the true posterior by introducing an auxiliary random variable $\boldsymbol{u} \sim q(\boldsymbol{u})$ such that $f \mid \boldsymbol{u}$ resembles $f \mid \boldsymbol{y}$ according to a chosen measure of similarity (Nickisch and Rasmussen, 2008; Hensman et al., 2015). For practical reasons, $\boldsymbol{u}$ is typically assumed to be jointly Gaussian with $\boldsymbol{f}$.[7] Consequently, non-conjugate priors $p(\boldsymbol{f}, \boldsymbol{\gamma})$ are replaced by conjugate ones $p(\boldsymbol{f}, \boldsymbol{u})$ to aid in the construction of approximate posteriors, whereupon Matheron's update rule holds once more. The following section explores these *sparse* approximations in greater detail.

3.4.3 Sparse updates

Approximations to GP posteriors frequently revolve around conditioning a process f on a random variable $\boldsymbol{u} = (u_1, \ldots, u_m) \in \mathbb{R}^m$. Per the previous section, this may be because the outcome variable $\boldsymbol{\gamma}$ is non-Gaussian (Nickisch and Rasmussen, 2008; Titsias and Lawrence, 2010; Hensman et al., 2015). Alternatively, the $\mathcal{O}(n^3)$ cost for directly conditioning on all $n = |\boldsymbol{y}|$ observations may be prohibitive (Titsias, 2009a; Hensman et al., 2013). In these cases and more, we would like to infer a distribution $q(\boldsymbol{u})$ such that $f \mid \boldsymbol{u}$ explains the data. Defining (approximate) posteriors in this way not only avoids potential issues arising from the non-Gaussianity of $\boldsymbol{\gamma}$, but associates the computational cost of conditioning with $\boldsymbol{u}$. As discussed below, this leads to pathwise updates that run in $\mathcal{O}(m^3)$ time.

Comprehensive treatment of different approaches to learning *inducing distributions* $q(\boldsymbol{u})$ is beyond the scope of this work. In general, however, these procedures operate by finding an approximate posterior $q(\boldsymbol{f}, \boldsymbol{u})$ within a tractable family of approximating distributions $\mathcal{Q}$. For reasons that will soon become clear, this family of distributions typically includes an additional set of parameters $\mathbf{Z}$, which help to define the joint distribution $p(\boldsymbol{f}, \boldsymbol{u})$. To help streamline our presentation, we focus on the simplest and most widely used abstraction for inducing variables $\boldsymbol{u}$: namely, *pseudo-data*.

The noise-free pseudo-data framework (Snelson and Ghahramani, 2006; Quiñonero-Candela et al., 2007; Titsias, 2009a) treats each draw of a random vector $\boldsymbol{u} \sim q(\boldsymbol{u})$ as a realization of process values $\boldsymbol{f}_m = f(\mathbf{Z})$ at a corresponding set of tunable locations $\mathbf{Z} \in \mathcal{X}^m$. This paradigm gets its name from the intuition that the (random) collection

[7] In the special case where $p(\boldsymbol{f}, \boldsymbol{\gamma})$ is Gaussian, the optimal q is also Gaussian (Titsias, 2009b).

of pseudo-data $(\boldsymbol{z}_j, u_j)_{j=1}^m$ mimics the effect of a noise-free data set $(\boldsymbol{x}_i, f_i)_{i=1}^n$ on f. By construction, $\boldsymbol{u}$ is jointly Gaussian with f.[8] Appealing to Corollary 3.17, we define the *sparse pathwise update* as

$$(f \mid \boldsymbol{u})(\,\cdot\,) \overset{\mathrm{d}}{=} f(\,\cdot\,) + \underbrace{\sum_{i=1}^m v_i k(\,\cdot\,, \boldsymbol{z}_i)}_{m\text{-dimensional basis}}, \tag{3.31}$$

where $\boldsymbol{v} = \mathbf{K}_{m,m}^{-1}(\boldsymbol{u} - \boldsymbol{f}_m)$. This formula is identical to the one given by Corollary 3.17, save for the fact that we now sample $\boldsymbol{u} \sim q(\boldsymbol{u})$ and solve for a linear system involving the $m \times m$ covariance matrix $\mathbf{K}_{m,m} = k(\mathbf{Z}, \mathbf{Z})$ at $\mathcal{O}(m^3)$ cost. The middle column of Figure 3.4 illustrates the sparse update induced by Gaussian $\boldsymbol{u} \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{u}}, \boldsymbol{\Sigma}_{\boldsymbol{u}})$ with learned moments $\boldsymbol{\mu}_{\boldsymbol{u}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{u}}$.

Just as we can imitate process values $\boldsymbol{f}_n$, we can also emulate (Gaussian) observations $\boldsymbol{y}$. This intuition leads to the Gaussian pseudo-data family of inducing distributions, whose moments

$$\boldsymbol{\mu}_u = \mathbf{K}_{m,m}(\mathbf{K}_{m,m} + \boldsymbol{\Lambda})^{-1}\tilde{\boldsymbol{y}} \qquad \boldsymbol{\Sigma}_{\boldsymbol{u}} = (\mathbf{K}_{m,m}^{-1} + \boldsymbol{\Lambda})^{-1} \tag{3.32}$$

are parameterized by *pseudo-observations* $\tilde{\boldsymbol{y}} \in \mathbb{R}^m$ and *pseudo-noise* $\tilde{\boldsymbol{\sigma}} \in \mathbb{R}_+^m$, where $\boldsymbol{\Lambda} = \operatorname{diag}(\tilde{\boldsymbol{\sigma}}^2)$. This choice of parameterization is motivated by the observation that, given $n \leq m$ Gaussian random variables $\boldsymbol{\gamma} \sim \mathcal{N}(\boldsymbol{f}_n, \sigma^2\mathbf{I})$, the family of distributions it generates contains the optimal q despite housing only $\mathcal{O}(m)$ free terms (Seeger, 1999; Opper and Archambeau, 2009).[9] Using the Gaussian pathwise update (3.29), we may express $\boldsymbol{u}$ itself as

$$\boldsymbol{u} \overset{\mathrm{d}}{=} \boldsymbol{f}_m + \mathbf{K}_{m,m}(\mathbf{K}_{m,m} + \boldsymbol{\Lambda})^{-1}(\tilde{\boldsymbol{y}} - \boldsymbol{f}_m - \tilde{\boldsymbol{\varepsilon}}) \qquad \tilde{\boldsymbol{\varepsilon}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}). \tag{3.33}$$

Here, despite the fact that $\boldsymbol{f}_m$ and $\tilde{\boldsymbol{\varepsilon}}$ generate $\boldsymbol{u}$, it remains the case that $\operatorname{Cov}(\boldsymbol{f}_m + \tilde{\boldsymbol{\varepsilon}}, \boldsymbol{u}) = \mathbf{0}$. Substituting this expression into (3.31) and simplifying gives the pathwise update[10]

$$(f \mid \boldsymbol{u})(\,\cdot\,) \overset{\mathrm{d}}{=} f(\,\cdot\,) + k(\,\cdot\,, \mathbf{Z})(\mathbf{K}_{m,m} + \boldsymbol{\Lambda})^{-1}(\tilde{\boldsymbol{y}} - \boldsymbol{f}_m - \tilde{\boldsymbol{\varepsilon}}). \tag{3.34}$$

Hence, while sampling $\boldsymbol{u}$ is more complicated in the Gaussian pseudo-data case, the resulting pathwise update is straightforward. This family of inducing distributions is particularly advantageous in the large m setting, both because it contains only $\mathcal{O}(m)$ free parameters and for reasons discussed in the following section.

In rough analogy to methods discussed in Section 3.3, we may think of the sparse updates introduced here as using an m-dimensional basis $k(\,\cdot\,, \mathbf{Z})$ to approximate functions defined in terms of the n-dimensional basis $k(\,\cdot\,, \mathbf{X}_n)$. In practice, this basis is often efficient because neighboring training locations give rise to similar basis

[8] This holds whenever $\boldsymbol{u}$ and f are linearly related (Lázaro-Gredilla and Figueiras-Vidal, 2009).

[9] We may recover the true posterior by taking $(\tilde{y}_i, \tilde{\sigma}_i) = (y_i, \sigma)$ for all $i \leq n$ and sending $\tilde{\sigma}_i \to \infty$ otherwise.

[10] This same line of reasoning leads to a *rank-1 pathwise update* for cases where conditions arrive online.

functions. Kernel basis functions at appropriately chosen sets of $m \ll n$ locations $\mathbf{Z}$ exploit this redundancy to produce a sparser, more cost-efficient representation. Burt et al. (2020) study this problem in detail and derive bounds on the quality of variational approximations to GP posteriors as $m \to n$.

3.4.4 Iterative solvers

Throughout this section, we have focused on the high-level properties of pathwise updates in relation to various problem settings. We have said little, however, regarding the explicit means of executing such an update. In all cases discussed here, pathwise updates have amounted to solutions to systems of linear equations. For example, the update originally featured in Corollary 3.17 solves the system $\mathbf{K}_{n,n}\boldsymbol{v} = \boldsymbol{y} - \boldsymbol{f}_n$ for a vector of coefficients $\boldsymbol{v}$, which defines how the same realization of f changes when subjected to the condition $\boldsymbol{f}_n = \boldsymbol{y}$. Given a reasonable number of conditions n (up to several thousand), we may obtain $\boldsymbol{v}$ by first computing the Cholesky factor $\mathbf{L}_{n,n} = \mathbf{K}_{n,n}^{1/2}$ and then solving for a pair of triangular systems $\mathbf{L}_{n,n}\bar{\boldsymbol{v}} = \boldsymbol{u} - \boldsymbol{f}_n$ and $\mathbf{L}_{n,n}^{\top}\boldsymbol{v} = \bar{\boldsymbol{v}}$. For large n, however, the $\mathcal{O}(n^3)$ time complexity for carrying out this recipe is typically prohibitive.

Rather than solving for coefficients $\boldsymbol{v}$ directly, we may instead employ an *iterative solver* that constructs a sequence of estimates $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \ldots$ to $\boldsymbol{v}$, such that $\boldsymbol{v}^{(j)}$ converges to the true $\boldsymbol{v}$ as j increases. Depending on the numerical properties of the linear system in question, it is possible (or even likely) that a high-quality estimate $\boldsymbol{v}^{(j)}$ will be obtained after only $j \ll n$ iterations. This line of reasoning features prominently in a number of recent works, where iterative solvers have been shown to be highly competitive for purposes of approximating GP posteriors (Pleiss et al., 2018; Gardner et al., 2018; Wang et al., 2019). The right column of Figure 3.4 visualizes an iterative solution to the Gaussian pathwise update (3.29) obtained using preconditioned conjugate gradients (Gardner et al., 2018).

In these cases, posterior sampling via pathwise conditioning enjoys an important advantage over distributional approaches: it allows us to solve for linear systems of the form $\mathbf{K}_{n,n}^{-1}\boldsymbol{v}$ rather than working with $\mathbf{K}_{*,*|n}^{1/2}\boldsymbol{\zeta}$. Whereas the former amounts to a standard solve, the latter often requires special considerations (Pleiss et al., 2020) and can be difficult to work with when typical square root decompositions prove impractical (Parker and Fox, 2012).

Lastly, we note that these techniques can be combined with sparse approximations for improved scaling in m and faster convergence of iterative solves. As a concrete example, we return to the Gaussian pseudo-data variational family (3.32). By construction, the corresponding pathwise update (3.34) closely resembles the original Gaussian update (3.29). In general, however, pseudo-noise variances $\tilde{\sigma}_i^2$ are often significantly larger than the true noise variance σ^2. The resulting linear system $(\mathbf{K}_{m,m} + \boldsymbol{\Lambda})^{-1}\boldsymbol{v}$ is, therefore, substantially better-conditioned than that of the exact alternative—implying that it can be solved in far fewer iterations.

3.4.5 Discussion

In Section 3.3.5, we discussed finite-dimensional approximations of Gaussian process posteriors. There, we explored how the globality of the prior reinforces the use of basis functions $\phi_i : \mathcal{X} \to \mathbb{R}$ that inform us about f over the entire domain $\mathcal{X}$, while the localized effects of the data encourage the use of ϕ_i that only tell us about f on subsets of $\mathcal{X}$. This conflict hinders our ability to efficiently represent both the prior and the data (i.e., the posterior) using a single class of basis functions. That discussion ended with a demonstration of what happens when $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_\ell)$ solely consists of global basis functions, specifically random Fourier features. Most works, however, have focused on the use of canonical basis functions $k(\,\cdot\,, \boldsymbol{x})$, which are typically local. This section, therefore, aims to fill in the gaps.

At the end of Section 3.3.5, we saw how trouble conveying the data in global bases led to approximate posteriors that were starved for variance (Figure 3.3). Writing the update rules—for a draw from an approximate prior $\tilde{f}(\,\cdot\,) = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w}$ subject to the condition $\tilde{\boldsymbol{f}}_n = \boldsymbol{y}$—in both unified and decoupled bases side-by-side helps to highlight their key differences

$$\underbrace{\tilde{f}(\,\cdot\,) + \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{\Phi}_n^\top \Big(\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^\top\Big)^{-1} \Big(\boldsymbol{y} - \tilde{\boldsymbol{f}}_n\Big)}_{\text{unified approximate posterior}} \quad \underbrace{\tilde{f}(\,\cdot\,) + k(\,\cdot\,, \mathbf{X}_n) \mathbf{K}_{n,n}^{-1} \Big(\boldsymbol{y} - \tilde{\boldsymbol{f}}_n\Big)}_{\text{decoupled approximate posterior}}. \tag{3.35}$$

On the right, the cross-covariance term $\boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{\Phi}_n^\top = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{\phi}(\mathbf{X}_n)$ is replaced by $k(\,\cdot\,, \mathbf{X}_n)$. Seeing as the former is often chosen to approximate the latter in a way that converges when an appropriate limit is taken, for instance in (3.22), it comes as no surprise that $k(\,\cdot\,, \mathbf{X}_n)$ more accurately represents data. Moreover, the matrix inverse $\Big(\boldsymbol{\Phi}_n \boldsymbol{\Phi}_n^\top\Big)^{-1}$ appearing on the left is often ill-conditioned and, therefore, amplifies numerical errors. Finite-dimensional GPs constructed from local basis functions exhibit similar issues, albeit for essentially the opposite reason. Rather than failing to adequately represent the data, local basis functions struggle to reproduce the prior.

Many approaches to approximating Gaussian processes $f \sim \mathcal{GP}(0, k)$ revolve around representing the data in terms of m-dimensional canonical bases $k(\,\cdot\,, \mathbf{Z})$; for a review, see Quiñonero-Candela et al. (2007). Early iterations of this strategy (Silverman, 1985; Wahba, 1990; Tipping, 2000), typically used $k(\,\cdot\,, \mathbf{Z})$ to define degenerate Gaussian processes (Rasmussen and Williams, 2006). Here, the term *degenerate* emphasizes the fact that the covariance function

$$\tilde{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = k(\boldsymbol{x}_i, \mathbf{Z}) k(\mathbf{Z}, \mathbf{Z})^{-1} k(\mathbf{Z}, \boldsymbol{x}_j) \tag{3.36}$$

of such a process has a finite number of non-zero eigenvalues. From the weight-space perspective, degenerate GPs are Bayesian linear models $\tilde{f}(\,\cdot\,) = k(\,\cdot\,, \mathbf{Z}) \boldsymbol{w}$, which makes it clear that $\tilde{f}(\,\cdot\,)$ goes to zero as $k(\,\cdot\,, \mathbf{Z}) \to \mathbf{0}$. This behavior is particularly troublesome if all $\boldsymbol{z} \in \mathbf{Z}$ are positioned near training locations $\mathbf{X}_n$: since $k(\boldsymbol{x}_*, \mathbf{Z})$ typically vanishes as $\boldsymbol{x}_*$ retreats from $\mathbf{Z}$, both the prior and the posterior collapse to

point masses away from the data.

Instead of focusing on the data, one idea is to start by finding a basis $k(\,\cdot\,, \mathbf{Z})$ capable of accurately reproducing the prior. Accomplishing this feat will require us to use a relatively large number of basis functions, since $\mathbf{Z}$ will need to effectively cover the (compact) domain $\mathcal{X}$. As mentioned in Section 3.3.1, certain kernels produce special kinds of matrices when evaluated on particular sets. Exploiting these special properties—e.g., by taking the Toeplitz matrices formed when evaluating a stationary product kernel k on a regularly spaced grid $\mathbf{Z}$ and embedding them inside of circulant ones (Wood and Chan, 1994; Dietrich and Newsam, 1997)—enables us to drastically reduce the cost of expensive matrix operations, such as multiplies, decompositions, and inverses. Especially when $\mathcal{X}$ is low dimensional, then, we can use the canonical basis to efficiently approximate the prior.

Kernel interpolation methods (Wilson and Nickisch, 2015; Pleiss et al., 2018) take this idea a step further. Given a set of m inducing locations $\mathbf{Z}$, let $\boldsymbol{\xi} : \mathcal{X} \to \mathbb{R}^m$ be a *weight function* (Silverman, 1984) mapping locations $\boldsymbol{x}_i$ onto (sparse) weight vectors $\boldsymbol{\xi}_i$ such that $k(\boldsymbol{x}_i, \mathbf{Z}) \approx \boldsymbol{\xi}_i^\top k(\mathbf{Z}, \mathbf{Z})$. By applying this technique to (3.36), we can define another Gaussian process $g \sim \mathcal{GP}(0, c)$ with degenerate covariance $c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\xi}_i^\top k(\mathbf{Z}, \mathbf{Z})\boldsymbol{\xi}_j$. As a Bayesian linear model, we have $g(\,\cdot\,) = \boldsymbol{\xi}(\,\cdot\,)^\top \boldsymbol{g}_m$. Notice that process values $\boldsymbol{g}_m = g(\mathbf{Z})$ now play the role of random weights $\boldsymbol{w}$ and fully determine the behavior of the random function g. Assuming $\mathbf{Z}$ was chosen so that $k(\mathbf{Z}, \mathbf{Z})$ admits convenient structure, random vectors $\boldsymbol{g}_m \mid \boldsymbol{y}$ and, hence, random functions $(g \mid \boldsymbol{y})(\,\cdot\,)$ can be obtained cheaply (Pleiss et al., 2018). When $\mathbf{Z}$ is sufficiently dense in $\mathcal{X}$ (so as to be reasonably close to $\boldsymbol{x}_*$), this strategy provides an alternative means of efficiently sampling from GP posteriors.

3.4.6 An empirical study

By now, we have explored a variety of techniques for sampling from GP posteriors. Each of these methods is well suited for a particular type of problem. To help shed light on their respective niches, we conducted a simple controlled experiment.

Here, our goal is to better understand how different methods balance the tradeoff of cost and accuracy. We measured cost in terms of runtimes and accuracy in terms of 2-Wasserstein distances between empirical distributions and true posterior (see Section 3.5). To eliminate confounding variables, we assumed a known Matérn-$5/2$ prior on random functions $f : \mathbb{R}^4 \to \mathbb{R}$. All trials began by sampling this prior at n training locations $\mathbf{X}_n$ and 1024 test locations $\mathbf{X}_*$, using either location-scale transforms or random Fourier features. We then used the various update rules explored in this section to condition on n observations $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{f}_n, 10^{-3}\mathbf{I})$.

Sparse updates were constructed using $m = \frac{n}{4}$ inducing variables $\boldsymbol{u}$, whose distributions $q(\boldsymbol{u})$ and inducing locations $\mathbf{Z}$ were obtained by minimizing Kullback–Leibler divergences. Conjugate-gradient-based updates were carried out by, first, computing partial pivoted Cholesky decompositions in order to precondition linear systems $(\mathbf{K}_{n,n} + \sigma^2\mathbf{I})\boldsymbol{v} = (\boldsymbol{y} - \boldsymbol{f}_n - \boldsymbol{\varepsilon})$. We then iteratively solved for Gaussian pathwise updates using the method of conjugate gradients. Stopping conditions for both the

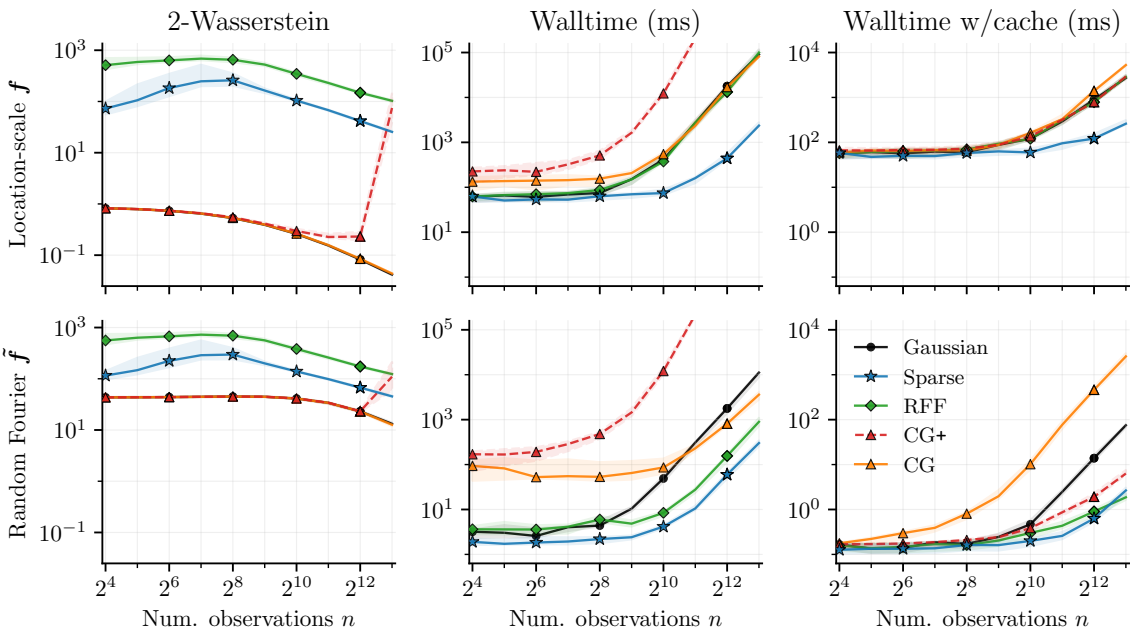

Figure 3.5: Accuracy and cost of different methods for sampling from GP posteriors given n observations. Draws from the prior are generated using either location-scale (*top*) or $\ell = 4096$ random Fourier features (*bottom*). We denote exact Gaussian updates by black dots, sparse updates by blue stars, CG updates by orange and red triangles, and RFF updates by green diamonds. Sparse and RFF updates both utilized $m = \frac{n}{4}$ basis functions. All results are reported as medians and interquartile ranges measured over 32 independent trials. *Left:* 2-Wasserstein distances of empirical distributions of 10^5 samples from the ground truth GP posterior. *Middle and right:* Time taken to generate a draw of $(\boldsymbol{f}_* \mid \cdot\,) \in \mathbb{R}^{1024}$ with and without caching of terms that are independent of $\mathbf{X}_*$.

partial pivoted Cholesky decomposition and conjugate gradient solver were chosen to match those of Gardner et al. (2018). Prior to discussing trends in Figure 3.5, we would like to point out that curves associated with Gaussian updates (black) are heavily obscured: in the left column, by CG-based ones (orange and red) and in top middle and top right plots by RFF-based ones (green).

Comparing the rows of Figure 3.5, we see that random Fourier feature (RFF) approximations to priors introduce modest amounts of error in exchange for large cost reductions. These savings are particularly dramatic in cases where test inputs $\mathbf{X}_*$ significantly outnumber training locations $\mathbf{X}_n$. Echoing discussion in Section 3.3.5, however, m-dimensional random Fourier bases struggle to represent the data. All other things being held equal, sparse updates performed in the canonical basis consistently outperform RFF-based ones. These sparse methods are also considerably faster than competing approaches when $m \ll n$.

Direct comparison of sparse and CG updates is difficult, since both methods are sensitive to various design choices. In our experiments, CG-based updates behaved tantamount to exact ones—with two important caveats. First, CG-based updates were initially slower than exact ones but outpace them as n increased. Second, naïvely computing pathwise updates using CG is highly inefficient when it comes

to caching. When repeatedly conditioning on (potentially different realizations of) $\boldsymbol{\gamma} = \boldsymbol{y}$, one option is to use CG to precompute the matrix inverse $(\mathbf{K}_{n,n} + \sigma^2\mathbf{I})^{-1}$. This CG+ variant is significantly more cache-friendly, but also much more susceptible to round-off error—see dashed red curves in Figure 3.5.

These empirical results help to characterize the behaviors of errors introduced by different approximation schemes, but leave many questions unanswered. In order to fill in some of the remaining gaps, we now analyze various types of approximation error in details.

3.5 Error analysis

Over the course of this section, we will analyze the different types of error introduced by pathwise approximations. Speaking about these errors requires us to agree upon a suitable notion of similarity between Gaussian processes. We investigate 2-Wasserstein distances between true and approximate posteriors, which provide useful information about downstream Monte Carlo errors. These distances measure the similarity of Gaussian processes $\tilde{f}$ and f as the expectation of a metric $d\big(\tilde{f}, f\big)$ under the best possible *coupling* of the two processes. Formally, we have

$$W_{2,d}\big(\tilde{f}, f\big) = \left[\inf_{\pi \in \Pi(\tilde{\mu},\mu)} \mathbb{E}_\pi \, d\big(\tilde{f}, f\big)^2\right]^{1/2}, \tag{3.37}$$

where $\Pi(\tilde{\mu}, \mu)$ denotes the set of valid couplings (Mallasto and Feragen, 2017), i.e. joint measures whose marginals correspond with the Gaussian measures $\tilde{\mu}$ and μ induced by processes $\tilde{f}$ and f, respectively.

Below, we focus on cases where $W_{2,d}$ acts as a proper metric on a space of probability measures such that: if $W_{2,d}(\tilde{f}, f) = 0$, then Monte Carlo estimates based on f and $\tilde{f}$, respectively, will be identically distributed. Since 2-Wasserstein distances majorize 1-Wasserstein distances, this claim may be strengthen in the special case of 1-Lipschitz functionals by appealing to Kantorovich–Rubinstein duality (Villani, 2008)

$$W_{1,d}(\tilde{f}, f) = \left[\inf_{\pi \in \Pi(\tilde{\mu},\mu)} \mathbb{E}_\pi \, d\big(\tilde{f}, f\big)\right] = \sup_{\|T\|_{\mathrm{Lip}\leq 1}} \left\{\int_{\mathcal{F}} T(\tilde{f}) d\tilde{\mu} - \int_{\mathcal{F}} T(f) d\mu\right\}, \tag{3.38}$$

where $\mathcal{F}$ is an appropriately chosen family of functions. In order for this line of reasoning to hold, we assume that the domain $\mathcal{X}$ is a compact subset of some metric measure space $\mathcal{M}$, that $\mathcal{X}$ has finite measure, and that realizations of f almost surely belong to the space of continuous functions equipped with the supremum norm $\mathcal{F} = \mathcal{C}(\mathcal{X})$. Depending on the case, the metric d will either be the L^2 norm or the supremum norm and will be indicated by the appropriate subscript.

Lastly, let us introduce some additional notation to simplify materials presented below. First, we will use $\tilde{f} \mid \boldsymbol{y}$ and $\tilde{f} \mid \boldsymbol{u}$ to denote pathwise conditioning of an approximate prior $\tilde{f}$ via canonical (3.16) and sparse (3.31) update rules, respectively.

These constructions should not be confused with the approximate posteriors discussed in Sections 3.3.5 and 3.4.5. Second, we will superscript covariance functions k to convey their corresponding processes. For example, $k^{(\tilde{f})}$ will denote the kernel of the approximation prior $\tilde{f}$. Third and finally, given a set of n training locations $\mathbf{X}_n \subset \mathcal{X}$, define the *weight function* $\boldsymbol{\xi} : \mathcal{X} \to \mathbb{R}^n$ as

$$\boldsymbol{\xi}(\,\cdot\,) = k(\mathbf{X}_n, \mathbf{X}_n)^{-1} k(\mathbf{X}_n, \,\cdot\,). \tag{3.39}$$

Variants of this function have been extensively studied in the context of regression; see Silverman (1984) and Sollich and Williams (2005) and references contained therein.

3.5.1 Posterior approximation errors

This section adapts the results of Wilson et al. (2020) to study the error in the decoupled approximate posterior

$$(\tilde{f} \mid \boldsymbol{y})(\,\cdot\,) \overset{\mathrm{d}}{=} \tilde{f}(\,\cdot\,) + k(\,\cdot\,, \mathbf{X}_n)\mathbf{K}_{n,n}^{-1}(\boldsymbol{y} - \tilde{\boldsymbol{f}}) = \tilde{f}(\,\cdot\,) + \boldsymbol{\xi}(\,\cdot\,)^\top(\boldsymbol{y} - \tilde{\boldsymbol{f}}) \tag{3.40}$$

formed by updating an ℓ-dimensional approximate priors $\tilde{f}(\,\cdot\,) = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w}$ via an n-dimensional canonical basis $k(\,\cdot\,, \mathbf{X}_n)$ so as to satisfy the condition imposed by n noise-free observations $\boldsymbol{y}$.

Proposition 3.18. *Assume that $\mathcal{X} \subset \mathbb{R}^d$ is compact and that the stationary kernel k is sufficiently regular for $f \sim \mathcal{GP}(\mu, k)$ to be almost surely continuous. Accordingly, if we define $C_1 = \sqrt{2}\,\mathrm{diam}(\mathcal{X})^{d/2}\Big(1 + \|k\|^2_{\mathcal{C}(\mathcal{X}^2)}\|\mathbf{K}_{n,n}^{-1}\|^2_{L(\ell^\infty;\ell^1)}\Big)^{1/2}$, then we have*

$$W_{2,L^2(\mathcal{X})}\Big(\tilde{f} \mid \boldsymbol{y}, f \mid \boldsymbol{y}\Big) = \left(\inf_{\pi \in \Pi(\tilde{\mu},\mu)} \mathbb{E}_\pi \Big\|(\tilde{f} \mid \boldsymbol{y}) - (f \mid \boldsymbol{y})\Big\|^2_{L^2(\mathcal{X})}\right)^{1/2} \leq C_1 W_{2,\mathcal{C}(\mathcal{X})}\Big(\tilde{f}, f\Big), \tag{3.41}$$

where $W_{2,L^2(\mathcal{X})}$ and $W_{2,\mathcal{C}(\mathcal{X})}$ respectively denote 2-Wasserstein distances over the Lebesgue space $L^2(\mathcal{X})$ and the space of continuous functions $\mathcal{C}(\mathcal{X})$ equipped with the supremum norm, $\|\,\cdot\,\|_{\mathcal{C}(\mathcal{X}^2)}$ is the supremum norm over continuous functions, and $\|\,\cdot\,\|_{L(\ell^\infty;\ell^1)}$ is the operator norm between ℓ^∞ and ℓ^1 spaces.

Proof We begin by considering the term inside the expectation in (3.41). Applying Matheron's rule followed by Hölder's inequality ($p = 1$, $q = \infty$), we have

$$\begin{aligned}\Big|(\tilde{f} \mid \boldsymbol{y})(\boldsymbol{x}) - (f \mid \boldsymbol{y})(\boldsymbol{x})\Big|^2 &\leq 2\Big|\tilde{f}(\boldsymbol{x}) - f(\boldsymbol{x})\Big|^2 + 2\Big|\boldsymbol{\xi}(\boldsymbol{x})^\top(\tilde{\boldsymbol{f}}_n - \boldsymbol{f}_n)\Big|^2 \\ &\leq 2\Big\|\tilde{f} - f\Big\|^2_{L^\infty(\mathcal{X})} + 2\|\boldsymbol{\xi}(\boldsymbol{x})\|^2_{\ell^1}\Big\|\tilde{\boldsymbol{f}}_n - \boldsymbol{f}_n\Big\|^2_{\ell^\infty}.\end{aligned} \tag{3.42}$$

Continuing from the second line, the definition of the operator norm implies that

$$\begin{aligned}\Big|(\tilde{f} \mid \boldsymbol{y})(\boldsymbol{x}) - (f \mid \boldsymbol{y})(\boldsymbol{x})\Big|^2 &\leq 2\Big(1 + \|k(\boldsymbol{x}, \mathbf{X}_n)\|_{\ell^\infty}^2 \Big\|\mathbf{K}_{n,n}^{-1}\Big\|_{L(\ell^\infty;\ell^1)}^2\Big)\Big\|\tilde{f} - f\Big\|_{L^\infty(\mathcal{X})}^2 \\ &\leq 2\Big(1 + \Big\|k\Big\|_{\mathcal{C}(\mathcal{X}^2)}^2 \Big\|\mathbf{K}_{n,n}^{-1}\Big\|_{L(\ell^\infty;\ell^1)}^2\Big)\Big\|\tilde{f} - f\Big\|_{L^\infty(\mathcal{X})}^2 \\ &= \underbrace{2\Big(1 + \Big\|k\Big\|_{\mathcal{C}(\mathcal{X}^2)}^2 \Big\|\mathbf{K}_{n,n}^{-1}\Big\|_{L(\ell^\infty;\ell^1)}^2\Big)}_{C_0}\Big\|\tilde{f} - f\Big\|_{\mathcal{C}(\mathcal{X})}^2,\end{aligned} \tag{3.43}$$

where, in the final line, we have used continuity of sample paths to replace $\|\cdot\|_{L^\infty(\mathcal{X})}$ with $\|\cdot\|_{\mathcal{C}(\mathcal{X})}$. We now lift this bound between sample paths to one on 2-Wasserstein distances by integrating both sides with respect to the optimal coupling $\pi \in \Pi(\tilde{\mu}, \mu)$

$$\begin{aligned}W_{2,L^2(\mathcal{X})}\Big(\tilde{f} \mid \boldsymbol{y}, f \mid \boldsymbol{y}\Big) &= \left(\inf_{\pi \in \Pi(\tilde{\mu},\mu)} \mathbb{E}_\pi\Big\|(\tilde{f} \mid \boldsymbol{y}) - (f \mid \boldsymbol{y})\Big\|_{L^2(\mathcal{X})}^2\right)^{1/2} \\ &\leq \left(C_0 \operatorname{vol}(\mathcal{X}) \inf_{\pi \in \Pi(\tilde{\mu},\mu)} \mathbb{E}_\pi\Big\|\tilde{f} - f\Big\|_{\mathcal{C}(\mathcal{X})}^2\right)^{1/2} \\ &\leq C_1 W_{2,\mathcal{C}(\mathcal{X})}\Big(\tilde{f}, f\Big),\end{aligned} \tag{3.44}$$

where $\operatorname{vol}(\mathcal{X})$ denotes the Lebesgue measure of $\mathcal{X}$. Hence, the claim follows. ■

Proposition 3.19. *With the same assumptions, let* $C_2 = n\Big(1 + \|\mathbf{K}_{n,n}^{-1}\|_{\mathcal{C}(\mathcal{X}^2)}\|k\|_{\mathcal{C}(\mathcal{X}^2)}\Big)^2$. *Then,*

$$\mathbb{E}_{\boldsymbol{\phi}}\Big\|k^{(\tilde{f}|\boldsymbol{y})} - k^{(f|\boldsymbol{y})}\Big\|_{\mathcal{C}(\mathcal{X}^2)} \leq C_2 \mathbb{E}_{\boldsymbol{\phi}}\Big\|k^{(\tilde{f})} - k\Big\|_{\mathcal{C}(\mathcal{X}^2)}. \tag{3.45}$$

Moreover, when $\tilde{f}$ *is a random Fourier feature approximation of the prior, it follows that*

$$\mathbb{E}_{\boldsymbol{\phi}}\Big\|k^{(\tilde{f}|\boldsymbol{y})} - k^{(f|\boldsymbol{y})}\Big\|_{\mathcal{C}(\mathcal{X}^2)} \leq \ell^{-1/2} C_2 C_3, \tag{3.46}$$

where C_3 *is one of several possible constants given by Sutherland and Schneider (2015).*

Proof Let $M_k : \mathcal{C}(\mathcal{X} \times \mathcal{X}) \to \mathcal{C}(\mathcal{X} \times \mathcal{X})$ be the bounded linear operator given by

$$(M_k c)(\boldsymbol{x}, \boldsymbol{x}') = c(\boldsymbol{x}, \boldsymbol{x}') - c(\boldsymbol{x}, \mathbf{X}_n)\boldsymbol{\xi}(\boldsymbol{x}') - \boldsymbol{\xi}(\boldsymbol{x})^\top c(\mathbf{X}_n, \boldsymbol{x}') + \boldsymbol{\xi}(\boldsymbol{x})^\top c(\mathbf{X}_n, \mathbf{X}_n)\boldsymbol{\xi}(\boldsymbol{x}'). \tag{3.47}$$

Henceforth, we omit the subscript from M_k to ease notation. Note that, by construction,

$$k^{(f|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}') = (Mk)(\boldsymbol{x}, \boldsymbol{x}') \qquad k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}') = (Mk^{(\tilde{f})})(\boldsymbol{x}, \boldsymbol{x}'). \tag{3.48}$$

Focusing on the integrand on the left-hand side of (3.45), we begin by separating

out the operator norm $\|M\|_{L(\mathcal{C}(\mathcal{X}^2);\mathcal{C}(\mathcal{X}^2))}$ as

$$\left\|k^{(\tilde{f}|\boldsymbol{y})} - k^{(f|\boldsymbol{y})}\right\|_{\mathcal{C}(\mathcal{X}^2)} = \left\|Mk^{(\tilde{f})} - Mk\right\|_{\mathcal{C}(\mathcal{X}^2)} \leq \|M\|_{L(\mathcal{C}(\mathcal{X}^2);\mathcal{C}(\mathcal{X}^2))}\left\|k^{(\tilde{f})} - k\right\|_{\mathcal{C}(\mathcal{X}^2)}. \tag{3.49}$$

Refining this inequality requires us to upper bound $\|M\|_{L(\mathcal{C}(\mathcal{X}^2);\mathcal{C}(\mathcal{X}^2))}$. To do so, we write

$$\|Mc\|_{\mathcal{C}(\mathcal{X}^2)} \leq \|c\|_{\mathcal{C}(\mathcal{X}^2)} + 2\|c(\,\cdot\,,\mathbf{X}_n)\boldsymbol{\xi}(\,\cdot\,)\|_{\mathcal{C}(\mathcal{X}^2)} + \left\|\boldsymbol{\xi}(\,\cdot\,)^\top c(\mathbf{X}_n,\mathbf{X}_n)\boldsymbol{\xi}(\,\cdot\,)\right\|_{\mathcal{C}(\mathcal{X}^2)}. \tag{3.50}$$

We now use Hölder's inequality ($p = 1$, $q = \infty$) followed by the definition of the operator norm $\|\,\cdot\,\|_{L(\ell^\infty;\ell^1)}$ to bound the second and third terms on the right as

$$\begin{aligned}\|c(\,\cdot\,,\mathbf{X}_n)\boldsymbol{\xi}(\,\cdot\,)\|_{\mathcal{C}(\mathcal{X}^2)} &= \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}[c(\boldsymbol{x},\mathbf{X}_n)\boldsymbol{\xi}(\boldsymbol{x}')] \\ &\leq \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\left[\|c(\boldsymbol{x},\mathbf{X}_n)\|_{\ell^\infty}\left\|\mathbf{K}_{n,n}^{-1}\right\|_{L(\ell^\infty;\ell^1)}\|k(\mathbf{X}_n,\boldsymbol{x}')\|_{\ell^\infty}\right] \\ &\leq \|c\|_{\mathcal{C}(\mathcal{X}^2)}\left\|\mathbf{K}_{n,n}^{-1}\right\|_{L(\ell^\infty;\ell^1)}\|k\|_{\mathcal{C}(\mathcal{X}^2)}\end{aligned} \tag{3.51}$$

and

$$\left\|\boldsymbol{\xi}(\,\cdot\,)^\top c(\mathbf{X}_n,\mathbf{X}_n)\boldsymbol{\xi}(\,\cdot\,)\right\|_{\mathcal{C}(\mathcal{X}^2)} \leq n\|c\|_{\mathcal{C}(\mathcal{X}^2)}\left\|\mathbf{K}_{n,n}^{-1}\right\|^2_{L(\ell^\infty;\ell^1)}\|k\|^2_{\mathcal{C}(\mathcal{X}^2)}. \tag{3.52}$$

Returning to (3.50), we may now bound $\|Mc\|_{\mathcal{C}(\mathcal{X}^2)}$ by writing

$$\begin{aligned}\|Mc\|_{\mathcal{C}(\mathcal{X}^2)} &\leq \|c\|_{\mathcal{C}(\mathcal{X}^2)}\left(1 + 2\left\|\mathbf{K}_{n,n}^{-1}\right\|_{L(\ell^\infty;\ell^1)}\|k\|_{\mathcal{C}(\mathcal{X}^2)} + n\left\|\mathbf{K}_{n,n}^{-1}\right\|^2_{L(\ell^\infty;\ell^1)}\|k\|^2_{\mathcal{C}(\mathcal{X}^2)}\right) \\ &\leq \|c\|_{\mathcal{C}(\mathcal{X}^2)}\left(n\left[1 + \left\|\mathbf{K}_{n,n}^{-1}\right\|_{L(\ell^\infty;\ell^1)}\|k\|_{\mathcal{C}(\mathcal{X}^2)}\right]^2\right),\end{aligned} \tag{3.53}$$

which immediately implies that

$$\|M\|_{L(\mathcal{C}(\mathcal{X}^2);\mathcal{C}(\mathcal{X}^2))} = \sup_{c\neq 0}\frac{\|Mc\|_{\mathcal{C}(\mathcal{X}^2)}}{\|c\|_{\mathcal{C}(\mathcal{X}^2)}} \leq n\left[1 + \left\|\mathbf{K}_{n,n}^{-1}\right\|_{L(\ell^\infty;\ell^1)}\|k\|_{\mathcal{C}(\mathcal{X}^2)}\right]^2. \tag{3.54}$$

Note that, since this bound is independent of the particular realization of the ℓ-dimensional random Fourier basis $\boldsymbol{\phi}$ used to construct the approximate prior $\tilde{f}$, it is constant with respect to the expectation (3.45). Finally, Sutherland and Schneider (2015) have shown that there exists a constant C_3 satisfying

$$\mathbb{E}_{\boldsymbol{\phi}}\left\|k^{(\tilde{f})} - k\right\|_{\mathcal{C}(\mathcal{X}^2)} \leq \ell^{-1/2}C_3. \tag{3.55}$$

Combining this inequality with the preceding ones gives the result. ■

Together, Propositions 3.18 and 3.19 show that error in the approximate prior $\tilde{f}$ controls the error in the resulting approximate posterior $\tilde{f} \mid \boldsymbol{y}$. These bounds are not tight, seeing as constants C_1 and C_2 both depend on the number of observations n. Based on this observation, it is tempting to think that the error in $\tilde{f} \mid \boldsymbol{y}$ therefore increases in n. Empirically, however, the opposite trend is observed: the error in $\tilde{f} \mid \boldsymbol{y}$ actually diminishes as n grows (Wilson et al., 2020). To better understand this behavior, we now study the conditions under which a pathwise update may counteract the error introduced by an approximate prior.

3.5.2 Contraction of approximate posteriors with noise-free observations

This section formalizes the following syllogism: (i) the true posterior $f \mid \boldsymbol{y}$ and the approximate posterior $\tilde{f} \mid \boldsymbol{y}$ have the same mean; (ii) as n increases, both posteriors contract to their respective means; (iii) therefore, as n increases, the error introduced by the approximate prior $\tilde{f}$ washes out.

To begin, let $\boldsymbol{\phi} : \mathcal{M} \to \mathbb{R}^\ell$ be an ℓ-dimensional feature map on an ambient space $\mathcal{M}$ consisting of linearly independent basis functions ϕ_i. We will say that $\tilde{f}$ is a *standard normal Bayesian linear model* if it admits the representation

$$\tilde{f}(\,\cdot\,) = \sum_{i=1}^{\ell} w_i \phi_i(\,\cdot\,) \qquad\qquad w_i \sim \mathcal{N}(0,1). \tag{3.56}$$

This description includes the Karhunen–Loève and Fourier feature approximations described in Section 3.3. As before, let $\boldsymbol{\Phi}_n = \boldsymbol{\phi}(\mathbf{X}_n)$ be an $n \times \ell$ feature matrix and $\mathcal{H}_k$ be the reproducing kernel Hilbert space associated with a kernel k. We say that a function ϕ_i *lies locally* in $\mathcal{H}_k$ for a compact $\mathcal{X} \subseteq \mathcal{M}$ if there exists a function $\psi_j \in \mathcal{H}_k$ that agrees with ϕ_i on $\mathcal{X}$, i.e. $\phi_i\big|_{\mathcal{X}} = \psi_j\big|_{\mathcal{X}}$.

When $\mathcal{M}$ is a compact metric space, the eigenfunctions ϕ_i used to construct (truncated) Karhunen–Loève expansions belong to $\mathcal{H}_k$ by construction. More generally, assessing whether or not ϕ_i lies locally in $\mathcal{H}_k$ is often straightforward for kernels with known reproducing kernel Hilbert spaces. As a concrete example, the RKHS of a Matérn-$\nu/2$ kernel is the Sobolev space of order $\kappa = \nu + d/2$. For integer values of κ, this is the space of square-integrable functions with κ square-integrable derivatives. Trigonometric basis functions $\phi_i(\boldsymbol{x}) = \cos(2\pi \boldsymbol{\omega}_i^\top \boldsymbol{x} + \tau_i)$ can readily be adapted to satisfy this requirement. Specifically, we may multiply them by a suitably chosen, infinitely-differentiable function that ensures they decay to zero outside of $\mathcal{X}$, such that the resulting basis functions (and their derivatives) are square-integrable.

We are now ready to state and prove the primary claim. In the following, Proposition 3.20 and Corollary 3.21 will demonstrate that $\tilde{f} \mid \boldsymbol{y}$ contracts at the same rate as $f \mid \boldsymbol{y}$. Subsequently, Corollary 3.22 will show that the error in $\tilde{f} \mid \boldsymbol{y}$ vanishes as $n \to \infty$ in any reasonable limit where the variance of the true posterior contracts to zero everywhere on $\mathcal{X}$.

Proposition 3.20. *Suppose $\mathcal{X} \subseteq \mathcal{M}$ is compact and that each of the ℓ basis functions ϕ_i used to construct the standard normal Bayesian linear model $\tilde{f}$ lies locally in $\mathcal{H}_k$. If the points $\mathbf{X}_n \subset \mathcal{X}$ used to condition the approximate posterior $\tilde{f} \mid \boldsymbol{y}$ are chosen such that $f \mid \boldsymbol{y}$ satisfies $\sup_{\boldsymbol{x}\in\mathcal{X}} k^{(f|\boldsymbol{y})}(\boldsymbol{x},\boldsymbol{x}) \leq \varepsilon$, then it follows that*[11]

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \left| k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x},\boldsymbol{x}) \right| \leq C_4 \varepsilon, \tag{3.57}$$

where we have defined $C_4 = \ell \max_i \inf \left\{ \|\psi_i\|^2_{\mathcal{H}_k} : \psi_i|_{\mathcal{X}} = \phi_i|_{\mathcal{X}}, \forall \psi_i \in \mathcal{H}_k \right\}$.

Proof Recall from (3.40) that we may use the weight function

$$\boldsymbol{\xi}(\,\cdot\,) = k(\mathbf{X}_n, \mathbf{X}_n)^{-1} k(\mathbf{X}_n, \,\cdot\,)$$

(3.58)

to express the approximate posterior as $(\tilde{f} \mid \boldsymbol{y})(\,\cdot\,) \overset{\mathrm{d}}{=} \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w} - \boldsymbol{\xi}(\,\cdot\,)^\top (\boldsymbol{y} - \boldsymbol{\Phi}_n \boldsymbol{w})$. Under this notation, it is clear that we may immediately upper bound the variance of the $\tilde{f} \mid \boldsymbol{y}$ as

$$\mathrm{Var}\Big((\tilde{f} \mid \boldsymbol{y})(\,\cdot\,)\Big) = \mathbb{E}\Big[\Big(\boldsymbol{\phi}(\,\cdot\,)^\top - \boldsymbol{\xi}(\,\cdot\,)^\top \boldsymbol{\Phi}_n\Big)\boldsymbol{w}\Big]^2 \leq \ell \max_i \Big(\phi_i(\,\cdot\,) - \boldsymbol{\xi}(\,\cdot\,)^\top \phi_i(\mathbf{X}_n)\Big)^2, \tag{3.59}$$

where, on the right, we have used the fact that $\mathbb{E}\|\boldsymbol{w}\|^2 = \ell$. By further denoting $\mathcal{G} = \{g \in \mathcal{H}_k : \|g\|_{\mathcal{H}_k} = 1\}$, we may now exploit the dual representation of the RKHS norm to write

$$\begin{aligned} \left|\phi_i(\boldsymbol{x}_*) - \boldsymbol{\xi}(\boldsymbol{x}_*)^\top \phi_i(\mathbf{X}_n)\right| &\leq \|\phi_i\|_{\mathcal{H}_k} \sup_{g\in\mathcal{G}} \left| g(\boldsymbol{x}_*) - \boldsymbol{\xi}(\boldsymbol{x}_*)^\top g(\mathbf{X}_n) \right| \\ &= \|\phi_i\|_{\mathcal{H}_k} \left\| k(\,\cdot\,, \boldsymbol{x}_*) - \boldsymbol{\xi}(\boldsymbol{x}_*)^\top \mathbf{K}_{n,*} \right\|_{\mathcal{H}_k} \\ &= \|\phi_i\|_{\mathcal{H}_k} \underbrace{\sqrt{k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \mathbf{K}_{*,n} \mathbf{K}_{n,n}^{-1} \mathbf{K}_{n,*}}}_{\mathcal{P}_{\mathbf{X}}(\boldsymbol{x}_*)}, \end{aligned} \tag{3.60}$$

where, because ϕ_i lies locally in $\mathcal{H}_k$, we may replace it with any $\psi_i \in \mathcal{H}_k : \ \psi_i|_{\mathcal{X}} = \phi_i|_{\mathcal{X}}$. Noting that $\mathcal{P}_{\mathbf{X}}(\,\cdot\,) = \sqrt{k^{(f|\boldsymbol{y})}(\,\cdot\,,\,\cdot\,)}$ and collecting terms gives the result. ■

Corollary 3.21. *With the same assumptions, as $\sup_{\boldsymbol{x}\in\mathcal{X}} k^{(f|\boldsymbol{y})}(\boldsymbol{x},\boldsymbol{x}) \to 0$, it follows that*

$$\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}} \left| k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x},\boldsymbol{x}') - k^{(f|\boldsymbol{y})}(\boldsymbol{x},\boldsymbol{x}') \right| \to 0. \tag{3.61}$$

Proof Begin by applying the triangle inequality to the above and, subsequently, use

[11] This result holds even when the weights are not assumed i.i.d., albeit with a different constant.

the Cauchy-Schwarz inequality to bound $k(\boldsymbol{x}, \boldsymbol{x}') \leq \sqrt{k(\boldsymbol{x}, \boldsymbol{x})}\sqrt{k(\boldsymbol{x}', \boldsymbol{x}')}$, which gives

$$\begin{aligned}\sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\left|k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}') - k^{(f|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}')\right| &\leq \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\left|k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}')\right| + \sup_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{X}}\left|k^{(f|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}')\right| \\ &\leq \sup_{\boldsymbol{x}\in\mathcal{X}}\left|k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x})\right| + \sup_{\boldsymbol{x}\in\mathcal{X}}\left|k^{(f|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x})\right|.\end{aligned} \tag{3.62}$$

In the final expression, convergence of the former term is given by Proposition 3.20, while the latter goes to zero by assumption. ■

Corollary 3.22. *With the same assumptions, as* $\sup_{\boldsymbol{x}\in\mathcal{X}} k^{(f|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x}) \to 0$*, it follows that*

$$W_{2,L^2(\mathcal{X})}(f \mid \boldsymbol{y}, \tilde{f} \mid \boldsymbol{y}) \to 0. \tag{3.63}$$

Proof Since $L^2(\mathcal{X})$ is a normed space and $\mathbb{E}(f \mid \boldsymbol{y}) = \mathbb{E}(\tilde{f} \mid \boldsymbol{y})$, we have that

$$W_{2,L^2(\mathcal{X})}(f \mid \boldsymbol{y}, \tilde{f} \mid \boldsymbol{y}) = W_{2,L^2(\mathcal{X})}\Bigg(\underbrace{f(\,\cdot\,) - \boldsymbol{\xi}(\,\cdot\,)^\top f(\mathbf{X}_n)}_{(f|\boldsymbol{y})_0}, \underbrace{\boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w} - \boldsymbol{\xi}(\,\cdot\,)^\top \boldsymbol{\Phi}_n \boldsymbol{w}}_{(\tilde{f}|\boldsymbol{y})_0}\Bigg), \tag{3.64}$$

where $(f \mid \boldsymbol{y})_0$ and $(\tilde{f} \mid \boldsymbol{y})_0$ denote centered processes. Now, let $\mathbb{0}$ be an almost surely zero stochastic process over $\mathcal{X}$. Then, by the triangle inequality,

$$W_{2,L^2(\mathcal{X})}\Big((f \mid \boldsymbol{y})_0, (\tilde{f} \mid \boldsymbol{y})_0\Big) \leq W_{2,L^2(\mathcal{X})}\Big((f \mid \boldsymbol{y})_0, \mathbb{0}\Big) + W_{2,L^2(\mathcal{X})}\Big((\tilde{f} \mid \boldsymbol{y})_0, \mathbb{0}\Big). \tag{3.65}$$

Expanding the definition of Wasserstein distances $W_{2,L^2(\mathcal{X})}$ before using Tonelli's theorem to change the order of integration gives

$$\begin{aligned}W_{2,L^2(\mathcal{X})}\Big((f \mid \boldsymbol{y})_0, (\tilde{f} \mid \boldsymbol{y})_0\Big) &\leq \Big(\mathbb{E}\big\|(f \mid \boldsymbol{y})_0 - \mathbb{0}\big\|^2_{L^2(\mathcal{X})}\Big)^{1/2} + \Big(\mathbb{E}\big\|(\tilde{f} \mid \boldsymbol{y})_0 - \mathbb{0}\big\|^2_{L^2(\mathcal{X})}\Big)^{1/2} \\ &= \Big(\int_{\mathcal{X}} k^{(f|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x})\, d\boldsymbol{x}\Big)^{1/2} + \Big(\int_{\mathcal{X}} k^{(\tilde{f}|\boldsymbol{y})}(\boldsymbol{x}, \boldsymbol{x})\, d\boldsymbol{x}\Big)^{1/2},\end{aligned} \tag{3.66}$$

where both terms in the final expression converge to zero by compactness of $\mathcal{X}$ together with Proposition 3.20. ■

Together, these claims demonstrate that the decoupled approximate posterior $\tilde{f} \mid \boldsymbol{y}$, formed by using the canonical basis $k(\,\cdot\,, \mathbf{X}_n)$ to update a well-specified approximate prior $\tilde{f}$, *inherits* the contractive properties of the true posterior $f \mid \boldsymbol{y}$.

Per the beginning of this section, approximate priors $\tilde{f}$ defined as standard normal Bayesian linear models with basis functions that lie locally in $\mathcal{H}_k$ are well-specified. The following counterexample helps clarify what can happen when $\tilde{f}$ is misspecified. Consider an approximate prior $\tilde{f} \sim \mathcal{GP}(0, \delta)$ equipped with the Kronecker delta kernel δ such that $\mathrm{Cov}\Big(\tilde{f}(\boldsymbol{x}_i), \tilde{f}(\boldsymbol{x}_j)\Big) = 1$ if $\boldsymbol{x}_i = \boldsymbol{x}_j$ and 0 otherwise. Given a finite

set of test locations $\mathbf{X}_* \subset \mathcal{X} \setminus \mathbf{X}_n$, let $\boldsymbol{\Xi} = \boldsymbol{\xi}(\mathbf{X}_*)^\top$. Applying the pathwise update (3.17) to $\tilde{f}$, the posterior covariance is then

$$\mathrm{Cov}\Big(\tilde{\boldsymbol{f}}_* \mid \boldsymbol{y}\Big) = \mathrm{Cov}\Big(\tilde{\boldsymbol{f}}_*\Big) + \boldsymbol{\Xi}\,\mathrm{Cov}\Big(\tilde{\boldsymbol{f}}_n\Big)\boldsymbol{\Xi}^\top - 2\,\mathrm{Cov}\Big(\tilde{\boldsymbol{f}}_*, \tilde{\boldsymbol{f}}_n\Big)\boldsymbol{\Xi}^\top = \mathbf{I} + \mathbf{K}_{*,n}\mathbf{K}_{n,n}^{-2}\mathbf{K}_{n,*}. \tag{3.67}$$

Since the second of the two terms on the right is guaranteed non-negative, the variance of the resulting posterior is bounded from below by 1. For this choice of $\tilde{f}$, then, the approximation error inherent to $\tilde{f} \mid \boldsymbol{y}$ does not diminish as n increases.[12]

3.5.3 Sparse approximation errors

We now examine the error introduced by using a sparse pathwise update (3.31) to construct an approximate posterior. As notation, we write $f \mid \boldsymbol{u}$ and $\tilde{f} \mid \boldsymbol{u}$ for the approximate posteriors formed by applying the sparse update to the true prior f and to the approximate prior $\tilde{f}$, respectively. Results discussed here mirror those presented by Wilson et al. (2020). Appealing to the triangle inequality, we have

$$\begin{aligned} W_{2,L^2(\mathcal{X})}\Big(\tilde{f} \mid \boldsymbol{u}, f \mid \boldsymbol{y}\Big) &\leq \underbrace{W_{2,L^2(\mathcal{X})}\Big(\tilde{f} \mid \boldsymbol{u}, f \mid \boldsymbol{u}\Big)}_{\text{error in approximate prior}} + \underbrace{W_{2,L^2(\mathcal{X})}\Big(f \mid \boldsymbol{u}, f \mid \boldsymbol{y}\Big)}_{\text{error in sparse update}} \\ \mathbb{E}_{\boldsymbol{\phi}}\Big\|k^{(\tilde{f}\mid\boldsymbol{u})} - k^{(f\mid\boldsymbol{y})}\Big\|_{\mathcal{C}(\mathcal{X}^2)} &\leq \overbrace{\mathbb{E}_{\boldsymbol{\phi}}\Big\|k^{(\tilde{f}\mid\boldsymbol{u})} - k^{(f\mid\boldsymbol{u})}\Big\|_{\mathcal{C}(\mathcal{X}^2)}} + \overbrace{\Big\|k^{(f\mid\boldsymbol{u})} - k^{(f\mid\boldsymbol{y})}\Big\|_{\mathcal{C}(\mathcal{X}^2)}} \quad . \end{aligned} \tag{3.68}$$

From here, any of the previously presented propositions enable us to control the total error. For the first terms on the right, the same arguments as before lead to the same results; however, the constants involved will change, since the sparse update now assumes the role of the canonical one. The latter terms do not involve the approximate prior and are therefore beyond the scope of our present analysis. Note that similar statements hold for the Gaussian pathwise update (3.29).

As a final remark, note that we may reduce the total error (3.68) by incorporating additional basis functions $k(\,\cdot\,, \mathbf{X})$ into the sparse update. Conceptually, the act of *augmenting* a sparse update amounts to replacing $\boldsymbol{u} \sim q(\boldsymbol{u})$ with $\boldsymbol{u}' \sim q(\boldsymbol{u}') = p(\boldsymbol{f} \mid \boldsymbol{u})q(\boldsymbol{u})$, where $\boldsymbol{f}$ are process values at centers $\mathbf{X}$ (Rasmussen and Quiñonero-Candela, 2005; Quiñonero-Candela et al., 2007). By construction, $q(\boldsymbol{u})$ and $q(\boldsymbol{u}')$ induce the same posterior on f. However, because the augmented update utilizes additional basis functions, the error in the induced distribution of $\tilde{\boldsymbol{f}}_*$ diminishes. This result follows from the same line of reasoning as before: since $\mathbb{E}\Big(\boldsymbol{f}_* \mid \boldsymbol{u}'\Big) = \mathbb{E}\Big(\tilde{\boldsymbol{f}}_* \mid \boldsymbol{u}'\Big)$, $f \mid \boldsymbol{u}'$ and $\tilde{f} \mid \boldsymbol{u}'$ contract to the same function as $|\boldsymbol{u}'| \to \infty$. Hence, the approximate prior washes out and the total error decreases.

[12] Contraction of the true posterior is well-studied and has strong ties to the literature on kernel methods. Kanagawa et al. (2018) reviews these connections in greater detail: there, Theorem 5.4 shows how the *power function* $\mathcal{P}_{\mathbf{X}}$ can be bounded in terms of the fill distance $h(\mathbf{X}_n) = \sup_{\boldsymbol{x}_* \in \mathcal{X}} \inf_{\boldsymbol{x} \in \mathbf{X}_n} \|\boldsymbol{x}_* - \boldsymbol{x}\|$.

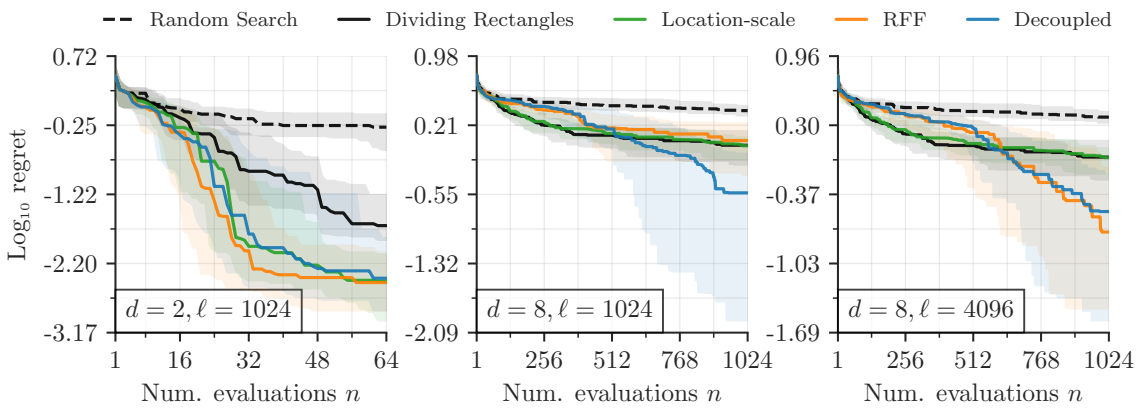

Figure 3.6: Median performances and interquartile ranges of Thompson sampling methods and popular baselines when optimizing function draws from known GP priors on $d = \dim(\mathcal{X})$ dimensional domains. Location-scale Thompson sampling performs well in low-dimensional settings (left), but struggles as d increase due to its inability to efficiently utilize gradient information. RFF posteriors enable us to generate function draws, but demand many more basis functions $b = \ell + n$ than data points n (middle vs. right). Decoupled approaches using canonical basis functions $k(\,\cdot\,, \boldsymbol{x})$ to update RFF priors $\tilde{f}$ avoids these pitfalls and consistently match or outperform competing strategies.

3.6 Applications

This section examines the practical consequences of pathwise conditioning in terms of a curated set of representative tasks. Throughout, we focus on how pathwise methods for efficiently generating function draws from GP posteriors enable us to overcome common obstacles and open doors for new research. We provide a general framework for pathwise conditioning of Gaussian processes based on GPflow (Matthews et al., 2017).[13]

3.6.1 Optimizing black-box functions

Global optimization revolves around the challenge of efficiently identifying a global minimizer

$$\boldsymbol{x}_{\min} \in \mathbf{X}_{\min} \qquad\qquad \mathbf{X}_{\min} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \tag{3.69}$$

of a black-box function $f : \mathcal{X} \to \mathbb{R}$. Since f is a black box, our understanding of its behavior is limited to a set of observations $\boldsymbol{y}$ at locations $\mathbf{X}_n$. Gaussian processes are a natural and widely used way of representing possible functions $f \mid \boldsymbol{y}$ (Močkus, 1975; Srinivas et al., 2010b; Frazier, 2018). In these cases, we reason about global minimizers (3.69) in terms of a belief over the random set

$$\mathbf{X}_{\min}^{(f \mid \boldsymbol{y})} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} (f \mid \boldsymbol{y})(\boldsymbol{x}). \tag{3.70}$$

[13]Code is available online at `https://github.com/j-wilson/GPflowSampling`.

Approaches to these problems are often characterized as striking a balance between two competing agendas: the need to learn about the function's global behavior by *exploring* the domain $\mathcal{X}$ and the need to obtain (potentially local) minimizers by *exploiting* what is already known.

Thompson sampling is a classic decision-making strategy that balances the tradeoff between exploration and exploitation by sampling actions $\boldsymbol{x} \in \mathcal{X}$ in proportion to the probability that $\boldsymbol{x} \in \mathbf{X}_{\min}^{(f|\boldsymbol{y})}$ (Thompson, 1933). At first glance, this task may seem daunting, since $\mathbf{X}_{\min}^{(f|\boldsymbol{y})}$ is random. For a given draw of $f \mid \boldsymbol{y}$, however, $\mathbf{X}_{\min}^{(f|\boldsymbol{y})}$ is deterministic. Accordingly, we may Thompson sample an action by generating a function $f \mid \boldsymbol{y}$ and, subsequently, finding a pathwise global minimizer.

Thompson sampling's relative simplicity makes it a natural test bed for evaluating different sampling strategies, while its real-world performance (Chapelle and Li, 2011) assures its ongoing relevance in applied settings. A key strength of these methods is that they support embarassingly-parallel batch selection (Hernández-Lobato et al., 2017; Kandasamy et al., 2018). While many GP-based search strategies allow us to choose $\kappa > 1$ queries at a time (Snoek et al., 2012; Wilson et al., 2018), their compute costs tend to scale aggressively in κ. Especially when evaluations can be carried out in parallel, then, Thompson sampling provides an affordable alternative to comparable approaches.

We considered three different variants of Thompson sampling, corresponding with different approaches to sampling from GP posteriors. The first approach samples random vectors $\boldsymbol{f}_* \mid \boldsymbol{y}$ using location-scale transforms (3.19); the second approximates posteriors with Bayesian linear models; and, the third updates function draws from ℓ-dimensional approximate priors $\tilde{f} = \boldsymbol{\phi}(\,\cdot\,)^\top \boldsymbol{w}$ using canonical basis functions centered at the n training locations.[14] For fair comparison, we allocate $b = \ell + n$ random Fourier basis functions to Bayesian linear models employed by the second approach.

At each round of Thompson sampling, we began by sampling process values $f_i \mid \boldsymbol{y}$ independently on a randomly generated discretization of $\mathcal{X} = [0, 1]^d$. Next, we constructed a candidate set $\mathbf{X}_*$ using the locations that produce the smallest realizations of $f_i \mid \boldsymbol{y}$. Under a location-scale approach, we then jointly sampled process values at $|\mathbf{X}_*| = 2048$ candidates. For both of the alternatives, we instead used $|\mathbf{X}_*| = 32$ candidates to initialize multi-start gradient descent. In all three cases, queries were chosen as minimizers of the resulting vector $\boldsymbol{f}_* \mid \boldsymbol{y}$. Batches of queries were obtained using κ independent runs of this algorithm.

To eliminate confounding variables, we experimented with black-box functions drawn from a known Matérn-$5/2$ prior with an isotropic length scale $l = \sqrt{d/100}$ and Gaussian observations $y \sim \mathcal{N}(f(\boldsymbol{x}), 10^{-3})$. We set $\kappa = d$, but this choice was not found to significantly influence our results. Below, we focus on comparing each Thompson sampling variant's behavior for different amounts of design variables d and basis functions ℓ.

Figure 3.6 reports key findings based on 32 independent trials; for extended results,

[14] Equation (3.35) highlights the difference between the second and third approaches.

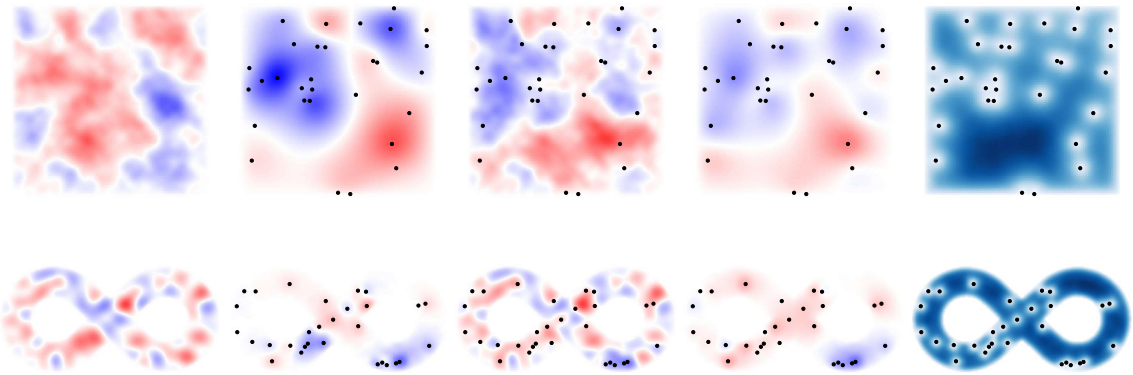

Figure 3.7: Pathwise conditioning of samples from Matérn priors subject to observations $\boldsymbol{y}$ (black dots) and Dirichlet boundary conditions $f\big|_{\partial\mathcal{X}} = 0$. From left to right, the first three columns show a draw from the prior, a pathwise update, and the corresponding realization of the posterior. The final two columns communicate the empirical mean and standard deviation of the posterior, respectively. *Top:* Illustration of a rectangular domain for which Laplacian eigenpairs are calculated analytically. *Bottom:* A non-trivial domain for which the eigenpairs are approximated numerically.

see Wilson et al. (2020). First, location-scale methods' inability to use gradient information to efficiently find pathwise minimizers causes its performance to wane as d increases. In contrast, both of the alternative variants of Thompson sampling rely on pathwise-differentiable function draws and, therefore, scale more gracefully in d. Second, RFF-based Bayesian linear models struggle to represent posteriors due to variance starvation (Section 3.3.5). As the number of observations n increases relative to the number of basis functions $b = \ell + n$, the function draws they produce come to inadequately characterize the true posterior, causing Thompson sampling to falter. Decoupled approaches to updating $\tilde{f}$ avoid this issue by, e.g., associating the data with the n-dimensional canonical basis $k(\,\cdot\,, \mathbf{X}_n)$.

3.6.2 Generating boundary-constrained sample paths

This section illustrates how techniques introduced in the preceding sections can be used to efficiently sample Gaussian process posteriors subject to boundary conditions (Solin and Kok, 2019). Whittle (1963) showed that a Matérn GP f defined over $\mathbb{R}^d$ satisfies the stochastic partial differential equation

$$\left(\frac{2\nu}{\kappa^2} - \Delta\right)^{\frac{\nu}{2}+\frac{d}{4}} f = \mathcal{W}, \tag{3.71}$$

where $\mathcal{W}$ is a (rescaled) white noise process, and Δ is the Laplacian. Following Solin and Kok (2019) and Rue and Held (2005), we restrict (3.71) onto a (well-behaved) compact domain $\mathcal{X} \subset \mathbb{R}^d$ and impose Dirichlet boundary conditions $f\Big|_{\partial\mathcal{X}} = 0$ to define a boundary-constrained Matérn Gaussian process over $\mathcal{X}$. Solin and Kok

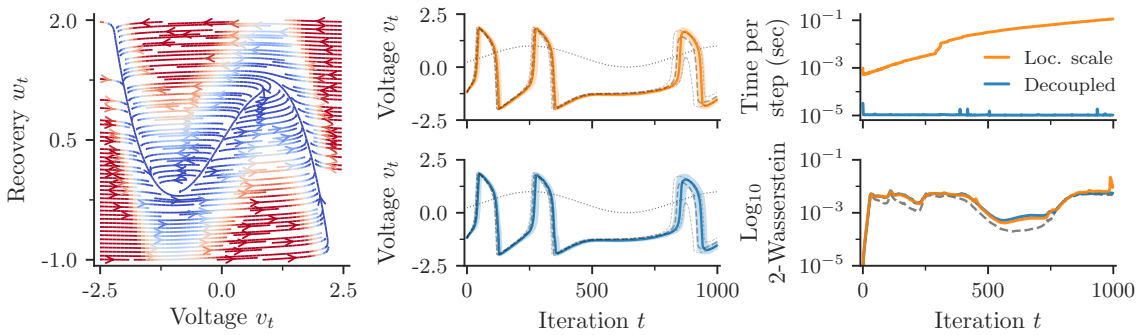

Figure 3.8: Model-based simulations of a stochastic FitzHugh–Nagumo neuron. *Left:* Phase portrait of the true drift function subject to a fixed current $a = 0.5$. *Middle:* Empirical medians and interquartile ranges of simulated voltage traces driven by a sinusoidal current (dotted black); ground truth quartiles are shown in dashed gray. Trajectories generated via location-scale transforms are summarized on the top in orange, while those produced by decoupled drift functions are portrayed on the bottom in blue. *Top right:* Comparison of simulation runtimes. *Bottom right:* Sinkhorn estimates (Cuturi, 2013) to 2-Wasserstein distances between model-based and ground truth state distributions at each step t. The noise floor (dashed gray) was found using additional ground truth simulations.

(2019) demonstrate that such a prior admits the Karhunen–Loève expansion

$$f(\,\cdot\,) = \sum_{i=1}^{\infty} w_i \phi_i(\,\cdot\,) \qquad\qquad w_i \sim \mathcal{N}\left(0, \frac{\sigma^2}{C_\nu}\left(\frac{2\nu}{\kappa^2} + \lambda_i\right)^{-\nu-\frac{d}{2}}\right), \tag{3.72}$$

where ϕ_i are eigenfunctions of the *boundary-constrained* Laplacian. We truncate this expansion to obtain the ℓ-dimensional Bayesian linear model $\tilde{f}$, which we use together with a pathwise update to construct the posterior.

Figure 3.7 visualizes function draws from boundary-constrained priors and posterior for two choices of boundaries on $\mathbb{R}^2$, a rectangle and the symbol for infinity. Note that eigenfunctions for rectangular regions of Euclidean domains are available analytically, while those of the infinity symbol are obtained numerically by solving a Helmholtz equation. Examining this figure, we see that the sample paths respect the Dirichlet boundary condition $f\big|_{\partial\mathcal{X}} = 0$. Karhunen–Loève expansions enable boundary-constrained GPs, an important class of non-stationary priors, to be used within the pathwise conditioning framework.

3.6.3 Simulating dynamical systems

Gaussian process posteriors are commonly used to simulate complex, real-world phenomena in cases where we are unable to actively collect additional data. These phenomena include dynamical systems that describe how physical states evolve over time.

We focus on cases where a Gaussian process prior is placed on the *drift* $f : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ of a time-invariant system, which maps from a state vector $\boldsymbol{x}_t \in \mathcal{X}$ and a control input $\boldsymbol{a}_t \in \mathcal{A}$ to a tangent vector $\boldsymbol{f}_t \in \mathcal{X}$. Using an Euler–Maruyama scheme to

discretize the dynamical system's equations of motion, we obtain the stochastic difference equation (SDE)

$$\boldsymbol{x}_{t+1} - \boldsymbol{x}_t = \tau f(\boldsymbol{x}_t, \boldsymbol{a}_t) + \sqrt{\tau}\boldsymbol{\varepsilon}_t = \boldsymbol{y}_t \qquad \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_\varepsilon), \tag{3.73}$$

where τ is the chosen step size and $\boldsymbol{\varepsilon}$ denotes process diffusion. Together with control inputs $\mathbf{A}_T = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_T)$ and diffusion variables $\mathbf{E}_T = (\boldsymbol{\varepsilon}_1, \ldots, \boldsymbol{\varepsilon}_T)$, each draw of f fully characterizes how an initial state $\boldsymbol{x}_1 \sim p(\boldsymbol{x}_1)$ evolves over a series of T successive steps.

Since $\boldsymbol{x}_{t+1}$ depends on $\boldsymbol{x}_t$, strategies for jointly sampling $\mathbf{X}_{T+1} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T+1})$ are typically iterative. Under a distributional approach, we generate $\boldsymbol{x}_{t+1}$ by sampling from the conditional distribution $p(\boldsymbol{y}_t \mid \mathcal{D}_{t-1})$, where $\mathcal{D}_{t-1}$ denotes the union of the real data $(\boldsymbol{x}_i, \boldsymbol{y}_i)_{i=1}^n$ and the current trajectory $(\boldsymbol{x}_j, \boldsymbol{y}_j)_{j=1}^{t-1}$. As mentioned in Section 3.3.1, we may use low-rank matrix updates to efficiently obtain $p(\boldsymbol{y}_t \mid \mathcal{D}_{t-1})$ from $p(\boldsymbol{y}_t \mid \mathcal{D}_{t-2})$ in $\mathcal{O}(t^2)$ time. Nevertheless, the resulting algorithm suffers from $\mathcal{O}(T^3)$ time complexity. In contrast, approaches based on updating of (approximate) prior function draws scale linearly in T.

Many of the same issues were explored by Ialongo et al. (2019), who also proposed a linear-time generative strategy for GP-based trajectories. In the language of the present work, this alternative represents the SDE (3.73) by (i) formulating the unknown drift function as the conditional expectation $\mathbb{E}(f \mid \boldsymbol{u}) = k(\,\cdot\,, \mathbf{Z})\mathbf{K}_{m,m}^{-1}\boldsymbol{u}$ of a sparse Gaussian process f with inducing variables $\boldsymbol{u} \sim q(\boldsymbol{u})$ and (ii) defining process diffusion as the sum of the remaining terms $\boldsymbol{\varepsilon}_t \sim \mathcal{N}\Big(\mathbf{0}, k^{(f|\boldsymbol{u})}(\boldsymbol{x}_t, \boldsymbol{x}_t) + \boldsymbol{\Sigma}_\varepsilon\Big)$. Similar to the pathwise methods put forth here, this approach avoids inter-state dependencies while unrolling by exploiting the fact each draw of $\boldsymbol{u}$ realizes an entire drift function.

To better illustrate the practical implications of pathwise approaches to GP-based simulation, we trained a Gaussian process to represent a stochastic variant of the classic FitzHugh–Naguomo model neuron (FitzHugh, 1961; Nagumo et al., 1962). This model describes a biological neuron in terms of its membrane potential v_t and a recovery variable w_t that summarizes the state of its ion channels. Written in the form (3.73), we have

$$\boldsymbol{x}_{t+1} - \boldsymbol{x}_t = \begin{bmatrix} v_{t+1} - v_t \\ w_{t+1} - w_t \end{bmatrix} = \tau \begin{bmatrix} v_t - \frac{v_t^3}{3} - w_t + a_t \\ \frac{1}{\gamma}(v_t - \beta w_t + \alpha) \end{bmatrix} + \sqrt{\tau}\boldsymbol{\varepsilon}_t, \tag{3.74}$$

where we have chosen $\tau = 0.25$ ms, $\alpha = 0.75$, $\beta = 0.75$, $\gamma = 20$, and $\boldsymbol{\Sigma}_\varepsilon = 10^{-4}\mathbf{I}$. A two-dimensional phase portrait of this system's drift function given a current injection $a = 0.5$ is shown on the left in Figure 3.8.

Training data was generated by evaluating (3.74) for $n = 256$ state-action pairs $(\boldsymbol{x}_i, \boldsymbol{a}_i)$, chosen uniformly at random from $\mathcal{X} = [-2.5, 2.5] \times [-1, 2]$ and $\mathcal{A} = [0, 1]$. Changes in each of the state variables were modeled by independent, Matérn-$5/2$ GPs using $m = 32$ inducing variables. Both sparse GPs were trained by minimizing Kullback–Leibler divergences.

At test time, state trajectories were unrolled from steady state for $T = 1000$ steps

under the influence of a current injection; see middle column of Figure 3.8. Drift values $\boldsymbol{f}_t$ were realized using either the $\mathcal{O}(T^3)$ location-scale technique or the $\mathcal{O}(T)$ pathwise approach. As seen on the right in Figure 3.8, both strategies are capable of accurately characterizing possible state trajectories. At the same time, their difference in cost is striking: the location-scale method spent 10 hours generating 1000 state trajectories (run in parallel), while the pathwise one spent 20 seconds.

3.6.4 Efficiently solving reinforcement learning problems

Model-based approaches to autonomously controlling robotic systems often rely on Gaussian processes to infer system dynamics from a limited number of observations (Rasmussen and Kuss, 2004; Deisenroth et al., 2015; Kamthe and Deisenroth, 2018). Of these data-efficient methods, we focus on PILCO (Deisenroth and Rasmussen, 2011), which is an effective policy search method that uses Gaussian process dynamics models.[15]

Similar to the previous section, we begin by placing a GP prior on the drift function $f : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$ of a black-box dynamical system, now assumed to be deterministic. Rather than being given a sequence of actions $\mathbf{A}_T$ and asked to simulate trajectories $\mathbf{X}_{T+1}$, our new goal will be to find parameters $\boldsymbol{\theta} \in \Theta$ of a deterministic, feedback policy $\pi : \Theta \times \mathcal{X} \to \mathcal{A}$ that maximize the expected cumulative reward

$$R(\boldsymbol{\theta}) = \mathbb{E}_{f,\,\boldsymbol{x}_1}\left[\sum_{t=1}^{T} r\Big(\underbrace{\boldsymbol{x}_t + f(\boldsymbol{x}_t, \pi_{\boldsymbol{\theta}}(\boldsymbol{x}_t))}_{\boldsymbol{x}_{t+1}}\Big)\right] = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{x}_{t+1}}\Big[r(\boldsymbol{x}_{t+1})\Big]. \tag{3.75}$$

For suitably chosen reward functions $r : \mathcal{X} \to \mathbb{R}$, we may optimize $\boldsymbol{\theta}$ by differentiating (3.75). The challenge, however, is to evaluate this expectation in the first place.

The original PILCO algorithm tackles this problem by using moment matching to approximately propagate uncertainty through time. Given a random state $\boldsymbol{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{t,t})$, we begin by supposing that $\boldsymbol{x}_t$ and $\boldsymbol{a}_t = \pi_{\boldsymbol{\theta}}(\boldsymbol{x}_t)$ are jointly normal. Next, we obtain the corresponding optimal Gaussian approximation to $p(\boldsymbol{x}_t, \boldsymbol{a}_t)$ by analytically computing the required moments $\mathbb{E}(\boldsymbol{a}_t)$, $\mathrm{Cov}(\boldsymbol{a}_t, \boldsymbol{a}_t)$, and $\mathrm{Cov}(\boldsymbol{a}_t, \boldsymbol{x}_t)$. This step can also be seen as finding the affine approximation to $\pi_{\boldsymbol{\theta}}$ that best propagates $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{t,t})$. We now use moment matching to propagate this approximate joint distribution through f in order to construct a second Gaussian approximation, this time to $p(\boldsymbol{x}_t, \boldsymbol{f}_t)$.[16] By interpreting $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \boldsymbol{f}_t$ as the sum of jointly Gaussian random variables, we compute the corresponding right-hand side term of (3.75) and, finally, proceed to the next time step. Overall, this strategy works well when f and $\pi_{\boldsymbol{\theta}}$ are sufficiently regular and $\mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_{t,t})$ is sufficiently peaked that maps from $\boldsymbol{x}_t$ to $\boldsymbol{f}_t$ are nearly affine in a ball around $\boldsymbol{\mu}_t$ whose radius is dictated by $\boldsymbol{\Sigma}_{t,t}$.

Here, we are interested in comparing the behavior of moment-based and path-based approaches to optimizing (3.75). To shed light on how these approaches fare in the

[15] PILCO implementation available separately at `https://github.com/j-wilson/GPflowPILCO`.

[16] By appealing to the affine approximation view of moment matching, we obtain the approximate cross-covariance $\mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{f}_t) \approx \mathrm{Cov}(\boldsymbol{x}_t, \boldsymbol{s}_t)\,\mathrm{Cov}(\boldsymbol{s}_t, \boldsymbol{s}_t)^{-1}\,\mathrm{Cov}(\boldsymbol{s}_t, \boldsymbol{f}_t)$ where $\boldsymbol{s}_t = \boldsymbol{x}_t \oplus \boldsymbol{a}_t$.

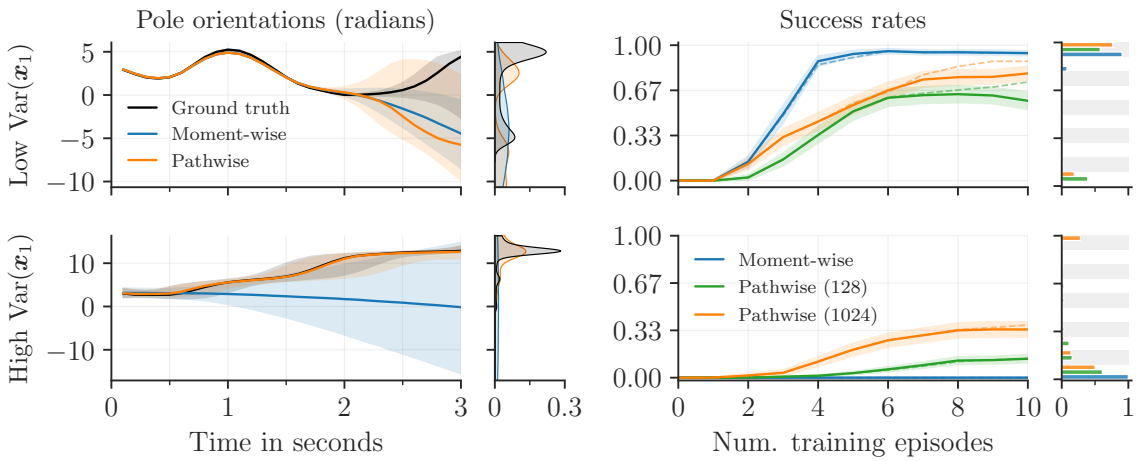

Figure 3.9: Behavior and performance of PILCO algorithms applied to different versions of cart-pole. Marginal distributions of terminal values are shown immediately to the right of each plot. In top and bottom rows, initial state $\boldsymbol{x}_1$ is nearly deterministic and highly randomized, respectively. *Left:* Medians and interquartile ranges of simulated pole orientations. *Right:* Means and standard errors of success rates (estimated separately by unrolling the true system 100 times); dashed lines represent average performances of incumbent policies. On the bottom right, Pathwise (s) indicates that s samples were used during training.

context of typical learning problems, we experimented with both methods on the *cart-pole* task (Barto et al., 1983), which consists of moving a cart horizontally along a track in order to swing up and balance a pole, upside down, at a target location. State vectors $\boldsymbol{x} = [x_0, \dot{x}_0, x_1, \dot{x}_1]^\top$ define the position of the cart x_0, angle of the pole x_1, and time derivatives thereof; while, actions $\boldsymbol{a} \in \mathcal{A} = [-10, 10]$ N represent the lateral forces applied to the cart.

We follow Deisenroth et al. (2015) by using a 0.5 m long, 0.5 kg pole and a 0.5 kg cart with a 0.1 Ns/m friction coefficient. Each episode ran for a length of 3 seconds, discretized at 0.1 s intervals during which time actions were held constant, i.e., zero-order hold control. We set the goal state to $\boldsymbol{x}_{\text{goal}} = \mathbf{0}$ and define rewards according to a Gaussian function

$$r(\boldsymbol{x}) = \exp\left(-\frac{1}{2}\underbrace{(\boldsymbol{x} - \boldsymbol{x}_{\text{goal}})^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{x} - \boldsymbol{x}_{\text{goal}})}_{\substack{\text{sq. Euclidean distance} \\ \text{between pendulum tip and goal}}}\right), \tag{3.76}$$

whose precision matrix $\boldsymbol{\Lambda}^{-1}$ was chosen such that the bracket term is proportional to the squared Euclidean distance between (the Cartesian coordinates of) the tip of the pole in states $\boldsymbol{x}$ and $\boldsymbol{x}_{\text{goal}}$. Along the same lines, an episode was considered successful if the tip of the pole was within 0.1 m of the goal for 10 or more consecutive time steps. Depending on the particular experiment, states were initialized in one of two ways: (i) the standard case $\boldsymbol{x}_1 \sim \mathcal{N}\big([0, \pi, 0, 0]^\top, 0.01\mathbf{I}\big)$ or (ii) a challenge variant $\boldsymbol{x}_1 \sim \mathcal{N}\big([0, \pi, 0, 0]^\top, \text{diag}(1, 1, \pi, \pi)\big)$.

In all cases, system dynamics were represented by a set of independent sparse GPs

with squared exponential kernels, each of which predicted a single component of the tangent vector $\boldsymbol{f} = f(\boldsymbol{x}, \boldsymbol{a})$. Upon collecting an additional episode of training data, these GPs were trained from scratch using L-BFGS (Liu and Nocedal, 1989) with $m = \min(n, 256)$ inducing variables, whose corresponding inducing locations $\mathbf{Z}$ were initialized via k-means.

We defined policies as kernel regressors with inverse link functions $g^{-1} : \mathbb{R} \to [-10, 10]$

$$\pi_{\boldsymbol{\theta}}(\,\cdot\,) = g^{-1}\left(\sum_{i=1}^{30} w_i k(\,\cdot\,, \boldsymbol{x}_i)\right) \qquad g^{-1}(\,\cdot\,) = 20\Phi(\,\cdot\,) - 10, \tag{3.77}$$

where k denotes a squared exponential kernel and $\Phi : \mathbb{R} \to [0, 1]$ is the standard normal CDF. Policy parameters $\boldsymbol{\theta}$ consisted of centers $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{30})$, weights $\boldsymbol{w}$, and length scales $\boldsymbol{l}$. Following Deisenroth et al. (2015), policies were initialized once after collecting a random initial episode and subsequently fine-tuned. At each round, $\boldsymbol{\theta}$ was updated 5000 times using ADAM (Kingma and Ba, 2015) with gradient norms clipped to one and an initial learning rate 0.01 that decreased by a factor of ten after every third of training. Pathwise approaches propagated uncertainty by unrolling a separate draw of $\boldsymbol{x}_1$ along each realization of f, both of which were resampled prior to each update of $\boldsymbol{\theta}$.

In line with previous findings, moment-wise PILCO consistently solves the standard cart-pole task within a few episodes (Deisenroth et al., 2015). As initial state distributions become increasingly diffuse, however, moment matching struggles to accurately propagate uncertainty. As seen in the bottom row of Figure 3.9, this inability prevents moment-wise PILCO from learning meaningful policies for the challenge variant of cart-pole. Pathwise alternatives do not experience this issue, but they are not without their own shortcomings. We now discuss the relative merits of both approaches to propagating uncertainty.

Pathwise uncertainty propagation is significantly faster than moment matching, enabling us to simulate (tens of) thousands of trajectories in the time it takes to complete a single forward pass of moment matching. As Monte Carlo methods, pathwise estimates of (3.75) allow us to easily achieve the desired balance of accuracy and cost by controlling the sample size. Here, the use of sampling conveys additional benefits. First, it frees us from the restrictive class of moment matchable models by eliminating the need for closed-form integration. Second, it drastically simplifies implementation and allows us to fully take advantage of modern hardware and software, such as GPUs and automatic differentiation.

On the other hand, we observe that moment-wise uncertainty propagation sometimes improves performance. By locally linearizing the functions it permeates, moment matching implicitly favors simpler, smoother dynamics f and policies $\pi_{\boldsymbol{\theta}}$ (see Figure 3.9). Perhaps for this very reason, moment-wise PILCO was found to train more robustly. In particular, its pathwise counterpart was more susceptible to catastrophic forgetting: after solving the problem during the previous round of training, policies trained via pathwise uncertainty propagation were more likely to diverge. To illustrate this behavior, we define the *incumbent* as the policy that achieves highest expected reward under the model f. Unlike those of its moment-wise analogue,

pathwise PILCO's incumbents (dashed lines) often outperform more recent policies (solid lines) by significant margins; see right side of Figure 3.9. While this issue was easy to reproduce, the relative abundance of moving pieces makes it difficult to pinpoint precisely why it occurs.

Many of the challenges highlighted above are common in reinforcement learning, where generic solutions are often outperformed by skillfully tuned, bespoke alternatives. Nevertheless, we hope that the ease and flexibility of pathwise approaches to simulating posteriors will allow Gaussian processes to be applied to a wide range of problems where data-efficiency and uncertainty calibration are paramount.

3.6.5 Evaluating deep Gaussian processes

When applying Gaussian process methods to novel problems, we are often faced with a natural dilemma: many phenomena of interest are definitively non-Gaussian. In order to leverage Gaussian processes to model these phenomena, we typically resort to nonlinearly transforming f. Seeing as Gaussian random variables pushed forward through nonlinear functions seldom admit convenient analytic expressions, we are forced to trade tractability for expressivity.

This issue has recently come to the fore in the context of deep Gaussian processes (Damianou and Lawrence, 2013), which represent function priors as compositions

$$f(\,\cdot\,) = \Big(f^{(T)} \circ \ldots \circ f^{(2)} \circ f^{(1)}\Big)\big(\,\cdot\,\big), \tag{3.78}$$

where $f^{(t)} \sim \mathcal{GP}\big(\mu^{(t)}, k^{(t)}\big)$ for $t = 1, \ldots, T$. Following Salimbeni and Deisenroth (2017), sample-based methods have become the standard approach for evaluating and training these models. When a composition (3.78) consists of independent layers made up of independent, scalar-valued GPs (or linear combinations thereof), $f(\boldsymbol{x})$ can be efficiently sampled without resorting to expensive matrix operations. When these assumptions are violated, however, sample-based evaluations of deep GPs quickly becomes expensive. One such example was implicitly touched on in preceding sections: Gaussian process models of time-varying stochastic differential equations can be seen as continuous-time analogues of certain deep GPs (Hegde et al., 2019). In these cases, dependencies between successive evaluations of a GP-based drift function $f^{(t)}(\,\cdot\,) = f(t,\,\cdot\,)$ cause location-scale based evaluations to grind to halt (see Section 3.6.3).

Similar issues arise when sampling from compositions of multioutput GPs (van der Wilk et al., 2020). The remainder of this section focuses on the particular case of deep convolutional GPs (Blomqvist et al., 2019; Dutordoir et al., 2020). Here, a deep GP is defined in close analogy to a convolutional neural network (van der Wilk et al., 2017): each layer consists of a set of independent maps that are convolved over local subsets (patches) of an image $\boldsymbol{x}_t \in \mathbb{R}^{c_t \times h_t \times w_t}$. For a convolutional neural network, these *patch response functions* are affine transformations followed by nonlinearities; while, for a convolutional Gaussian process, they are draws from GP posteriors.

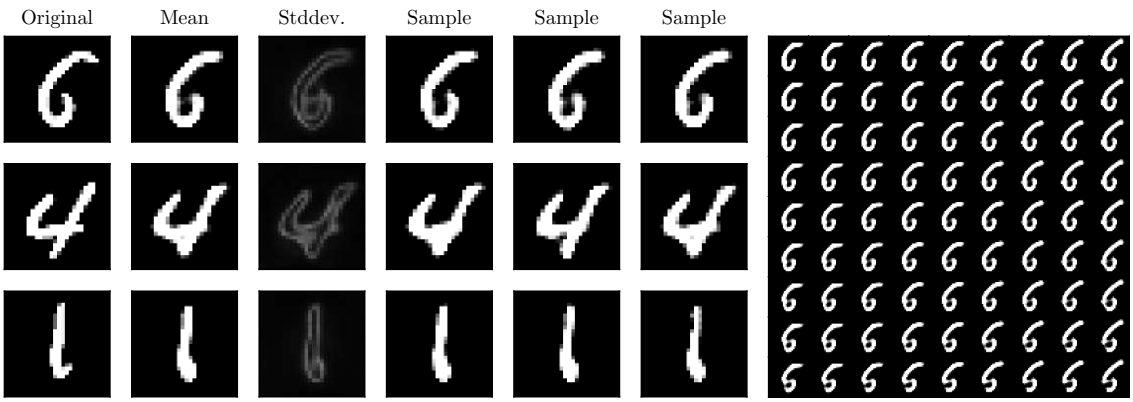

Figure 3.10: Reconstructions of MNIST digits by a deep convolutional GP trained to act as an autoencoder. *Left:* Mean and standard deviations of the (non-Gaussian) distribution over the reconstructions of randomly chosen test images are shown alongside three independently generated samples. *Right:* A 2-dimensional projection of a 25-dimensional latent space is found by performing SVD on the Jacobian of the mean response of the first decoder layer given an encoding of first image shown on the left. Reconstructions using the mean of each decoder layer are shown for a local walk in this 2-dimensional projected space.

Since each of the c_t independent patch response functions produces $h_t \times w_t$ output features, the covariance of the Gaussian random variables $\boldsymbol{x}_t = f^{(t-1)} * \boldsymbol{x}_{t-1}$ is a block diagonal square matrix of order $c_t \times h_t \times w_t$. Location-scale approaches to jointly sampling these feature maps incur $\mathcal{O}\left(c_t \times h_t^3 \times w_t^3\right)$ cost when computing matrix square roots.[17] Rather than sampling each layer at the current set of inputs, pathwise strategies sample entire models. Said again, pathwise approaches operate by drawing deterministic models from (approximations to) deep GP posteriors.[18] By doing so, these methods allow us to evaluate individual layers in $\mathcal{O}\left(c_t \times h_t \times w_t\right)$ time.

As an illustrative example, we trained a deep GP to act as an autoencoder for the MNIST dataset (LeCun and Cortes, 2010). For the encoder, we employed a sequence of three convolutional layers, each with 384 inducing patches $\mathbf{Z} \in \mathbb{R}^{c_{t-1} \times 3 \times 3}$ shared between $c_t \in (32, 32, 1)$ independent GPs. Strides and padding were chosen to produce a 25-dimensional encoding of a 784-dimensional image. Analogously, we defined the decoder using three transposed convolutional layers, each with 384 inducing patches $\mathbf{Z} \in \mathbb{R}^{c_{t-1} \times 3 \times 3}$ shared between $c_t \in (32, 32, 32)$ independent GPs. We then used a final decoder layer, consisting of a single convolutional GP (with the same general outline as above), to resolve penultimate feature maps $\mathbb{R}^{32 \times 28 \times 28}$ into image reconstructions $\mathbb{R}^{1 \times 28 \times 28}$. In all cases, we employ residual connections by using bilinear interpolation to define identity mean functions. Following Salimbeni and Deisenroth (2017), we initialized inducing patches $\mathbf{Z}$ using k-means and inducing distributions to be nearly deterministic.

Model evaluations were performed by using the sparse update (3.31) together with functions drawn from approximate priors constructed using $\ell = 256$ random Fourier

[17] This cost is separately incurred by each input to each layer, see Dutordoir et al. (2020).

[18] Here, we have assumed the use of approximate priors akin to those discussed in Section 3.3.

features. We associate each input image with a single draw of the model. Running on a single GPU, the model outlined above was jointly trained in just over 40 minutes using 10^4 steps of gradient descent with a batch size of 128. Figure 3.10 visualizes the behavior of reconstructions for a randomly chosen set of test images. While this GP-based autoencoder performs fairly well, there is an abundance of open questions regarding deep Gaussian processes in the wild. We hope that the ability to efficiently sample and evaluate draws of composite functions (3.78) will enable future works to further explore this space.

3.7 Discussion

Be it marginalizing out nuisance variables or evaluating expected utilities, integrals play a vital role in Bayesian algorithms. All too frequently, however, these integrals are intractable. Owing to their generality and ease of use, Monte Carlo estimators are often the weapons of choice in combating these impediments. Nevertheless, a Monte Carlo estimator is only as good as the samples it is based upon. And, while we have long been able to accurately sample Gaussian process posteriors, techniques that enable us to do so have rarely scaled well in the number of jointly distributed terms.

Through this chapter, we developed a pathwise interpretation of Gaussian process posteriors based on Matheron's update rule, Theorem 3.14. Adopting this viewpoint led to an interpretation of GP posteriors as the combination of a prior and a data-driven updated. The virtue of this approach is that it allows us to separately characterize the prior and the data in ways that are custom tailored to them. This added flexibility enables us to leverage existing methods for approximating GP priors without sacrificing our ability to faithfully represent the data.

These advantages are on full display when pathwise conditioning is used to power Monte Carlo algorithms. Here, the ability to efficiently draw functions from (approximate) posteriors allows us to simulate vectors $(f \mid \boldsymbol{y})(\mathbf{X}) \in \mathbb{R}^n$ in $\mathcal{O}(n)$ time. Further, the use of function draws means that locations $\mathbf{X}$ may be chosen ad hoc and typically provides access to pathwise derivatives, both of which are key properties exploited by algorithms discussed in the preceding text. We therefore argue for pathwise conditioning as a valuable addition to the metaphorical "toolkit".

CHAPTER 4

Conclusion

The goal of this thesis has been to advance the real-world application of Bayesian methods. All too often in Bayesian inference, the trouble is not so much figuring out *what* to compute but determining *how* to compute it. Hence, we have primarily focused on strategies for efficiently solving commonly occurring types of problems.

We began in Chapter 1 by reviewing the foundations of Bayesian decision theory. There, we saw how a binary relation and a handful of axioms led to a decision-making framework in which an agent's preference for a given action can be measured as the expectation of a corresponding utility function. This result paved the way for many of the algorithms discussed throughout this thesis.

This connection was stressed in Chapter 2, which introduced Bayesian optimization as the application of Bayesian decision theory to global optimization. Viewed from this perspective, Bayesian optimization is, conceptually, rather simple: (i) the agent will eventually use the available information to select a preferred solution, i.e. an incumbent; (ii) a model is used to simulate what updated information states might look like if an action is performed immediately; (iii) an optimal action is found by maximizing the expected utility of future incumbents under the model. While cogent, this story glosses over a number of key issues, such as how to obtain performant models or compute expected utilities.

Along these lines, the latter half of Chapter 2 revolved around techniques for maximizing acquisition functions. These techniques focus on two related problems. First, how should we optimize expected utilities when integration proves intractable? Section 2.4 attempted to help answer this question by investigating the use of pathwise gradient estimators. Second, how can we find efficiently (near-)optimal batches of queries $\mathbf{X} \in \mathcal{X}^q$? Section 2.5 showed that many popular batch acquisition

functions are submodular and used this fact to motivate greedy strategies. Together, these techniques give guidance on how to use Bayesian optimization in the real world.

Finally, in Chapter 3, we took a step back from Bayesian optimization to examine the general problem of scalable sampling of Gaussian process posteriors. This led us to a pathwise view of conditioning Gaussian random variables and processes. A key advantage of this framework is that it allows us to disentangle the way in which we represent the prior and the data. Translating theory into practice, we showed how these ideas open new doors for Monte Carlo methods by allowing us to efficiently sample functions from GP posteriors.

In closing, we argue that the combination of Gaussian process modeling and Bayesian decision-making is incredibly versatile and powerful. Gaussian processes allow us to encode our beliefs regarding the relatively likelihood of different events and Bayesian decision theory subsequently tells us *what* we should do. Adoption of this "meta-algorithm" is often only limited by a lack of knowledge for *how* to execute it in a limited amount of time. Our goal has been to help bridge this gap between what we would like to do and what we know how to do. This problem is by no means solved, but we hope to have done our small part to pave the way.

Bibliography

J. Azimi, A. Fern, and X. Fern. Batch Bayesian optimization via simulation matching. In *Advances in Neural Information Processing Systems*, 2010. Cited on pages 24, 35, 37.

F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3), 2013. Cited on page 35.

M. Balandat, B. Karrer, D. R. Jiang, S. Daulton, B. Letham, A. G. Wilson, and E. Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020. URL: `http://arxiv.org/abs/1910.06403`. Cited on pages vi, 25, 33, 43.

A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, (5):834–846, 1983. Cited on page 83.

J. Bect, D. Ginsbourger, L. Li, V. Picheny, and E. Vazquez. Sequential design of computer experiments for the estimation of a probability of failure. *Statistics and Computing*, 22(3):773–793, 2012. Cited on page 45.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 2012. Cited on page 31.

D. Blackwell. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*:105–110, 1947. Cited on page 39.

K. Blomqvist, S. Kaski, and M. Heinonen. Deep convolutional Gaussian processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 582–597. Springer, 2019. Cited on page 85.

V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. P. Deisenroth, and N. Durrande. Matérn Gaussian processes on graphs. In *Artificial Intelligence and Statistics*, pages 2593–2601, 2021. Cited on page 58.

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 58.

O. Bousquet and L. Bottou. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, 2008. Cited on page 32.

A. D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10), 2011. Cited on page 26.

D. R. Burt, C. E. Rasmussen, and M. van der Wilk. Convergence of sparse variational inference in Gaussian processes regression. *Journal of Machine Learning Research*, 21(131):1–63, 2020. Cited on page 65.

D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco. Gaussian process optimization with adaptive sketching: scalable and no regret. In *Conference on Learning Theory*, pages 533–557, 2019. Cited on page 61.

X. Cao. Convergence of parameter sensitivity estimates in a stochastic experiment. *IEEE Transactions on Automatic Control*, 30(9), 1985. Cited on pages 29, 30.

P. E. Castro, W. H. Lawton, and E. Sylvestre. Principal modes of variation for processes with continuous sample curves. *Technometrics*, 28(4):329–337, 1986. Cited on page 58.

J. T. Chang and D. Pollard. Conditioning as disintegration. *Statistica Neerlandica*, 51(3):287–317, 1997. Cited on page 49.

O. Chapelle and L. Li. An empirical evaluation of Thompson sampling. In *Advances in Neural Information Processing Systems*, pages 2249–2257, 2011. Cited on page 78.

Y. Chen and A. Krause. Near-optimal Batch Mode Active Learning and Adaptive Submodular Optimization. 2013. Cited on page 35.

C.-A. Cheng and B. Boots. Variational inference for Gaussian process models with linear complexity. In *Advances in Neural Information Processing Systems*, pages 5184–5194, 2017. Cited on page 60.

J.-P. Chilès and P. Delfiner. *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons, 2012. Cited on pages 46, 54.

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 54.

E. Contal, D. Buffoni, A. Robicquet, and N. Vayatis. Parallel Gaussian process optimization with upper confidence bound and pure exploration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2013. Cited on page 35.

M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pages 2292–2300, 2013. Cited on page 80.

A. Damianou and N. Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013. Cited on page 85.

B. de Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l'institut Henri Poincaré*, volume 7 of number 1, pages 1–68, 1937. Cited on pages 8, 10.

N. De Freitas, A. Smola, and M. Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. *arXiv preprint arXiv:1206.6457*, 2012. Cited on page 26.

C. de Fouquet. Reminders on the conditioning Kriging. In *Geostatistical Simulations*, pages 131–145. Springer, 1994. Cited on page 54.

M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015. Cited on pages 45, 82–84.

M. P. Deisenroth and C. E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning*, pages 465–472, 2011. Cited on page 82.

T. Desautels, A. Krause, and J. Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 2014. Cited on page 35.

L. Devroye. Nonuniform random variate generation. *Handbooks in operations research and management science*, 13:83–121, 2006. Cited on page 30.

C. R. Dietrich and G. N. Newsam. Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. *SIAM Journal of Scientific Computing*, 18:1088–1107, 1997. Cited on pages 56, 67.

A. Doucet. A note on efficient conditional simulation of Gaussian distributions. Technical report, University of British Columbia, 2010. Cited on page 54.

N. Durrande, V. Adam, L. Bordeaux, S. Eleftheriadis, and J. Hensman. Banded matrix operators for Gaussian Markov models in the automatic differentiation era. In *Artificial Intelligence and Statistics*, pages 2780–2789, 2019. Cited on page 56.

V. Dutordoir, M. van der Wilk, A. Artemev, and J. Hensman. Bayesian image classification with deep convolutional Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1529–1539, 2020. Cited on pages 85, 86.

X. Emery. Conditioning simulations of Gaussian random fields by ordinary Kriging. *Mathematical Geology*, 39(6):607–623, 2007. Cited on page 54.

L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010. Cited on page 59.

M. Figurnov, S. Mohamed, and A. Mnih. Implicit reparameterization gradients. *Advances in neural information processing systems*, 31, 2018. Cited on page 30.

P. C. Fishburn. The axioms of subjective probability. *Statistical Science*:335–345, 1986. Cited on page 8.

P. C. Fishburn. Utility theory for decision making. Technical report, Research analysis corp McLean VA, 1970. Cited on pages vi, 4, 11, 13.

R. FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445, 1961. Cited on page 81.

P. I. Frazier. A tutorial on Bayesian optimization. *arXiv:1807.02811*, 2018. Cited on page 77.

P. I. Frazier, W. B. Powell, and S. Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008. Cited on page 25.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Elsevier, 2013. Cited on page 58.

J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, and A. G. Wilson. GPyTorch: blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, pages 7576–7586, 2018. Cited on pages 65, 68.

M. Gelbart, J. Snoek, and R. Adams. Bayesian optimization with unknown constraints. *arXiv preprint arXiv:1403.5607*, 2014. Cited on page 26.

P. Glasserman. *Monte Carlo methods in financial engineering*. Springer, 2013. Cited on page 30.

P. Glasserman. Performance continuity and differentiability in Monte Carlo optimization. In *Simulation Conference Proceedings, 1988 Winter*. IEEE, 1988. Cited on pages 28–30.

A. Grigoryan. *Heat Kernel Analysis on Manifolds*. American Mathematical Society, 2009. Cited on page 59.

S. Grünewälder, J.-.-Y. Audibert, M. Opper, and J. Shawe–Taylor. Regret bounds for Gaussian process bandit problems. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 273–280. JMLR Workshop and Conference Proceedings, 2010. Cited on page 26.

S. S. Gupta and K. J. Miescke. Bayesian look ahead one-stage sampling allocations for selection of the best population. *Journal of statistical planning and inference*, 54(2):229–244, 1996. Cited on page 25.

N. Hansen. The CMA evolution strategy: a comparing review. *Towards a new evolutionary computation*:75–102, 2006. Cited on page 31.

P. Hegde, M. Heinonen, H. Lähdesmäki, and S. Kaski. Deep learning with differential Gaussian process flows. In *Artificial Intelligence and Statistics*, pages 1812–1821, 2019. Cited on page 85.

J. Hensman, N. Durrande, and A. Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18(151):1–151, 2017. Cited on page 60.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, pages 282–290, 2013. Cited on page 63.

J. Hensman, A. Matthews, and Z. Ghahramani. Scalable variational Gaussian process classification. In *Artificial Statistics and Machine Learning*, 2015. Cited on page 63.

J. M. Hernández-Lobato, J. Requeima, E. O. Pyzer-Knapp, and A. Aspuru-Guzik. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning*, pages 1470–1479, 2017. Cited on page 78.

I. N. Herstein and J. Milnor. An axiomatic approach to measurable utility. *Econometrica, Journal of the Econometric Society*:291–297, 1953. Cited on page 2.

Y. Hoffman and E. Ribak. Constrained realizations of Gaussian fields: a simple algorithm. *The Astrophysical Journal*, 380:L5–L8, 1991. Cited on page 54.

A. D. Ialongo, M. van der Wilk, J. Hensman, and C. E. Rasmussen. Overcoming mean-field approximations in recurrent Gaussian process models. In *International Conference on Machine Learning*, pages 2931–2940. PMLR, 2019. Cited on page 81.

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 23.

A. G. Journel and C. J. Huijbregts. *Mining Geostatistics*. Academic Press, 1978. Cited on pages 46, 54.

O. Kallenberg. *Foundations of Modern Probability*. Springer, 2006. Cited on pages 49, 50.

S. Kamthe and M. P. Deisenroth. Data-efficient reinforcement learning with probabilistic model predictive control. In *Artificial Intelligence and Statistics*, pages 1701–1710, 2018. Cited on page 82.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv:1807.02582*, 2018. Cited on page 76.

K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos. Parallelised Bayesian optimisation via Thompson sampling. In *Artificial Intelligence and Statistics*, pages 133–142, 2018. Cited on page 78.

T. Kathuria, A. Deshpande, and P. Kohli. Batched Gaussian process bandit optimization via determinantal point processes. In *Advances in Neural Information Processing Systems*, 2016. Cited on page 35.

D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014. Cited on page 30.

D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015. Cited on pages 31, 33, 84.

C. H. Kraft, J. W. Pratt, and A. Seidenberg. Intuitive probability on finite sets. *The Annals of Mathematical Statistics*, 30(2):408–419, 1959. Cited on page 8.

E. T. 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 58, 59.

A. Krause and D. Golovin. Submodular function maximization, 2014. Cited on page 35.

D. Kreps. *Notes On The Theory Of Choice*. Westview Press, 1988. Cited on pages 4, 7, 16.

M. Lázaro-Gredilla and A. Figueiras-Vidal. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, pages 1087–1095, 2009. Cited on pages 60, 64.

Y. LeCun and C. Cortes. MNIST handwritten digit database, 2010. URL: `http://yann.lecun.com/exdb/mnist/`. Cited on page 86.

M. Lifshits. *Lectures on Gaussian Processes*. Springer, 2012. Cited on page 59.

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 58, 59.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989. Cited on page 84.

J. Loper, D. Blei, J. P. Cunningham, and L. Paninski. General linear-time inference for Gaussian Processes on one dimension. *arXiv:2003.05554*, 2020. Cited on page 56.

G. J. Lord, C. E. Powell, and T. Shardlow. *An Introduction to Computational Stochastic PDEs*. Cambridge University Press, 2014. Cited on pages 58, 59.

D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, 1997. Cited on page 50.

A. Mallasto and A. Feragen. Learning from uncertain curves: the 2-Wasserstein metric for Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 5660–5670, 2017. Cited on page 69.

R. Martinez-Cantin. Bayesopt: A Bayesian optimization library for nonlinear optimization, experimental design and bandits. *Journal of Machine Learning Research*, 15(1), 2014. Cited on page 26.

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 77.

J. Močkus. On Bayesian methods for seeking the extremum. In *Optimization techniques IFIP Technical Conference*, pages 400–404. Springer, 1975. Cited on page 77.

S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih. Monte Carlo Gradient Estimation in Machine Learning. *J. Mach. Learn. Res.*, 21(132):1–62, 2020. Cited on pages 28, 30.

M. Mutny and A. Krause. Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features. In *Advances in Neural Information Processing Systems*, pages 9005–9016, 2018. Cited on page 61.

J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the Institute of Radio Engineers*, 50(10):2061–2070, 1962. Cited on page 81.

G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1), 1978. Cited on page 35.

H. Nickisch and C. E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008. Cited on page 63.

D. S. Oliver. On conditional simulation to inaccurate data. *Mathematical Geology*, 28(6):811–817, 1996. Cited on page 54.

M. Opper and C. Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009. Cited on page 64.

A. Parker and C. Fox. Sampling Gaussian distributions in Krylov spaces with conjugate gradients. *SIAM Journal on Scientific Computing*, 34(3):B312–B334, 2012. Cited on page 65.

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*, pages 4114–4123, 2018. Cited on pages 56, 65, 67.

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*, pages 22268–22281, 2020. Cited on page 65.

J. Quiñonero-Candela, C. E. Rasmussen, and C. K. I. Williams. Approximation methods for Gaussian process regression. In *Large-scale Kernel Machines*, pages 203–223. MIT Press, 2007. Cited on pages 63, 66, 76.

A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2008. Cited on page 57.

F. P. Ramsey. Truth and Probability. In R. B. Braithwaite, editor, *The Foundations of Mathematics and other Logical Essays*, chapter 7, pages 156–198. McMaster University Archive for the History of Economic Thought, 1926. URL: `https://EconPapers.repec.org/RePEc:hay:hetcha:ramsey1926`. Cited on page 10.

C. R. Rao. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 37(3):81–91, 1945. Cited on page 39.

C. E. Rasmussen and M. Kuss. Gaussian processes in reinforcement learning. In *Advances in Neural Information Processing Systems*, 2004. Cited on page 82.

C. E. Rasmussen and J. Quiñonero-Candela. Healing the relevance vector machine through augmentation. In *International Conference on Machine Learning*, pages 689–696, 2005. Cited on page 76.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. Cited on pages 45, 55, 59, 63, 66.

D. Rezende, M. Shakir, and D. Wierstra. Stochastic Backpropagation and Variational Inference in Deep Latent Gaussian Models. In *International Conference on Machine Learning*, 2014. Cited on page 30.

H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications*. CRC Press, 2005. Cited on pages 56, 79.

H. Salimbeni, C.-A. Cheng, B. Boots, and M. P. Deisenroth. Orthogonally decoupled variational Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 8711–8720, 2018. Cited on page 60.

H. Salimbeni and M. P. Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, 2017. Cited on pages 85, 86.

L. J. Savage. *The Foundations of Statistics*. Wiley Publications in Statistics, 1954. Cited on page 9.

B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001. Cited on page 57.

M. Seeger. Bayesian methods for support vector machines and Gaussian processes. Technical report, 1999. Cited on page 64.

A. Shah and Z. Ghahramani. Parallel predictive entropy search for batch global optimization of expensive objective functions. In *Advances in Neural Information Processing Systems*, 2015. Cited on page 35.

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 45.

J. Shi, M. K. Titsias, and A. Mnih. Sparse orthogonal variational inference for Gaussian processes. In *Artificial Intelligence and Statistics*, pages 1932–1942, 2020. Cited on page 60.

B. W. Silverman. Spline smoothing: the equivalent variable kernel method. *The Annals of Statistics*:898–916, 1984. Cited on pages 67, 70.

B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society: Series B (Methodological)*, 47(1):1–21, 1985. Cited on page 66.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2006. Cited on page 63.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012. Cited on page 78.

A. Solin and M. Kok. Know your boundaries: Constraining Gaussian processes by variational harmonic features. In *Artificial Intelligence and Statistics*, pages 2193–2202, 2019. Cited on pages 58, 79.

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 58.

P. Sollich and C. Williams. Using the equivalent kernel to understand Gaussian process regression. In *Advances in Neural Information Processing Systems*, pages 1313–1320, 2005. Cited on page 70.

N. Srinivas, A. Krause, S. 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 pages 26, 35.

N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Inernational Conference on Machine Learning*, pages 1015–1022, 2010. Cited on page 77.

D. Sutherland and J. Schneider. On the error of random Fourier features. In *Uncertainty in Artificial Intelligence*, pages 862–871, 2015. Cited on pages 57, 71, 72.

L. Tallorin, J. Wang, W. E. Kim, S. Sahu, N. M. Kosa, P. Yang, M. Thompson, M. K. Gilson, P. I. Frazier, M. D. Burkart, et al. Discovering de novo peptide substrates for enzymes using machine learning. *Nature communications*, 9(1):5253, 2018. Cited on page 37.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. Cited on page 78.

M. E. Tipping. The relevance vector machine. In *Advances in Neural Information Processing Systems*, pages 652–658, 2000. Cited on page 66.

M. Titsias and N. D. Lawrence. Bayesian Gaussian process latent variable model. In *Artificial Intelligence and Statistics*, pages 844–851, 2010. Cited on page 63.

M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009. Cited on page 63.

M. K. Titsias. Variational model selection for sparse Gaussian process regression. Technical report, University of Manchester, 2009. Cited on page 63.

M. van der Wilk, V. Dutordoir, S. T. John, A. Artemev, V. Adam, and J. Hensman. A framework for interdomain and multioutput Gaussian processes. *arXiv:2003.01115*, 2020. Cited on page 85.

M. van der Wilk, C. E. Rasmussen, and J. Hensman. Convolutional Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 2849–2858, 2017. Cited on page 85.

E. Vazquez and J. Bect. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and inference*, 140(11):3088–3095, 2010. Cited on page 26.

C. Villani. *Optimal Transport: Old and New*. Springer, 2008. Cited on page 69.

C. Villegas. On Qualitative Probability /*sigma*-Algebras. *The Annals of Mathematical Statistics*, 35(4):1787–1796, 1964. Cited on page 8.

J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944. Cited on page 2.

G. Wahba. *Spline models for observational data*. Society for Industrial and Applied Mathematics, 1990. Cited on page 66.

J. Wang, S. Clark, E. Liu, and P. Frazier. Parallel Bayesian global optimization of expensive functions. *arXiv preprint arXiv:1602.05149*, 2016. Cited on pages vi, 26, 30, 38, 43.

K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, and A. G. Wilson. Exact Gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, pages 14622–14632, 2019. Cited on page 65.

Z. Wang, C. Gehring, P. Kohli, and S. Jegelka. Batched large-scale Bayesian optimization in high-dimensional spaces. In *Artificial Intelligence and Statistics*, pages 745–754, 2018. Cited on page 61.

P. Whittle. Stochastic processes in several dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994, 1963. Cited on page 79.

A. Wilson and H. Nickisch. Kernel interpolation for scalable structured Gaussian processes. In *International Conference on Machine Learning*, pages 1775–1784, 2015. Cited on pages 56, 67.

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 vi.

J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowski, and M. P. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, pages 7470–7480, 2020. Cited on pages vi, 70, 73, 76, 79.

J. T. Wilson, F. Hutter, and M. P. Deisenroth. Maximizing acquisition functions for Bayesian optimization. In *Advances in Neural Information Processing Systems*, pages 9884–9895, 2018. Cited on pages vi, 26, 78.

A. T. Wood and G. Chan. Simulation of stationary Gaussian processes in $[0, 1]^d$. *Journal of Computational and Graphical Statistics*, 3(4):409–432, 1994. Cited on pages 56, 67.

J. Wu and P. Frazier. The parallel Knowledge Gradient method for batch Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2016. Cited on pages vi, 25, 31.

J. Wu, M. Poloczek, A. Wilson, and P. Frazier. Bayesian optimization with gradients. In *Advances in Neural Information Processing Systems*, pages 5267–5278, 2017. Cited on pages 25, 31.

H. Zhu, C. K. Williams, R. Rohwer, and M. Morciniec. Gaussian regression and optimal finite dimensional linear models. Technical report, Aston University, 1997. Cited on page 58.

D. L. Zimmerman. Computationally exploitable structure of covariance matrices and generalized convariance matrices in spatial models. *Journal of Statistical Computation and Simulation*, 32(1-2):1–15, 1989. Cited on page 56.