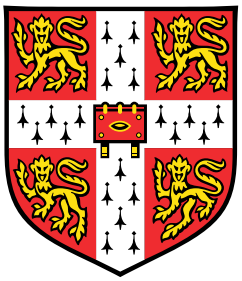
Scalable Approximate Inference and Model Selection in Gaussian Process Regression



David R. Burt

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Emmanuel College

April 2022

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

David R. Burt
April 2022

Abstract

Models with Gaussian process priors and Gaussian likelihoods are one of only a handful of Bayesian models where inference can be performed without the need for approximation. However, a frequent criticism of these models from practitioners of Bayesian machine learning is that they are challenging to scale to large datasets due to the need to compute a large kernel matrix and perform standard linear-algebraic operations with this matrix. This limitation has driven decades of research in both statistics and machine learning seeking to scale Gaussian process regression models to ever-larger datasets. This thesis builds on this line of research. We focus on the problem of approximate inference and model selection with approximate maximum marginal likelihood as applied to Gaussian process regression. Our discussion is guided by three questions: *Does an approximation work on a range of models and datasets? Can you verify that an approximation has worked on a given dataset? Is an approximation easy for a practitioner to use?* While we are far from the first to ask these questions, we offer new insights into each question in the context of Gaussian process regression.

In the first part of this thesis, we focus on sparse variational Gaussian process regression (Titsias, 2009). We provide new diagnostics for inference with this method that can be used as practical guides for practitioners trying to balance computation and accuracy with this approximation. We then provide an asymptotic analysis that highlights properties of the model and dataset that are sufficient for this approximation to perform reliable inference with a small computational cost. This analysis builds on an approach laid out in Burt (2018), as well as on similar guarantees in the kernel ridge regression literature.

In the second part of this thesis, we consider iterative methods, especially the method of conjugate gradients, as applied to Gaussian process regression (Gibbs and MacKay, 1997). We primarily focus on improving the reliability of approximate maximum marginal likelihood when using these approximations. We investigate how the method of conjugate gradients and related approaches can be used to derive bounds on quantities related to the log marginal likelihood. This idea can be used to improve the speed and stability of model selection with these approaches, making them easier to use in practice.

Acknowledgements

I would like to thank my supervisor, Carl Edward Rasmussen for his advice over the past four years. I have learned a great deal both from Carl's technical perspective, and perhaps more importantly, his perspective on the value of careful research and inquiry. I would also like to thank my advisor Richard E. Turner. Rich always seems to have the right question to get to the heart of a problem, and has been generous with his time when I had a question or needed help. I would also like to thank Mark van der Wilk, who has been an excellent mentor and collaborator. Many parts of this thesis build on discussions with Mark, and working with and learning from Mark has been a joy. Thanks also to my examiners Neil Lawrence and Aki Vehtari. I greatly appreciate the feedback they provided, as well as the opportunity to discuss this thesis with them and hear their insights.

I would also like to thank my other collaborators. I would particularly like to single out Andrew Y.K. Foong, Sebastian W. Ober, Artem Artemev, Wessel P. Bruinsma, Adrià Garriga-Alonso and David Janz. As well as many useful discussions when collaborating, they have provided feedback on writing and help with proofreading numerous times, often on work that they were not directly involved with. Their insightful comments, as well as good-natured teasing about typos, undoubtedly improved my writing over the past four years. Thanks also to the entire Cambridge Machine Learning group; I have enjoyed many lunch-time and tea-time discussions, and learned a great deal from them.

Finally, on a personal level I would like to thank my friends and family for all of their support not only over the past four years, but for the process to getting here. Particularly, thanks to Victoria Hughes and Jim Goodwin, who have supported my curiosity for as long as I can remember. To Anna Caliandro, for her patience, kindness, sense of humor and emotional support over the past five years. And last, but of course not least to my brother Craig and my parents Lynda and Nathan. If I thanked you three for all of the things I should, I would reach the word limit before I got into the content of the thesis, so I will leave it at thanks and I love you.

Relationship to Previous Work and Publications

Chapters 1 and 2 The first two chapters provide background material. In Chapter 2, some diagnostics for approximate inference in Gaussian process regression are discussed that have not appeared previously in the literature and were derived in the writing of this thesis.

Chapter 3 Chapter 3 gives an a priori analysis of sparse variational Gaussian process regression. The results in this chapter appeared in the conference paper Burt et al. (2019) and the journal paper Burt et al. (2020b), both of which were co-authored with Carl Edward Rasmussen and Mark van der Wilk.

The initial idea and the overall approach which we take in section 3.1 was outlined in Burt (2018), a dissertation submitted in partial fulfillment of my Master of Philosophy degree at the University of Cambridge. In the introduction to the chapter we discuss specifically which results were contained in Burt (2018). A summary of this is:

- The general proof structure in section 3.1 was outlined in Burt (2018). However, the structure has been clarified leading to improved exposition. Additionally, the methods for defining inducing points in Burt (2018) were impractical in general. We adapt the analysis to consider practical methods which retain similar guarantees. Further, an analysis under the assumption that the model is correctly specified that leads to sharper bounds than in Burt (2018) was added in Burt et al. (2019) and is contained here.
- In section 3.2 we consider a much broader range of assumptions on the kernel and data-generating process than considered in Burt (2018), which only considered the squared exponential kernel and Matérn $1/2$ kernel in one-dimension.

The lower bounds in section 3.3 were not considered in Burt (2018), and the discussion of related literature is far more comprehensive.

Chapter 4 Chapter 4 discusses the application of the method of conjugate gradients and other iterative approaches to approximate Gaussian process regression. The first two sections of Chapter 4 are review, and we provide citations to relevant papers and textbooks as appropriate. The later sections present and build on ideas from Artemev et al. (2021) and Burt et al. (2021). The experimental results presented in these chapters are from these papers, which were co-authored with Artem Artemev and Mark van der Wilk. Artem and I contributed equally to both of these works. The experiments were conducted by Artem Artemev.

Chapter 5 Chapter 5 contains concluding remarks written for this thesis.

Notation

Table 1 **Matrices**

Spaces:	
$\mathbb{R}^{N\times M}$	$N \times M$ matrices with real entries
S^N (symmetric matrix)	$\{\mathbf{A} \in \mathbb{R}^{N\times N} : \mathbf{A} = \mathbf{A}^\top\}$
S^N_+ (positive semi-definite matrix)	$\{\mathbf{A} \in S^N : \forall \mathbf{y} \in \mathbb{R}^N, \mathbf{y}^\top \mathbf{A}\mathbf{y} \geq 0\}$
S^N_{++} (positive definite matrix)	$\{\mathbf{A} \in S^N : \forall \mathbf{y} \in \mathbb{R}^N, \mathbf{y}^\top \mathbf{A}\mathbf{y} > 0\}$
Special Matrices:	
$\mathbf{I}$ (identity matrix)	$\mathbf{I}\mathbf{v} = \mathbf{v}$ for all $\mathbf{v} \in \mathbb{R}^N$
Basic Operations:	
$\cdot^\top$ (transpose)	$[\mathbf{A}^\top]_{ji} = [\mathbf{A}]_{ij}$
$\cdot^{-1}$ (inverse)	$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
$\lambda_i(\cdot)$ (i^{th} largest eigenvalue counted with multiplicity)	see definition A.3
$\sigma_i(\cdot)$ (i^{th} largest singular value)	$\sigma_i(\mathbf{A}) = \sqrt{\lambda_i(\mathbf{A}^\top \mathbf{A})}$
$\operatorname{tr}(\cdot)$ (trace)	$\operatorname{tr}(\mathbf{A}) = \sum_{n=1}^N [\mathbf{A}]_{nn}$
$\det(\cdot)$ (determinant)	see definition A.11
Norms:	
$\|\cdot\|, \|\cdot\|_{\text{op}}$ (Euclidean operator norm)	$\sup_{\mathbf{v}\neq\mathbf{0}} \frac{\|\mathbf{A}\mathbf{v}\|_2}{\|\mathbf{v}\|_2} (= \sigma_1(\mathbf{A}))$
$\|\cdot\|_{\text{Sc},p}$ (Schatten p-norm)	$(\sum_{n=1}^N \lambda_n(\mathbf{A})^p)^{1/p}$
Order Relations:	
$\mathbf{A} \succ \mathbf{B}$ (Loewner order, $\mathbf{A}, \mathbf{B} \in S^N_+$)	$\mathbf{A} - \mathbf{B} \in S^N_+$
$\mathbf{A} \prec \mathbf{B}$ ($\mathbf{A}, \mathbf{B} \in S^N$)	$\mathbf{B} - \mathbf{A} \in S^N_+$.

Table 2 **Probabilities and Modeling**

Probabilistic Notation:	
$x \sim P$	x is distributed according to probability measure P.
$x_i \overset{\text{i.i.d.}}{\sim} P$	$x_1, \ldots x_I$ are independent and identically distributed each according to probability measure P
$\mathcal{N}(a,b)$	Gaussian measure with mean a and variance b
$\mathcal{N}(\mathbf{\mu}, \mathbf{\Sigma})$	Gaussian measure on $\mathbb{R}^D$ with mean vector $\mathbf{\mu}$ and covariance matrix $\mathbf{\Sigma}$
$\mathcal{N}(x;a,b)$	Density of Gaussian measure with mean a and variance b evaluated at x.
$\mathcal{N}(\mathbf{x};\mathbf{\mu},\mathbf{\Sigma})$	Density of Gaussian measure on $\mathbb{R}^D$ with mean vector $\mathbf{\mu}$ and covariance matrix $\mathbf{\Sigma}$ evaluated at $\mathbf{x} \in \mathbb{R}^D$
$\mathcal{GP}(\mu,k)$	Gaussian process with mean function μ and covariance function k.
$\mathfrak{D}_{\text{KL}}(Q,P)$	Kullback-Leibler Divergence.
$\mathfrak{D}_{\text{TV}}(Q,P)$	Total variation distance.
Modeling Notation:	
$\mathcal{X}$	Space containing inputs, commonly $\mathbb{R}^D$
N	Number of observations.
$\mathbf{x} \in \mathcal{X}^N$	Inputs or covariates.
$\mathbf{y} \in \mathbb{R}^N$	Outputs or response variables.
$\mathcal{D}$	Dataset, $\mathcal{D} = (x_n, y_n)_{n=1}^N$
θ	Hyperparameters of the model.
$\ell_{\mathcal{D}} : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}$	The likelihood of a regressor given dataset $\mathcal{D}$.
$\mathcal{L}(\theta)$	Log marginal likelihood of θ.
P	Prior measure on a latent function.
$P_{\mid \mathcal{D}}$	Posterior measure on a latent function.
Sparse Variational Inference Notation:	
$\mathbf{z}$	Inducing points.
M	Number of Inducing Points.
Q	Variational approximation to the posterior measure.
$\underline{\mathcal{L}}(Q,\theta)$	Evidence lower bound of sparse variational Gaussian process with approximate posterior Q and hyperparameters θ.
$\underline{\mathcal{L}}(\mathbf{z},\theta)$	Collapsed evidence lower bound of sparse variational Gaussian process with inducing points $\mathbf{z}$, hyperparameters θ and the optimal choice of variational parameters $(\mathbf{\mu},\mathbf{\Sigma})$.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The topic of this thesis is the analysis and development of methods for approximate inference and maximum marginal likelihood model selection in Gaussian process regression models. Before discussing approximate versions of Gaussian process regression, there are several questions we must address, which will be the topic of the first three sections of this chapter:

- How do I perform exact inference in Gaussian process regression (section 1.1)?
- What choices do I need to make when applying Gaussian process regression (section 1.2)?
- Why would I use an approximate version of a Gaussian process regression workflow (section 1.3)?

Readers who are well-versed in modeling with Gaussian process regression and model selection with maximum likelihood may safely skim these sections to familiarize themselves with the notation used in the remainder of this thesis.

The fourth and fifth sections in this chapter discuss approximate Gaussian process regression and address the following questions:

- What properties should an approximation to Gaussian process regression satisfy (section 1.4)?
- What are some paradigms for approximating Gaussian process regression (section 1.5)?

We conclude the chapter with a summary of the remainder of this thesis, highlighting the contributions in subsequent chapters (section 1.6).

The technical depth at which we describe topics will depend partially on the extent to which we rely on the topic in this thesis. For example, we spend more time discussing properties of kernels, particularly from the viewpoint of Mercer's theorem, than is typical in an introduction to Gaussian process regression. This emphasis is because Mercer's theorem will play an important role in later results on the quality of approximations to Gaussian process regression that are a primary contribution of chapter 3.

1.1 Gaussian Process Modeling

We consider a supervised learning problem with real-valued response variables: we have observed a dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$, with $x_n \in \mathcal{X}$ and $y_n \in \mathbb{R}$. $\mathcal{X}$ can be any non-empty set; often we have $\mathcal{X} \subset \mathbb{R}^D$.

We will interchangeably use the terms *inputs* and *covariates* to refer to the x_n and the terms *outputs* and *response variables* to refer to the y_n. Given some new values of covariates, $x_{n+1}, \ldots, x_{n+k}$ we want to make predictions about the corresponding response variable $y_{n+1}, \ldots, y_{n+k}$.

In order to extract information from the data, we need to make some assumptions about the relationship between the covariates and response variables. We posit a relationship of the form

$$y_n = f(x_n) + \varepsilon_n, \tag{1.1}$$

where f is an (unknown) function, $\varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$ and $\sigma^2 > 0$ determines the magnitude of the variance in the response variables that is not explained by the latent function, f. We refer to σ^2 as the *likelihood variance* or the *noise variance*, and for the moment assume it is known. In words, the response variables are obtained by applying a function to the covariates and then adding homoscedastic, Gaussian noise. Figure 1.1 illustrates such a process, with the orange dots representing the input output variable pairs (x_n, y_n), the blue curve representing the graph of the latent function, f, and the dashed black lines between the blue curve and orange dots the additive noise, ε_n. The central question of inference is to determine (a distribution over) f given the observed dataset, $\mathcal{D}$, which can then be used to make predictions about the outputs at previously unseen input values.

For many tasks simple parametric forms of f, such as linear functions of features defined using x, are appropriate. However, in cases where we want the model to be more flexible, a non-parametric approach is appealing. We assume a Gaussian process prior is taken over f, meaning that for any finite collection of possible covariates, $x'_1, \ldots, x'_s$, we assume (a priori) that $f(x'_1), \ldots, f(x'_s)$ is a jointly Gaussian random variable. By the Gaussian assumption, the prior over f can be fully-specified by defining a *mean function* $\mu \colon \mathcal{X} \to \mathbb{R}$ and a *covariance function, kernel function* or simply *kernel* $k \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where k is positive semi-definite, meaning that for any $I \in \mathbb{N}$, $(x_1, \ldots, x_I) \in \mathcal{X}^I$ and $\mathbf{v} \in \mathbb{R}^I$,

$$\sum_{1 \leq i,j \leq I} \mathbf{v_i} \mathbf{v_j} k(x_i, x_j) \geq 0. \tag{1.2}$$

Putting together these assumptions, the model is

$$\underbrace{y_n = f(x_n) + \varepsilon_n, \quad \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)}_{\text{likelihood}}, \quad \underbrace{f \sim \mathcal{GP}(\mu, k)}_{\text{prior}}. \tag{1.3}$$

For notational simplicity, we assume throughout this thesis that $\mu \equiv 0$, although all the results discussed can be extended to the case of a known, non-zero mean function with minor modifications.

1.1.1 Inference in Gaussian Process Regression

Having specified the model (eq. 1.3), Bayes' rule tells us the form of the conditional distribution $f|\mathcal{D}$. In particular, let P denote the distribution of f, and $P_{|\mathcal{D}}$ the distribution of $f|\mathcal{D}$. Let p and $p_{|\mathcal{D}}$ denote the

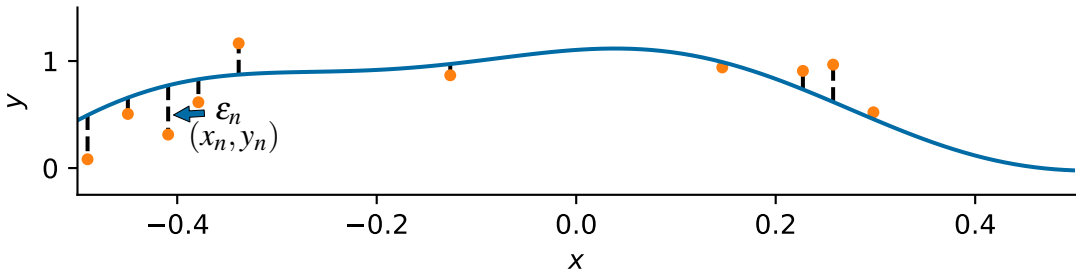


Fig. 1.1 The data generating process (eq. 1.3): A latent function, f, (blue) is drawn from a Gaussian process prior. Datapoints, (x_n, y_n), (orange) are assumed to be noisy observations of the latent function with additive noise, ε_n, (dashed black lines).

densities of the prior and posterior.[1] Bayes' rule states that

$$\frac{p_{|\mathcal{D}}(f)}{p(f)} = \frac{1}{\mathcal{L}}\ell_{\mathcal{D}}(f), \tag{1.4}$$

where $\ell_{\mathcal{D}}(f)$ is the *likelihood* of the latent function f, and the *marginal likelihood* of the model $\mathcal{L} := \int \ell_{\mathcal{D}} dP$ is such that the posterior is a probability measure. The likelihood function is a function of f, defined as the probability density of the observed data conditioned on the value of f at which it is evaluated.[2] For the model in eq. (1.3)

$$\ell_{\mathcal{D}}(f) = \prod_{n=1}^{N} \mathcal{N}(y_n; f(x_n), \sigma^2), \tag{1.5}$$

where we use $\mathcal{N}(a; b, c)$ to denote the density function of a normal random variable with mean b and variance c evaluated at a.

A significant benefit of regression models with Gaussian process priors and Gaussian likelihoods is that they admit closed-form Bayesian inference using only standard linear-algebraic operations. Inference in the model defined in eq. (1.3) on the dataset from figure 1.1 is shown in figure 1.2.

Before stating the central equations of inference in Gaussian process regression, we introduce several bits of notation that will be used throughout the paper. We define $\mathbf{x} \in \mathcal{X}^N$ to have n^{th} coordinate x_n and $\mathbf{y} \in \mathbb{R}^N$ to have n^{th} coordinate y_n for $1 \leq n \leq N$. For any natural numbers I, J and any tuples $\mathbf{z} \in \mathcal{X}^I$ and $\mathbf{z}' \in \mathcal{X}^J$ we define the matrix $\mathbf{K}_{\mathbf{z},\mathbf{z}'} \in \mathbb{R}^{I \times J}$ by $(\mathbf{K}_{\mathbf{z},\mathbf{z}'})_{ij} = k(z_i, z'_j)$ for $1 \leq i \leq I, 1 \leq j \leq J$. In the case when $I = 1$ or $J = 1$, we slightly abuse notation and identify single element sets with the elements they contain, and write $\mathbf{k}_{z\mathbf{z}'} \in \mathbb{R}^{1 \times J}$. Further, we define the matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ by $\mathbf{K} = \mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 \mathbf{I}_N$, where

[1] Generally, densities will be respect to Lebesgue measure. In cases where this does not make sense such as in eq. (1.4), they are with respect to an arbitrary suitable base measure.

[2] This is commonly written as $\ell_{\mathcal{D}}(f) = p(\mathbf{y}|f)$.

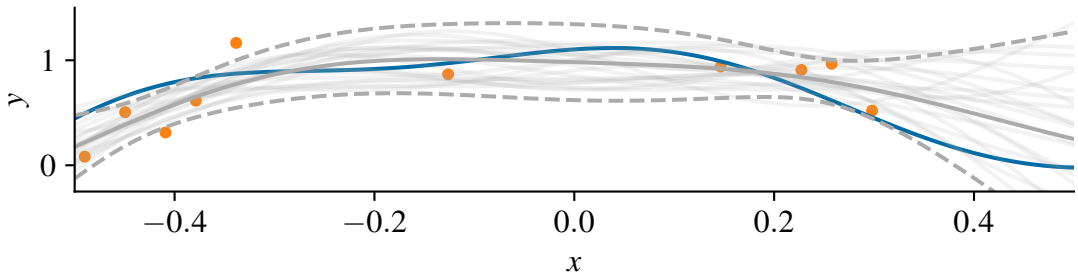


Fig. 1.2 An illustration of the posterior distribution of the model eq. (1.3), fit on data (orange points). The data is generated as in figure 1.1. Light gray curves are samples from $f|\mathcal{D}$, a posterior Gaussian process fit on the data. The solid gray line is the posterior mean, and the area between the dashed gray lines represents a 95% credible interval for the latent function $f|\mathcal{D}$.

$\mathbf{I}_N \in \mathbb{R}^{N\times N}$ is the identity matrix. When no confusion can arise regarding dimension, we simply write $\mathbf{I}$ for the identity matrix.

The posterior distribution of f in the model eq. (1.3), conditioned on a dataset $\mathcal{D}$, takes the form (see Rasmussen and Williams 2006, Chapter 3)

$$f|\mathcal{D} \sim \mathcal{GP}(\hat{\mu}, \hat{k}), \quad \hat{\mu}(x) = \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{y}, \quad \hat{k}(x,x') = k(x,x') - \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x}x'}. \tag{1.6}$$

The details of the calculation involved to sample from and evaluate moments of arbitrary finite-dimensional marginal distributions of eq. (1.6) are outlined in section 1.3 and can be implemented numerically using standard linear algebra libraries.

1.2 Modeling Choices

In order to specify the model (eq. 1.3), the practitioner must select the likelihood variance σ^2, a mean function μ, and a kernel function k. We now review several standard choices of kernel functions that we will use as running examples throughout this thesis. More examples and properties of kernel functions, within the context of Gaussian process modeling, can be found in Rasmussen and Williams (2006, Chapter 4) and Duvenaud (2014, Chapter 2). The latter part of this section focuses on Mercer's and Bochner's theorems, both of which give alternative representations of kernel functions that can offer insight into the assumptions implicitly encoded by the kernel.

1.2.1 Common Kernel Functions

For any kernel k and $\sigma_k^2 \geq 0$, a new kernel can be formed by $k(x,x') \to \sigma_k^2 k(x,x')$. We refer to σ_k^2 as the *kernel variance parameter*. When $\mathcal{X} = \mathbb{R}^D$ for any *isotropic kernel* (i.e. a kernel that is a function only of $\|x-x'\|$) or any *inner product kernel* (i.e. a kernel that is a function only of $x^\top x'$) we can introduce

a positive definite *lengthscale matrix*, $\mathbf{L} \in S_{++}^N$, that scales and possibly rotates the normal Euclidean inner product and notion of distance. In particular, the lengthscale matrix induces a new kernel via $k(x,x') \to k(\mathbf{L}^{-1/2}x, \mathbf{L}^{-1/2}x)$. Frequently, $\mathbf{L}$ is chosen to be diagonal, in which case $\sqrt{\mathbf{L}_{dd}}$, $1 \leq d \leq D$ is referred to as the *lengthscale for input dimension* d. When giving examples of kernels, we omit σ_k^2 and $\mathbf{L}$ for brevity.

We state several well-known examples of kernels. In all cases described in this section, it is assumed that $\mathcal{X} \subset \mathbb{R}^D$, although Gaussian process regression models can be defined on many other interesting domains, see for example Borovitskiy et al. (2020, 2021).

Linear Kernel

The simplest interesting kernel is the *linear kernel* defined by

$$k(x,x') = x^\top x'. \tag{1.7}$$

For this kernel and $\mu \equiv 0$, the model eq. (1.3) is equivalent to the linear-Gaussian model

$$y_n = \boldsymbol{\alpha}^\top x_n + \varepsilon_n, \quad \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2), \quad \boldsymbol{\alpha} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{1.8}$$

To see that these models are equivalent, define $f_{\boldsymbol{\alpha}}(x) = \boldsymbol{\alpha}^\top x$ in eq. (1.8). As $f_{\boldsymbol{\alpha}}$ is a linear transformation of a Gaussian random variable, it is a Gaussian process, so it suffices to check that the first two moments coincide with those of the Gaussian process in eq. (1.3) with $\mu \equiv 0$ and a linear kernel.

$$\mathbb{E}[f_{\boldsymbol{\alpha}}(x)] = \mathbb{E}[\boldsymbol{\alpha}^\top]x = 0 \quad \text{and} \tag{1.9}$$

$$\mathbb{E}[f_{\boldsymbol{\alpha}}(x) f_{\boldsymbol{\alpha}}(x')] = x^\top \mathbb{E}[\boldsymbol{\alpha}\boldsymbol{\alpha}^\top] x' = k(x,x'), \tag{1.10}$$

so $f_{\boldsymbol{\alpha}}$ is the Gaussian process with mean function 0 and kernel $k(x,x') = x^\top x$ as claimed.

Samples from the prior (and hence also from the posterior) are linear functions with probability 1. Often this is a useful modeling assumption, and leads to interpretable models if each dimension of the inputs has a physical meaning. However, this model cannot capture non-linear patterns in the data, and as such is overly simplistic for many modeling tasks.

Squared Exponential Kernel

The squared exponential (SE), radial basis function (RBF) or exponentiated quadratic (EQ) kernel is defined by

$$k(x,x') = \exp\left(-\frac{1}{2}\|x - x'\|^2\right). \tag{1.11}$$

The squared exponential kernel is *stationary*, meaning that $k(x,x') = k(0, x - x')$. This implies that the prior is a stationary stochastic process. Unlike the linear kernel, the squared exponential kernel is

an example of a universal kernel (Micchelli et al., 2006), meaning that any continuous function on a compact subset of $\mathbb{R}^D$ can be uniformly approximated by linear combinations of functions of the form $\{k(x,\cdot)\colon x \in \mathbb{R}^D\}$, that is by functions in

$$\operatorname{span}\{k(x,\cdot)\colon x \in \mathbb{R}^D\}. \tag{1.12}$$

The posterior mean is in the span of these features with coefficients depending on the data (eq. 1.6). Moreover, for this kernel and a sufficiently small noise, the posterior mean will nearly interpolate the training data. This provides some indication that the Gaussian process model with this kernel will be able to model a wide range of functions. However, both the posterior mean and sample functions from the prior and posterior are almost surely *smooth* meaning that they are infinitely differentiable. At times the assumption that a latent function is smooth is unrealistic for real-world data and the existence of only finitely many derivatives may be more realistic (Stein, 2012).

Matérn kernel

The Matérn kernel with smoothness parameter $\nu > 0$ is defined by

$$k_\nu(x,x') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\|x-x'\|_2\right)^\nu K_\nu\left(\sqrt{2\nu}\|x-x'\|\right), \tag{1.13}$$

with K_ν a modified Bessel function of the second kind. When the smoothness parameter ν has fractional part $1/2$, k_ν can be written in terms of elementary functions (Rasmussen and Williams, 2006, Equation 4.16). Like the squared exponential kernel, Matérn kernels with any degree of smoothness are universal (Micchelli et al., 2006). Moreover, as $\nu \to \infty$, the Matérn kernel converges to the squared exponential kernel. In general, the posterior mean of a Gaussian process with prior mean $\mu \equiv 0$ and Matérn kernel with smoothness parameter ν is $\lfloor \nu \rfloor$-times continuously differentiable. For $\nu > D/2$, sample paths defined on sufficiently nice subsets of $\mathbb{R}^D$ are $(\lceil \nu - D/2 \rceil - 1)$-times differentiable (Kanagawa et al., 2018, Remarks 2.10 and 2.11, Corollary 4.15).

Asymptotically (as $N \to \infty$) properties such as generalization and ease of approximation of Gaussian process regression models depend primarily on differentiability of the prior process. This leads to qualitative differences in the analysis of models using Matérn and squared exponential kernels. In practice, for finite samples sizes, both the generalization properties and ease of approximation for Gaussian process regression depends heavily on hyperparameters such as the lengthscale matrix $\mathbf{L}$ and the likelihood variance σ^2. As observed in Rasmussen and Williams (2006, Chapter 4), distinguishing models using a Matérn kernel with a high degree of smoothness from models using the squared exponential kernel can be nearly impossible with only finite data.

1.2.2 Mercer's Theorem and Bochner's Theorem

While defining a kernel function explicitly in terms of an expression for $k(x,x')$ is useful for evaluating the covariance of function values at several inputs, other representations of the kernel can give insight into properties of the prior. We now review Mercer's theorem and Bochner's theorem, two of the best-known representations of kernel functions.

Mercer's Theorem

Under broad assumptions, a kernel can be decomposed into a *countable* sum of features with non-negative weights. Mercer's theorem establishes sufficient conditions for the kernel function to have a decomposition

$$k(x,x') = \sum_{m=1}^{\infty} \lambda_m \phi_m(x) \phi_m(x'), \tag{1.14}$$

where the functions $\{\phi_m\}_{m=1}^{\infty}, \phi_m : \mathcal{X} \to \mathbb{R}$ are orthonormal in an appropriate sense and the $\{\lambda_m\}_{m=1}^{\infty}$ satisfy $\lambda_m \geq 0$ and $\sum_{m=1}^{\infty} \lambda_m < \infty$.

A kernel will generally have many such decompositions. In order to specify a particular decomposition, we will also need to specify a measure ρ on $\mathcal{X}$.[3] We generally assume that ρ is a probability measure related to the distribution of covariates. We define the space of (equivalence classes of) functions

$$[f] \coloneqq \{g : \mathcal{X} \to \mathbb{R} \text{ s.t. } \int (f-g)^2 d\rho = 0\}, \tag{1.15}$$

$$L^2(\mathcal{X},\rho) \coloneqq \left\{ [f] \text{ s.t. } f : \mathcal{X} \to \mathbb{R}, \int f^2 d\rho < \infty \right\}. \tag{1.16}$$

Moving forward, we follow the standard abuse of notation and denote elements of $L^2(\mathcal{X},\rho)$ by representative functions f, instead of equivalence classes $[f]$. $L^2(\mathcal{X},\rho)$ is a Hilbert space equipped with the inner product

$$\langle f, g \rangle_{L^2(\mathcal{X},\rho)} \coloneqq \int f g d\rho. \tag{1.17}$$

It is with respect to the inner product eq. (1.17) that the $\{\phi_m\}_{m=1}^{\infty}$ are orthonormal.

Intuition for the representation eq. (1.14) can be gained by considering the closely-related Karhunen-Loève expansion of the prior,

$$f(x) \overset{\mathrm{d}}{=} \sum_{m=1}^{\infty} \sqrt{\lambda_m} \phi_m(x) \alpha_m \tag{1.18}$$

[3] We assume $(\mathcal{X},\Sigma)$ is a measurable space, and leave the σ-algebra, Σ, implicit. In many instances $\mathcal{X}$ will have a topology and Σ is the Borel σ-algebra.

where $\alpha_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)$ and $\overset{\text{d}}{=}$ denotes equality in distribution.[4] From eq. (1.18), a Gaussian process with a kernel that has a Mercer decomposition can be viewed as a featurized linear model with a standard Gaussian prior and features $\{\sqrt{\lambda_m}\phi_m\}_{m=1}^{\infty}$.

Before giving a formal statement of Mercer's theorem, we state the form of this decomposition for the kernels considered in the previous section.

Mercer's Decomposition for Linear Kernels

Let ρ be any probability measure on $\mathbb{R}^D$ with finite first two moments. For $1 \le d \le D$ define $\pi_d : \mathbb{R}^D \to \mathbb{R}$ to be projection onto coordinate d, that is $\pi_d(x) = x_d$. Then

$$k(x,x') = x^\top x' = \sum_{d=1}^{D} x_d x'_d = \sum_{d=1}^{D} \pi_d(x)\pi_d(x'). \tag{1.19}$$

This gives a representation of the kernel as a sum of features. However, the functions π_d are not generally orthonormal in $L^2(\mathbb{R}^D, \rho)$. In particular,

$$\int \pi_d(x)\pi_{d'}(x) d\rho(x) = \mathbb{E}_{x\sim\rho}[x_d x_{d'}] \tag{1.20}$$

so in the case when $\mathbb{E}_\rho[x] = 0$, orthogonality holds if any only if the dimensions of x are uncorrelated under ρ, and the features are normalized if x_d has variance 1 for all d. In order to go from eq. (1.19) to a Mercer representation of the linear kernel, we need to find an orthonormal set of features.

To do this, perform Gram-Schmidt on the π_d to obtain an orthonormal set in $L^2(\mathcal{X}, \rho)$.[5] Call the resulting functions $\{\phi_m\}_{m=1}^{D}$. We can extend this orthonormal set to an orthonormal basis in $L^2(\mathcal{X}, \rho)$, $\{\phi_m\}_{m=1}^{\infty}$. Perform a change of basis from $\{\pi_d\}_{d=1}^{D}$ to $\{\phi_m\}_{m=1}^{D}$ in eq. (1.19) to obtain

$$k(x,x') = \sum_{m=1}^{D} \lambda_m \phi_m(x)\phi_m(x') = \sum_{m=1}^{\infty} \lambda_m \phi_m(x)\phi_m(x'), \tag{1.21}$$

where $\{\lambda_m\}_{m=1}^{D}$ is determined by the change of basis and $\lambda_m = 0, \forall m > D$. Equation (1.21) is the Mercer representation of the linear kernel with respect to ρ.

Mercer's Decomposition for Squared Exponential Kernels

In the case of a squared exponential kernel and $\rho = \mathcal{N}(0, s^2)$, Mercer's Theorem (eq. 1.14) can be computed explicitly (Zhu et al., 1997):

$$\lambda_m = \sqrt{\tfrac{1}{2A}} B^{-(m-1)}, \quad \phi_m(x) = \exp(-\tfrac{1}{2}(c - \tfrac{1}{2})x^2) H_{m-1}(\sqrt{c}x), \tag{1.22}$$

$$c = \sqrt{\tfrac{1}{4} + s^2}, \quad A = \tfrac{1}{2}(\tfrac{1}{2} + s^2 + c), \quad B = \frac{2A}{s^2}, \tag{1.23}$$

[4] Convergence holds in the stronger, mean square sense. See, for example, (Wang, 2008) for details.

[5] The assumption of finite moments ensures $\pi_d \in L^2(\mathcal{X}, \rho)$.

and

$$H_m(x) = (-1)^m \exp(x^2) \frac{d^m}{dx^m} \exp(-x^2), \tag{1.24}$$

is the Hermite polynomial of degree m. From eq. (1.22), we see that the λ_m decay exponentially fast, with the exponent depending on $B > 1$. B decreases as the variance of the covariates increases, meaning the eigenvalues $\{\lambda_m\}_{m=1}^{\infty}$ decay more slowly if the data is more spread out. Formulae for kernels with a lengthscale not equal to one can be computed by noting that changing the lengthscale is equivalent to changing the standard deviation of the ρ.

In the case when $\mathcal{X} = \mathbb{R}^D$ the squared exponential kernel is *multiplicatively separable* or simply *separable*, meaning

$$k(x,x') = \prod_{d=1}^{D} k_d(x_d, x'_d) \tag{1.25}$$

where k_d is a one-dimensional squared exponential kernel defined on the d^{th} dimension of x. For separable kernels, and if the measure ρ is independent across dimensions, both the eigenvalues and eigenvectors of the k are the products of the eigenvalues of the k_d. From this and a counting argument, we can conclude that in D dimensions, we have $\lambda_m = O(a^{-m^{1/D}})$ for some $a > 1$ (Seeger et al., 2008).

For measures with compact support and bounded density with respect to Lebesgue measure, the ϕ_m and λ_m are not generally known in closed-form. However, in this case it can be shown that for any $a > 1$, $\lambda_m = O(a^{-m^{1/D}})$, where the implicit constant may depend on both a and D as well as properties of the kernel and measure, but not on m.

Mercer's Decomposition for Matérn Kernels

For Matérn kernels and Lebesgue measure on $[-1,1]$ the eigenvalues, $\{\lambda_m\}_{m=1}^{\infty}$ can be computed as the solution to a set of transcendental system of equations (Youla, 1957). The eigenfunctions are sinusoidal, with the frequencies also given as the solution to a transcendental system of equations. In all but the simplest cases, solving the necessary systems of equations is non-trivial. However, asymptotic statements can be made about the λ_m, as a special case of a more general theory relating smoothness of a stationary kernel to decay of the corresponding λ_m (discussed later in this section), leading to the conclusion that for compactly supported distributions with bounded density on $\mathbb{R}^D$, $\lambda_m = O(m^{-(2\nu+D)/D})$. Contrast this to the case of the squared exponential kernel: the infinitely differentiable squared exponential kernel leads to eigenvalues that decay exponentially quickly, while the finitely-many-times differentiable Matérn kernels eigenvalues exhibit polynomial decay, with the degree depending on the smoothness parameter of the kernel.

Conditions for Mercer's Theorem

We will refer to any kernel for which the decomposition eq. (1.14) holds in a suitably strong sense, and with $\{\phi_m\}_{m=1}^{\infty}$ orthonormal in $L^2(\mathcal{X},\rho)$ as a *Mercer kernel with respect to* ρ or simply a *Mercer kernel* if ρ is clear from the context. As a formal statement of Mercer's theorem relies heavily on results from functional analysis, this section is more technical than the rest of chapter 1. The conclusion of this discussion is that if ρ is a probability measure, a sufficient condition for k to be a Mercer kernel with respect to ρ is that k is a continuous function and $\int k(x,x)d\rho(x) < \infty$.

Consider the (linear) integral operator $T_{k,\rho}\colon L^2(\mathcal{X},\rho) \to L^2(\mathcal{X},\rho)$ defined by

$$T_{k,\rho}f(x) = \int f(x')k(x,x')d\rho(x'). \tag{1.26}$$

Under suitable assumptions on k and ρ, $T_{k,\rho}$ is well-defined, positive, self-adjoint and compact. If this holds, the spectral theorem guarantees the existence of a sequence of eigenvalues $\{\lambda_m\}_{m=1}^{\infty}$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$ and corresponding orthonormal eigenfunctions $\phi_m \in L^2(\mathcal{X},\rho)$ that can be chosen to be continuous. Mercer's theorem can be phrased as stating that k has a representation as an ℓ^2-inner product in the feature space defined using the eigenfunctions and eigenvalues of $T_{k,\rho}$. In order to state a general form of Mercer's theorem, we assume $\mathcal{X}$ is a topological space equipped with its Borel σ-algebra. In this case, we can define the *support* of a probability measure ρ, denoted by $\operatorname{supp}(\rho)$, to be the set of all $x \in \mathcal{X}$ such that ρ assigns any open set containing x positive probability.

Theorem 1.1 (Mercer's Theorem, Steinwart and Scovel, 2012, Lemma 2.3 and Corollary 3.5)**.** *Let $(\mathcal{X},\Sigma,\rho)$ be a probability space, with $\mathcal{X}$ a Hausdorff space and Σ its Borel σ-algebra. Let $k\colon \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ be a positive definite, continuous kernel such that*

$$\int k(x,x)d\rho(x) < \infty. \tag{1.27}$$

Then, $T_{k,\rho}$ is a self-adjoint, compact operator and

$$k(x,x') = \sum_{m=1}^{\infty} \lambda_m \phi_m(x)\phi_m(x'), \tag{1.28}$$

with $\{\lambda_m\}_{m=1}^{\infty}$ eigenvalues of $T_{k,\rho}$ and $\{\phi_m\}_{m=1}^{\infty}$ orthonormal eigenfunctions of $T_{k,\rho}$. Further $\sum_{m=1}^{\infty}\lambda_m < \infty$. For any fixed $x \in \operatorname{supp}\rho$ convergence in eq. (1.28) is absolute for all $x' \in \operatorname{supp}\rho$. Moreover, convergence is uniform in x and x' on all sets $A\times A \subset \operatorname{supp}\rho \times \operatorname{supp}\rho$ such that A is compact.

As eq. (1.26) only depends on k, ρ–almost everywhere,[6] it should not be surprising that Mercer's theorem need not provide a valid representation of the kernel outside the support of ρ.

Remark 1.2. *This version of Mercer's theorem is more general than classical statements of Mercer's theorem (it does not require a compact domain), and gives convergence on the entire support of ρ as*

[6] By ρ–almost everywhere, we mean on a set $A \subset \mathcal{X}$ such that $\rho(A) = 1$.

opposed to, for example, König (2013, 3.1.a) which establishes convergence $\rho \times \rho$–almost everywhere. While this may seem to be a technical detail, it will be crucial in later proofs. We will want to claim,

$$k(x,x) = \sum_{m=1}^{\infty} \lambda_m \phi_m(x)^2. \tag{1.29}$$

In particular, the diagonal of $\mathcal{X} \times \mathcal{X}$ may be a set of measure zero, so this is not immediately implied by the result of König (2013). As a cautionary note, the continuity assumption for the kernel is essential. To see this, we recall Example 9 from Steinwart and Scovel (2012). Consider the kernel

$$k(x,x') = \begin{cases} 1 & x = x' \\ 0 & \text{otherwise}. \end{cases} \tag{1.30}$$

If ρ is non-atomic, $k(x,x') = 0$ $(\rho \times \rho)$–almost everywhere and so $T_{k,\rho}$ is the zero operator. Mercer's expansion is valid in $L^2(\mathcal{X} \times \mathcal{X}, \rho \times \rho)$; however, attempting to restrict Mercer's decomposition to the diagonal in this case would lead to the incorrect conclusion that $k(x,x) = 0$.

Mercer's Theorem and the Complexity of the Model

Under the assumption that the data is a collection of random variables distributed according to eq. (1.3) and if the covariates are independently and identically distributed with measure ρ, the generalization properties of the Gaussian process are intimately related to decay of the λ_m (e.g. Seeger et al., 2008; Sollich, 1999). Roughly speaking, simple models and data-generating processes imply rapid decay of eigenvalues in Mercer's theorem, and also good generalization to unseen data if the model is well-specified.

For a linear model with D dimensions, for all $m \geq D$, $\lambda_m = 0$ (eq. 1.21). More generally, integrating the diagonal in Mercer's theorem and using that $\|\phi_m\| = 1$ shows

$$\sum_{m=1}^{\infty} \lambda_m = \int_{x \in \mathcal{X}} k(x,x) d\rho(x), \tag{1.31}$$

so the sum of the eigenvalues is the average variance of the prior process with respect to the input measure ρ. For any $M \in \mathbb{N}$, define

$$k_M = \sum_{m=1}^{M} \lambda_m \phi_m(x) \phi_m(x'). \tag{1.32}$$

k_M defines a finite dimensional linear-Gaussian model with dimension at most M. Integrating over the diagonal

$$\|k(\cdot,\cdot) - k_M(\cdot,\cdot)\|_{L^2(\mathcal{X},\rho)} = \sum_{m=M+1}^{\infty} \lambda_m. \tag{1.33}$$

We can think of this as the average amount of variance in the model (eq. 1.3) that is not explained by the M-dimensional model formed by replacing k with k_M. The approximation of k by k_M is the optimal a priori linear approximation to the original model (Zhu et al., 1997).

This interpretation leads to the conclusion that if the covariates follow the measure ρ and the eigenvalues of $T_{k,\rho}$ decay rapidly, most of the variance in the model can be explained by a low-dimensional model. We conclude that such a model is relatively simple. Conversely, slow decay of eigenvalues indicates a more flexible model. We emphasize that this notion of model complexity depends not just on the prior, but also on the distribution of covariates. Recall that for the Matérn kernel, the smoothness parameter ν controls the asymptotic decay of eigenvalues if the inputs are uniformly distributed on a compact domain. Considering connections between sums of eigenvalues and the model complexity, we see in this case that smoother Matérn kernels correspond to simple models, while rougher Matérn kernels lead to complicated models that are hard to approximate via low-dimensional linear models.

Bochner's Theorem

For stationary kernels defined on $\mathbb{R}^D$ including the squared exponential and Matérn kernels, *Bochner's theorem* gives an alternative representation of the kernel as an integral instead of the sum representation given by Mercer's theorem (eq. 1.14).

Theorem 1.3 (Bochner's Theorem, Wendland, 2004, page 67)**.** *Let k be a stationary, continuous, positive semi-definite kernel. Define $\kappa(x) = k(0,x)$. Suppose $\kappa \in L^2(\mathbb{R}^D, \lambda)$, where λ denotes Lebesgue measure. Then there exists a non-negative, finite (Borel) measure γ, such that*

$$\kappa(x) = \int e^{2\pi i x^\top \omega} d\gamma(\omega). \tag{1.34}$$

For a proof of theorem 1.3, see Wendland (2004, pages 72-74). The converse is also true: every square-integrable, positive semi-definite function from $\mathbb{R}^D \to \mathbb{C}$ arises as the Fourier inverse of a non-negative, finite measure, though we will not make use of this fact.

When γ has a density with respect to Lebesgue measure, we denote it by s. This density is the inverse Fourier transform of κ, and can be computed in many cases of interest. We will see that Bochner's theorem is closely related to Mercer's theorem, and the relative ease of computing the spectral density makes Bochner's theorem a useful intermediate tool for understanding properties of the eigenvalues in Mercer's theorem for stationary kernels.

Bochner's Theorem for the Squared Exponential Kernel

The inverse Fourier transform of $k(0, x - x')$ for a squared exponential kernel (eq. 1.11) is

$$s(\omega) = \int e^{-2\pi i x^\top \omega} \exp(-\tfrac{1}{2}\|x\|_2^2) dx = \sqrt{2\pi} \exp(-2\pi^2 \|\omega\|_2^2). \tag{1.35}$$

$s(\omega)$ is proportional to a Gaussian density and therefore decays rapidly to 0 as $\|\omega\| \to \infty$.

Bochner's Theorem for the Matérn Kernel

The spectral density of Matérn kernels also has a closed form (Rasmussen and Williams, 2006, Chapter 4):

$$s_\nu(\omega) = \frac{(2\pi)^{D/2}\Gamma(\nu + D/2)}{\Gamma(\nu)\nu^{D/2}} \left(1 + \frac{2}{\nu}\pi^2\|\omega\|^2\right)^{-(\nu+D/2)}. \tag{1.36}$$

The right-hand side of eq. (1.36) is (up to a change of variables) proportional to the density of a Student's t-distribution on $\mathbb{R}^D$ with 2ν degrees of freedom.

For the Matérn kernel, as $\|\omega\| \to \infty$ the spectral density only decays polynomially fast with the degree depending on the input dimension and smoothness parameter.

Connections between Mercer's Theorem and Bochner's Theorem

As alluded to earlier in this section, the smoothness of the kernel, the decay of the spectral density in Bochner's theorem, and the decay of the λ_m in Mercer's theorem are related. As an example, consider the case when $D = 1$ and ρ is uniform measure on $[0, 1]$. For sufficiently nice, stationary kernels, the following statements can be though of as roughly equivalent:[7]

1. $\kappa(x) \coloneqq k(0, x)$ has continuous derivatives up to order t,
2. $s(\omega) = O((1 + |\omega|)^{-(t+3/2)})$,
3. $\lambda_m = O(m^{-(t+3/2)})$.

The relationship between smoothness, spectral density and eigenvalue in Mercer's theorem is qualitatively illustrated in figure 1.3.[8]

A connection between the first two statements can be made rigorous with some assumptions on the decay of the derivative of κ via the connection between Sobolev spaces and their Fourier transforms (Adams and Fournier, 2003, 7.62). A connection between the second two statements is made precise, with some additional assumptions on the form of the spectral density of the kernel, in Widom (1963, Theorem 2).

In this thesis, we will primarily be interested in the eigenvalues of the kernel coming from Mercer's theorem, due to the connection between these eigenvalues and the complexity of the model discussed earlier in this section. However, these are generally difficult to calculate, whereas observing properties of the kernel function and its derivative, or computing the spectral measure of the kernel via the inverse Fourier transform, is often much easier. Table 1.1 summarizes properties of kernels discussed in this section.

[7] Making this entirely correct requires additional assumptions.

[8] Sample path smoothness is distinct from but related to smoothness of κ.

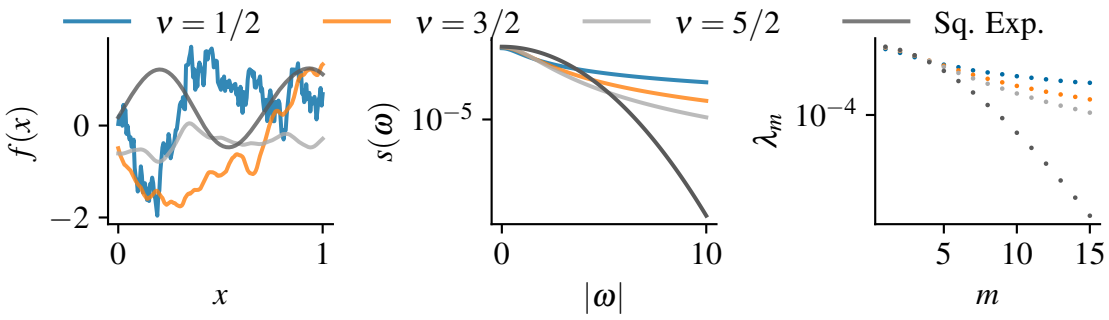


Fig. 1.3 From left to right, samples from a Gaussian process with Matérn ν kernel for $\nu \in \{\frac{1}{2}, \frac{3}{2}, \frac{5}{2}\}$ and a squared exponential kernel, the spectral density of each kernel (plotted on a log-linear scale) and the first 20 eigenvalues of the associated operator (plotted on a log-linear scale and estimated numerically). Smooth kernels, such as the squared exponential kernel, have rapidly decaying spectral density and rapidly decaying eigenvalues. Rough kernels, such as the Matérn 1/2 kernel, have rough sample paths and slowly decaying spectral densities and eigenvalues.

Table 1.1 Common kernels defined on $\mathbb{R}^D$, sample path properties, the corresponding spectral densities $s(\omega)$ (eq. 1.34) and the decay of eigenvalues λ_m of the associated operator defined in eq. (1.26). We define $r = \|x - x'\|_2$. The eigenvalue decay assumes the measure has compact support and bounded Lebesgue density. $a > 1$ is an arbitrary constant. Implicit constants may depend on properties of the data distribution, dimension and any parameters besides m.

Name	$k(x,x')$	Sample paths	$s(\omega)$	λ_m
Linear	$x^\top x'$	Linear	Non-stationary	0 if $m > D$
Matérn ν	$\frac{2^{1-\nu}(2\nu)^{\nu/2}}{\Gamma(\nu)} r^\nu K_\nu\left(\sqrt{2\nu} r\right)$	$\lceil \nu - \frac{D}{2} \rceil - 1$ diff.	$\frac{(2\pi)^{D/2}\Gamma(\nu+D/2)}{\Gamma(\nu)\nu^{D/2}}(1+\frac{2\pi^2}{\nu}\|\omega\|_2^2)^{-\nu-\frac{D}{2}}$	$O(m^{\frac{-(2\nu+D)}{D}})$
SE	$\exp(-\frac{1}{2}r^2)$	Analytic	$(2\pi)^{D/2}\exp(-2\pi^2\|\omega\|_2^2)$	$O(a^{-m^{1/D}})$

1.2.3 Model Selection and the Marginal Likelihood

Given the range of possible assumptions about the data that the kernel function can encode, the practitioner faces a non-trivial choice in specifying the model (eq. 1.3). Ideally, the practitioner would be well-informed about the data-generating process and able to fully specify the kernel. In practice, this is usually not the case. Instead, a family of kernels, such as the set of squared exponential kernels with varying kernel variances σ_k^2 and diagonal lengthscale matrices $\mathbf{L}$, is specified based on some vague prior knowledge of the data, and the particular kernel is chosen from this family based on the observed data. There are several approaches of this form, including hierarchical Bayes (placing a prior over parameters), cross-validation and maximum marginal likelihood (MML). We focus on model selection via maximum marginal likelihood.

The Marginal Likelihood

The log marginal likelihood is

$$\mathcal{L}(\theta) = \log \int \ell_{\mathcal{D}}(f) dP, \tag{1.37}$$

where $\ell_{\mathcal{D}}$ is the probability density of the observed data given the latent function f, as in eq. (1.4). This is the probability density of the data after integrating out the latent function f and viewed as a function of the model hyperparameters, θ. The right-hand side of eq. (1.37) depends on θ through the prior and likelihood, though we suppress this dependence. Model hyperparameters may include the noise variance, σ^2, as well as parameters of the kernel such as the kernel variance σ_k^2 and lengthscale matrix $\mathbf{L}$.

Unlike many Bayesian models in which the marginal likelihood is intractable, the log marginal likelihood of Gaussian process regression has a closed-form:

$$\mathcal{L}(\theta) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log\det \mathbf{K} - \frac{1}{2}\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}. \tag{1.38}$$

Recall that $\mathbf{K} = \mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}$. On the right-hand side of eq. (1.38) we have suppressed the dependence of $\mathbf{K}$ on θ. Equation (1.38) is the density of an N-dimensional Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{K}$ evaluated at $\mathbf{y}$.

Maximum marginal likelihood selects the model hyperparameters θ via (approximately) solving the optimization problem

$$\theta_{\text{MML}} \in \arg\max_{\theta} \mathcal{L}(\theta). \tag{1.39}$$

In the case of Gaussian process regression, an exact solution to eq. (1.39) is not generally tractable, and so local optimization is performed using gradient based methods. Figure 1.4 shows the impact of maximum marginal likelihood on posterior samples on a toy dataset.

1.2.4 Properties of Maximum Marginal Likelihood Model Selection

It is not hard to imagine alternative methods for performing model selection, for example k-fold cross-validation based on a test metric. We briefly review several favorable properties of maximum marginal likelihood.

Maximum marginal likelihood differs slightly from an idealized Bayesian approach, in which unknown parameters are treated probabilistically. The most compelling justification of maximum marginal likelihood for model selection in Gaussian process models is the empirical evidence that the method works well. For example, Rasmussen and Williams (2006, Chapter 5) provides a comparison between maximum marginal likelihood and leave-one-out cross-validation which, while far from conclusive, provides evidence the marginal likelihood gives a reasonable proxy for generalization of the model to unseen data. A more philosophical justification for maximum marginal likelihood advocated in, for example, Rasmussen and Ghahramani (2001) and MacKay (2003), hinges on its ability to

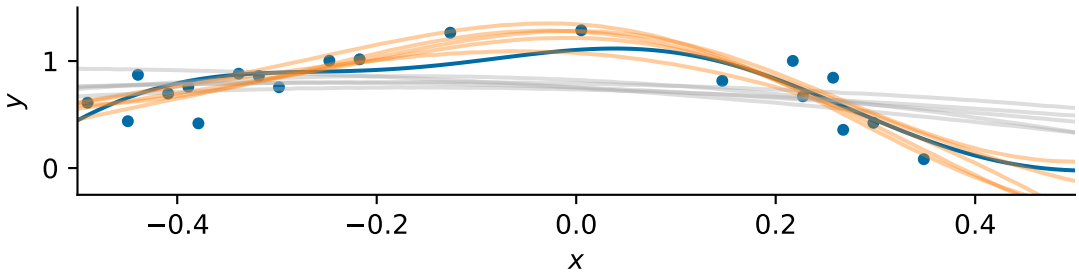


Fig. 1.4 An illustration of maximum marginal likelihood. The dataset and latent function used to generate it are shown in blue. A squared exponential kernel is selected to model the data. Initially a noise variance 0.04, kernel lengthscale of 1.5, and kernel variance of 1.0 is used to model the data. We see that the posterior under-fits (gray curves). The log marginal likelihood is then maximized with respect to these three parameters (eq. 1.39), and the samples from the resulting posterior are shown (orange curves). These samples model trends in the data better.

automatically trade off between complexity of the model and ability to fit the data. While a useful heuristic, this can only be pushed so far. In the case when the family of kernels is very large relative to the dataset size, maximum marginal likelihood amounts to the under-constrained problem of maximum likelihood estimation of the covariance matrix of an N-dimensional Gaussian given a single observation, which leads to severe over-fitting. From this extreme example, we see that we must be somewhat careful in designing the space of kernels we consider when performing maximum marginal likelihood: the space of kernels considered must contain kernels that lead to models that fit the data well, but not be so large that the posterior over-fits.

Asymptotic Properties of Maximum Marginal Likelihood An alternative perspective that sheds light on maximum marginal likelihood is to take a frequentist perspective, and assume the data is a random variable. This leads to questions such as: 1. Is maximum marginal likelihood consistent, that is, with enough data do we recover the hyperparameter values used to generate the data? 2. If it is consistent what is the (asymptotic) variance of the maximum marginal likelihood solution around the 'true' parameter values? 3. If it is not consistent, is the predictor obtained still 'good', in a quantitative sense?

The first question is well-studied; the answer depends on the class of kernels considered and assumptions about the data generating process. For stationary kernels on $\mathbb{R}^D$, if the covariates are observed over an increasing domain (e.g. time series), the set of parameters considered is compact, and the kernel satisfies certain regularity conditions, then all parameters can be identified and the distribution of the estimator around the true parameter value is asymptotically Gaussian (Bachoc, 2021). On the other hand, if the covariates are observed over a fixed, bounded domain, some combinations of parameters cannot be identified even asymptotically (Bachoc, 2021; Stein, 2012). However, the parameter settings found generally lead to similar predictors as the true parameter values, and in certain

instances the prediction made with the parameters found via maximum marginal likelihood may still be asymptotically optimal in a suitable sense (Stein, 2012, Chapter 6). In certain instances, the asymptotic variance of a modified maximum marginal likelihood procedure can be shown to be smaller than those found by a cross-validation procedure (Stein, 1990), suggesting it is preferable as a method of model selection. However, under model mis-specification, cross-validation is generally believed to be more robust (Bachoc, 2013; Stein, 2012).

While the asymptotic properties of maximum marginal likelihood are theoretically interesting, it is unclear how relevant they are for practical dataset sizes, which may contain anywhere from dozens to millions of datapoints. In this thesis, we will be largely interested in scalable approximations to Gaussian processes, so it is reasonable to think the asymptotic results are more relevant for our discussion than they are for classical examples of Gaussian process applications. However, there may still be a gap between theory given through asymptotic results and more reasonable finite dataset sizes.

Example: Estimating a Scale Parameter We consider a simple example of maximum likelihood estimation where the distribution of the estimator can be characterized for finite datasets. We assume we want to estimate a single parameter that controls the scale of the model. This case is misleadingly simple; as we shall see it is the same as maximum likelihood estimation of the variance of a collection of independent, identically distributed Gaussian random variables, and does not depend on the structure of the Gaussian process or distribution of covariates in an interesting way.

We assume for simplicity that the covariates are fixed, and the generative process has the form (eq. 1.3)

$$y_n = f(x_n) + \varepsilon_n, \quad \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \theta_0 \sigma^2), \quad f \sim \mathcal{GP}(0, \theta_0 k) \tag{1.40}$$

for some unknown $\theta_0 > 0$. In this case, the maximum marginal likelihood solution for θ exists, is unique, and is given by

$$\theta_{\text{MML}} = \frac{\mathbf{y}^\top \mathbf{K}^{-1} \mathbf{y}}{N}, \tag{1.41}$$

(see Rasmussen and Williams 2006, Excercise 5.6.1). Taking expectations on both sides of eq. (1.41) and using linearity and the cyclic property of the trace (proposition A.10)

$$\mathbb{E}[\theta_{\text{MML}}] = \theta_0, \tag{1.42}$$

meaning the maximum likelihood estimate for θ_0 is unbiased. Writing $\mathbf{y} \overset{\text{d}}{=} \sqrt{\theta_0}\mathbf{L}\mathbf{z}$, with $\mathbf{L} = \mathbf{K}^{1/2}$ and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

$$\frac{N}{\theta_0}\theta_{\text{MML}} \overset{\text{d}}{=} \|\mathbf{z}\|^2 \sim \chi^2(N), \tag{1.43}$$

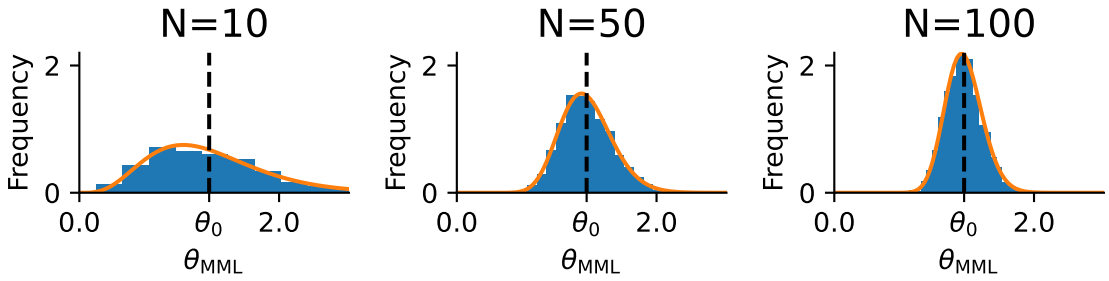


Fig. 1.5 We compare the maximum marginal likelihood estimate of a scale parameter (θ_{MML}) with data generated from the same model with scale parameter θ_0 (dotted line). The orange curve is the probability density function of the sampling distribution for the maximum likelihood estimate, while the bins represent normalized counts for θ_{MML} computed by generating 2000 datasets from the model and computing the maximum marginal likelihood estimate (eq. 1.41). As N increases, the distribution concentrates around the θ_0. In other words, estimation of the scale parameter is consistent.

where $\chi^2(N)$ denotes a χ^2-distribution with N degrees of freedom. As the distribution of θ_{MML} is independent of the covariates in this setup, the same conclusion holds if the covariates are random variables. Properties such as consistency and asymptotic variance can readily be determined from eq. (1.43) and properties of the χ^2-distribution as $N \to \infty$. Figure 1.5 illustrates the distribution of θ_{MML} for several sizes of datasets N. We see even that for $N = 100$, the distribution is reasonably concentrated around its true value, suggesting that maximum marginal likelihood often succeeds at finding a reasonable approximation to an unknown scale parameter in the case of an otherwise well-specified model.

Unfortunately, the above analysis does not generalize to other kernel hyperparameters and, to the best of the author's knowledge, a non-asymptotic theory for maximum marginal likelihood in Gaussian process regression models remains largely elusive.

1.2.5 Workflow for Model Selection and Inference in Gaussian Process Regression

Given a dataset, the prototypical workflow that we will consider for Gaussian process regression is:

Workflow 1 (Model Selection and Inference in Gaussian Process Regression)**.**

1. *Perform any pre-processing on the data. This may involve standardizing the covariates and response variables.*
2. *Select a family of kernels to consider as potential priors parameterized by θ. For example, consider the set of squared exponential kernels with unknown kernel variance and lengthscale matrix and define $\theta = \{\sigma_k^2, \mathbf{L}, \sigma^2\}$.*
3. *Solve the optimization problem,*

$$\theta_{\text{MML}} \in \arg\max_{\theta} \mathcal{L}(\theta), \tag{1.44}$$

with $\mathcal{L}(\theta)$ as in eq. (1.38).

4. *Perform inference and make predictions using eq. (1.6).*

Typically, the optimization problem in step 3 of workflow 1 is non-convex, and a local optimum is found using gradient based methods, for example L-BFGS (Liu and Nocedal, 1989; Nocedal, 1980). Several different initializations may be performed to better approximate the maximization problem.

It may be useful to augment workflow 1 with additional steps that check modeling assumptions, for example a prior sensitivity analysis as advocated in Stephenson et al. (2022). However, we will focus on the core workflow described above, leaving it to the practitioner to determine appropriate checks of modeling assumptions as well as sensitivity of conclusions to these assumptions.

1.3 Computational Considerations

Having outlined the modeling workflow (workflow 1), we now turn to the computational costs of this procedure. We first discuss computational considerations for a specific, commonly used implementation, which highlights scaling properties of algorithms used in practice. We then take a brief detour to discuss asymptotically better scaling implementations that are generally impractical. The discussion will focus on upper bounds; the simplest lower bound on algebraic time complexity (and the only that we are aware of) is that the computational complexity of Gaussian process regression with a generic kernel is $\Omega(N^2)$, as any exact implementation must evaluate all entries in the kernel matrix.

1.3.1 A Practical Summary of Computational Considerations

Many libraries that implement Gaussian process regression rely on computing a Cholesky decomposition of the kernel matrix in order to evaluate the log determinant and to solve a linear system of equations. We provide a detailed accounting of the operations and complexity used in this implementation in terms of the number of floating point operations involved in various operations. While the number of floating point operations can be misleading in terms of computation time (due to failing to account for parallelization and memory access) in this case the operations that require the most floating point operations are also empirically the slowest for reasonably large datasets.

First, we compute the kernel matrix, which has cost N^2A, where A is the number of operations to compute an entry in the kernel matrix. The cost of evaluating an entry in the kernel matrix often depends on the dimension of the dataset, for example in the case of a squared exponential kernel it requires roughly $2D$ multiplications and D additions, assuming a diagonal lengthscale matrix is used. We also store an $N \times N$ symmetric matrix in memory, which is problematic in many instances, particularly when GPUs are used. Next, many implementations involve approximately $N^3/6$ additions and $N^3/6$ multiplications to compute the Cholesky decomposition. Once the Cholesky decomposition, $\mathbf{L}$, has been computed, the log determinant can be computed by summing the log of the diagonal entries of $\mathbf{L}$ then multiplying by 2. This involves N additions and N logarithms of scalars. Finally, we must perform a triangular solve and an inner product to compute $\left(\mathbf{L}^{-1}\mathbf{y}\right)^\top \left(\mathbf{L}^{-1}\mathbf{y}\right)$. The triangular solve involves

roughly $N^2/2$ additions and $N^2/2$ multiplications, while the inner product requires N additions and N multiplications. We do not detail the calculation of the gradient of the log marginal likelihood, but note that it is generally less expensive than computation of the log marginal likelihood itself, assuming the number of parameters is relatively small and the Cholesky decomposition of $\mathbf{K}$ has already been stored during the computation of the objective. All the computations discussed above are numerically stable, so long as the noise variance σ^2 is not too close to 0 relative to the diagonal entries of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. This ensures that $\mathbf{K}$ is well-conditioned, meaning that solutions to systems of equations involving this matrix are robust to small perturbations. Despite the reasonable numerical stability of this procedure, computation is generally performed in double precision, as in single precision the Cholesky decomposition can lead to numerical issues.

When maximum marginal likelihood model selection is performed, both the log marginal likelihood and its gradient must be computed multiple (generally many) times. The precise number of evaluations of the log marginal likelihood and its gradient depend heavily on the family of kernels considered, the parameterization of the kernels, the particular dataset, and the optimizer employed. Empirically, the number of evaluations used by an optimizer such as L-BFGS does not scale with the dataset size for most problems, but can be on the order of several hundred, and generally increases with the number of parameters that we try to estimate (i.e. the dimensionality of θ). We will consider this to be a large constant, although we do not know of a proof that it does not scale with N.

In most applications of the workflow suggested above, model selection requires the vast majority of the computational effort. In order to compute the posterior predictive distribution at T points, we need to evaluate the kernel function at all pairs of these T points, as well as pairs from these points and the observed data. This has complexity $AT(N+T)$. The mean at new test points can be computed by solving an additional $N \times N$ system of equations, $\mathbf{L}^\top \mathbf{v} = (\mathbf{L}^{-1}\mathbf{y})$ and then computing T inner products. Computing the predictive covariance at each new test point involves solving a new $N \times N$ triangular system of equations $\mathbf{L}\mathbf{v} = \mathbf{k}_{\mathbf{x}x}$, taking an inner product and subtracting. Assuming the number of points that we want to predict at is T, we see that the complexity is roughly $TN^2 + T^2N$ operations (assuming we previously stored the Cholesky decomposition of $\mathbf{K}$). Finally, in order to sample from the posterior at T points, we will additionally need to compute a Cholesky decomposition of the posterior covariance at these test points, sample T independent, standard Gaussian variables, and perform a matrix-vector multiplication with a $T \times T$ matrix. For T even moderately large, the Cholesky decomposition will require the most time of these operations, as it requires roughly $T^3/3$ operations.

1.3.2 A Technical Summary of Computational Considerations

The computational complexities discussed above are for a particular, practical, implementation of workflow 1. As a portion of this thesis will be dedicated to discussion of asymptotic (in N) properties of approximate Gaussian process regression, we now mention that asymptotically faster implementations of eq. (1.38) are possible.

The constant of matrix multiplication is defined as the smallest ω such that for any $\varepsilon > 0$, two $N \times N$ matrices can be multiplied using $O(N^{\omega+\varepsilon})$ floating-point operations. Williams (2012) established $\omega < 2.373$. Computation of the log determinant and inverse of a matrix can be efficiently reduced to matrix-matrix multiplication, and therefore for any $\varepsilon > 0$ computed in $O(N^{\omega+\varepsilon})$ floating-point operations (Bürgisser et al., 1997, Chapter 16). Hence, one could accurately say that evaluating eq. (1.38) and eq. (1.6) requires computation $O(N^{2.373})$. Achieving this scaling relies on fast matrix-matrix multiplication. These methods are rarely, if ever, used. For reasonably sized matrices the implicit constants make them generally slower than the Cholesky-based implementation described above.

1.3.3 Computational Considerations for Approximate Gaussian Process Regression

Moving forward, when we discuss the complexity of computing eq. (1.38) and eq. (1.6), we mean the complexity of practical, commonly used algorithms for computing these quantities (section 1.3.1). In general, the $\Theta(N^3)$ time complexity of practical methods, as well as the $\Theta(N^2)$ memory complexity of these methods is cost-prohibitive for fitting Gaussian process regression on datasets with hundreds of thousands of points. As mentioned in section 1.3.1 the most computationally expensive part of workflow 1 is often step 2, in which the log marginal likelihood (eq. 1.38) and its gradients may be evaluated hundreds of times to approximately solve the optimization problem eq. (1.39).

In the next section, we discuss practical approximate solutions that have been proposed to reduce the computational complexity of workflow 1. When we quote the complexity of an approximate method, it will be of practical implementations of the method (just as we do for exact methods), and hence may be asymptotically more pessimistic than the optimal complexity of exact methods of Gaussian process regression, while still being useful. We consider any approximate method with computational cost $o(N^3)$ and space cost $o(N^2)$ to indicate a computational savings relative to exact inference, even if the bound provided is larger than tighter upper bounds on the algebraic complexity of an exact implementation.

Approximate methods may have some collection of parameters that trade-off between computational speed and fidelity of the approximation. We will only be interested in the computational complexity in cases where the approximation method faithfully represents the posterior.

1.4 Properties of Gaussian Process Approximations

Before discussing any particular method for approximating Gaussian process regression, we ask three questions of any approximate method: **Will it work?**, **Did it work?** and **Is it easy to use?**. Because of the structure of Gaussian process regression, we hope for more decisive answers to these questions than we can get in the general case of approximate Bayesian inference, in which decisive answers can be elusive.

1.4.1 What does a Method 'Working' Mean?

In order to discuss what we mean by **Will it work?** and **Did it work?** we first need to clarify what we mean by an approximate method 'working', as well as clarify the workflow we assume is used with an approximate method.

For approximate methods, we will consider the following minor modification of workflow 1:

Workflow 2 (Approximate Model Selection and Inference in Gaussian Process Regression)**.**

1. *Perform any pre-processing on data.*
2. *Select a family of kernels to consider as potential priors, parameterized by* θ.
3. *Select any parameters that control the trade-off between computation and the quality of approximation.*
4. *Solve the optimization problem,*

$$\theta_{\text{approx}} \in \arg\max_{\theta} \tilde{\mathcal{L}}(\theta), \tag{1.45}$$

   with $\tilde{\mathcal{L}}(\theta)$ *is an approximation to eq.* (1.38).
5. *Check the quality of approximate model selection with some diagnostics, depending on the approximation used.*
6. *Perform approximate inference and make predictions using an approximation to eq.* (1.6).
7. *Check the quality of approximate inference and predictions with some diagnostics, depending on the approximation used.*

In practice, the delineation between the steps in this process is unlikely to be as clean as suggested by this workflow. For example, if the checks fail, parameters controlling the quality of approximation need to be adjusted. For example, steps 3-5 may be interwoven, e.g. by checking upper bounds on $|\mathcal{L}(\theta) - \tilde{\mathcal{L}}(\theta)|$ for a candidate θ to ensure the approximation is reliable throughout model selection, adjusting approximation parameters accordingly.

We consider an approximate method to 'work' if the conclusions of inference using workflow 2 closely resemble the conclusion that would have been reached had inference been performed using workflow 1. There are other sensible definitions of an approximate method working, for example that it performs well on a specific task of interest, which may be independent of whether the approximate method accurately approximates Bayesian inference or maximum marginal likelihood model selection. While this latter definition is undoubtedly pragmatic, we prefer the former definition as it delineates cleanly between modeling assumptions and approximate inference.

1.4.2 Will it Work?

We want an approximate method to be applicable to a range of real-world problems. Ideally, a method comes with evidence, empirical or theoretical, that if a practitioner selects to use the method and is

given a typical dataset, the method is likely to work on that dataset. Such guarantees are referred to as 'a priori', as they reason about the quality of an approximate method prior to observing the dataset.

Confidence that a method works can be obtained empirically by running the method on a number of datasets on which ground truth model selection and inference can be performed. If in these instances the ground truth and approximation are similar, we might extrapolate that on larger datasets where the ground truth cannot be computed, the approximation will also be successful.

From the theoretical perspective, a priori guarantees will often rely on some assumptions about the data-generating process. For example, you might (as we will do in chapter 3) assume that the covariates are independently and identically distributed from some unknown distribution with certain properties, and show that with high probability the approximation resembles the posterior according to some metric. In this way, you can make conclusions about the quality of approximation for a 'typical' dataset, where typical depends on assumptions on the data-generating process.

1.4.3 Did it Work?

While similar to the first question we ask about an approximate method, whether a method has worked on a *particular dataset* is subtly different. Ideally, we might want a method that leads to high-quality approximations and is computationally efficient on *any* dataset. Unfortunately, the author is not aware of such strong a priori guarantees for a method, and if they hold they may be overly pessimistic. We therefore want to be able to certify after an analysis has been run that it has worked on the dataset we have observed. We refer to guarantees of this form as *a posteriori* as they will generally be computed *after running the analysis*.

The phrasing of this question is partially inspired by the title of Yao et al. (2018) "Yes, but did it work?", which developed a tool for performing this type of analysis within the framework of variational inference.

1.4.4 Is it Easy to Use?

This question is somewhat subjective subtle. For nearly every method commonly used for approximate Gaussian process regression, there is a limit in which the method works but is not computationally feasible. Parameters must generally be set to trade off between the quality of an approximation (and guarantees on the quality of the approximation) and the computation used. It is important that these parameters can be easily chosen, ideally in a way that adaptively determines the amount of computation needed to approximate the given inference and model selection task to high fidelity, and without requiring the practitioner to have an intimate understanding of the approximation. Without this level of automation, the method will not be able to be used by most practitioners, and will therefore have limited practical impact.

1.5 A Handful of Approximate Gaussian Process Methods

A range of methods have been suggested for allowing the application of Gaussian process models to large datasets. The list of methods provided here is far from exhaustive. Additionally, we do not attempt an extensive description of all the methods, as surveys and comparisons between them are already available (Chalupka et al., 2013; Liu et al., 2020).

1.5.1 Low-rank Approximations

One of the most ubiquitous method for approximating Gaussian process regression in the machine learning literature involves making some form of low-rank approximation to $\mathbf{K}_{\mathbf{x},\mathbf{x}}$, after which analogues of eq. (1.38) and eq. (1.6) can be computed quickly using Woodbury's lemma and the matrix determinant lemma (proposition A.16 and proposition A.17). A key insight of Gaussian process regression is that we do not need to realize all the features in a feature space in order to perform inference. In contrast, low-rank approximations construct a finite set of features, with cardinality $M \ll N$, that can be used to learn about the data. Once a finite set is selected, we can infer weights for these features instead of directly inferring values of the latent function. This approach raises the questions:

1. How do we choose a finite set of features to learn about the data?
2. How do we learn from the data using these features?

We now discuss two common methods used for defining features. Chapter 2 focuses on the second question within a variational Bayesian framework.

Nyström Approximations

One way to approximate the kernel matrix is to consider features of the form $\{k(z_m, \cdot)\}_{m=1}^M$ for $z_m \in \mathcal{X}$. The kernel function can then be replaced with a rank M kernel, that is obtained by projecting the infinitely many features associated to the original kernel onto $\text{span}\{k(z_m, \cdot)\}_{m=1}^M$. We give a precise description in section 2.3.

Selecting a fixed set of features yields the approximation $\mathbf{K}_{\mathbf{x},\mathbf{x}} \approx \mathbf{Q}_{\mathbf{x},\mathbf{x}}$, where $\mathbf{Q}_{\mathbf{x},\mathbf{x}} = \mathbf{K}_{\mathbf{x},\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^\top$. We refer to the approximation of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ by $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$ as the Nyström approximation (Williams and Seeger, 2001), although we note that this use of the term deviates from its original use within the field of numerical methods. Often these methods are referred to as "sparse" in analogy to sparse linear regression, where many coefficients of a linear model are set to 0, and so only a subset of features is used to model the data.

Various methods using an approximation of this form have been proposed, including Csató and Opper (2002); Seeger et al. (2003); Smola and Bartlett (2001); Snelson and Ghahramani (2006); Williams and Seeger (2001) and Titsias (2009). In Chapter 2, we will discuss the formulation due to Titsias (2009), which builds off of the earlier works Csató and Opper (2002) and Seeger et al. (2003) in motivating this approximation via variational inference, in some degree of detail. For a comparison between other approaches based on a Nyström approximation, see Quiñonero-Candela and Rasmussen (2005).

Random Fourier Feature Approximations

An alternative form of low-rank approximation to the kernel matrix can be derived from Bochner's theorem (eq. 1.34). Drawing M independent samples from the spectral measure and evaluating trigonometric features with these frequencies leads to a Monte Carlo estimate of the kernel, and a rank M approximation to the kernel matrix (Rahimi and Recht, 2007). In the case of Gaussian process regression, this amounts to performing inference in a Bayesian linear regression model with features defined by the sampled frequencies. These frequencies can be treated as fixed, or selected based on the data using maximum marginal likelihood (Lázaro-Gredilla et al., 2010).

Other Structured Matrix Approximations

While low-rank approximations are particularly versatile as a tool for scalable Gaussian process regression, any other approximation to the kernel matrix that admits fast linear algebra can be used. If the data is observed on a one-dimensional grid and the kernel is stationary, $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ is Toeplitz (has constant diagonals). This allows for (exact) matrix inversion in $O(N^2)$ via Trench's algorithm or approximate solutions to linear systems of equations via circulant embedding and fast Fourier transforms even more quickly. Extensions to this approach to other spaces were examined in Storkey (1999). In more than one dimension if the kernel is a product of stationary kernels defined along each dimension and the data is on a grid, the kernel matrix is a Kronecker product of Toeplitz matrices. This observation, along with additional approximations to allow for data that is not exactly on a multidimensional lattice were used to scale Gaussian process methods in several dimensions in Wilson and Nickisch (2015). Other approaches to approximate the kernel matrix with a structured matrix, for example based on matrices with hierarchical structure have been proposed (Ambikasaran et al., 2015; Geoga et al., 2020).

1.5.2 Iterative Approximations

A complementary approach to using matrix-structure to approximate the posterior distribution (eq. 1.6) and the log marginal likelihood (eq. 1.38) relies on producing a sequence of estimates for the desired quantity via an iterative method. These estimates should converge to the quantity if the algorithm is run for a sufficiently long time. Gibbs and MacKay (1997) proposed such a method building on work of Skilling (1993) and based on an application of the method of conjugate gradients (Hestenes and Stiefel, 1952). Chapter 4 discusses variations of this approach.

1.6 Outline and Contributions of this Thesis

The focus of this thesis will the analysis and application of scalable methods for model selection and approximate inference in Gaussian process regression models. We study when a method works and how we can verify this.

- Chapter 2 provides background on sparse variational Gaussian process regression. After giving a probabilistic derivation of the method that follows much of the literature in the area, we focus on a linear-algebraic view of the approximation. We give a novel derivation of the optimal variational parameters derived in Titsias (2009) through the lens of finite-dimensional Bayesian linear regression. We then discuss diagnostics that can be used to confirm that the approximate method has worked. Several of these diagnostics have not previously appeared in the literature. A worked example of checking these diagnostics on a small, one-dimensional dataset is provided.
- Chapter 3 contains an analysis of sparse variational Gaussian process regression. We construct upper and lower bounds on the number of inducing points needed to establish a computational savings (asymptotically) relative to exact inference. We study these bounds to determine under what sets of assumptions about the data and model sparse variational approximations work. These results were previously presented in Burt et al. (2019) and Burt et al. (2020b), and builds on earlier work presented in the Master of Philosophy thesis Burt (2018).
- Chapter 4 begins with background on the application of iterative methods to Gaussian process regression. A novel combination of low-rank and iterative methods is used to provide bounds on the log marginal likelihood. We discuss the utility of these bounds for approximate maximum marginal likelihood model selection. We link the stopping criterion of the iterative method to the tightness of the bound, and show this results in more stable training procedures than commonly used iterative approaches. This method was originally described in Artemev et al. (2021). We then show that the idea of linking stopping criterion to quality of estimation of the log marginal likelihood can be extended to iterative methods without any low-rank approximation. This approach was described in Burt et al. (2021). The methods presented generally have quite few hyperparameters to tune, making them easy-to-use without extensive knowledge of the underlying machinery.
- Chapter 5 provides a retrospective summary of the contributions of this thesis, and highlights several directions for future research.

Chapter 2

Variational Gaussian Process Regression

In this chapter, we introduce several diagnostic tools for determining the accuracy of approximate inference in Gaussian process regression. A theme of this chapter is that there is more structure in Gaussian process regression than a generic Bayesian inference problem, and so we should reasonably expect to be able to say more about the quality of approximate inference than we can generically. We provide an example of the application of diagnostic tools on a simple problem that shows that the tools can be practically applied in some instances, although they are pessimistic leading to the need for additional computation. We additionally discuss limitations of the proposed diagnostic tools, particularly in regards to model selection.

Prior to delving into diagnostic tools, we provide an introduction to the sparse variational approach for scaling Gaussian process regression introduced by Titsias (2009), which builds on earlier work by Csató and Opper (2002) and Seeger et al. (2003). This is among the most generally applicable and easy-to-use approaches to approximate inference and model selection in Gaussian process models with Gaussian likelihoods.

The first three sections of this chapter introduce variational Bayesian inference broadly, and its application to Gaussian process regression, answering:

- What is variational Bayesian inference (section 2.1)?
- What choices are made when applying variational inference to Gaussian process regression (section 2.2)?
- What specific structure is present in variational inference applied to Gaussian process regression (section 2.3)?

These sections are all background material. The reader familiar with the framework of Titsias (2009) can skim the first two sections. The third section takes a linear-algebraic view, that appears to be increasingly common in the approximate Gaussian process literature (see for example Hensman et al. 2018; van der Wilk 2019; Wild and Wynne 2021; Wilson et al. 2020) although it dates back much further (Csató and Opper, 2002). We place a particular emphasis on connections to finite-dimensional linear regression,

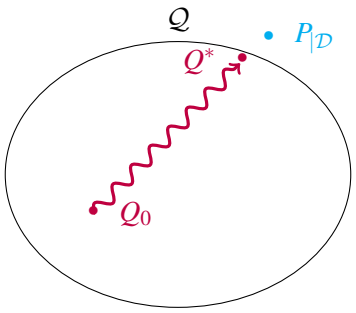


Fig. 2.1 A cartoon depiction of variational inference. The elliptical region represents the variational family considered. An initial approximation to the posterior Q_0 is selected. The divergence between distributions in $\mathcal{Q}$ and the posterior $P_{|\mathcal{D}}$ is minimized (represented by the squiggly curve), leading to the selection of $\mathcal{Q}^*$ as a tractable approximation to the posterior.

and give a derivation of the optimal form of the variational approximation presented in Titsias (2009) from this perspective.

The last three sections focus on other aspects of the approximate Gaussian process workflow (workflow 2). They are guided by the following questions,

- How can we check if variational Gaussian process regression has worked (section 2.4)?
- How can we perform model selection with variational Gaussian process regression (section 2.5)?
- What are the computational and numerical properties of workflow 2 using sparse variational Gaussian process regression (section 2.6)?

Scalable diagnostics for Gaussian process regression have been studied previously in Davies (2015); Titsias (2014) and Huggins et al. (2019). Diagnostics for variational inference with more general Bayesian models were considered in Yao et al. (2018) and Huggins et al. (2020). We derive several new bounds on posterior moments, as well as a refinement of an upper bound on the Kullback-Leibler divergence to the posterior, that can be used as diagnostic tools to assess the quality of inference (see corollary 2.10, proposition 2.15, proposition 2.13, proposition 2.18). Additionally, we provide a worked example showing how the diagnostics can be used to assess inference, which appears to be largely missing from the Gaussian process regression literature. We explore some difficulties of diagnosing model selection with sparse Gaussian process regression previously noted in Kim and Teh (2018). Understanding approximate maximum marginal likelihood in these models appears to be significantly more difficult task than assessing the quality of inference. Finally, we discuss details of the implementation of variational inference in Gaussian process regression, highlighting issues of numerical stability that are well-known, though not often discussed in the literature.

2.1 Variational Bayesian Inference

The goal of variational inference is to approximate the posterior distribution with a similar distribution that is easier to sample from and evaluate expectations under. Central to variational inference is the

notion of a *divergence* which generalizes the notion of a metric and is used to measure how similar probability measures are. We say that $\mathfrak{D} : \mathcal{M}_1 \times \mathcal{M}_1 \to \mathbb{R} \cup \{\infty\}$ is a divergence if[1]

$$\mathfrak{D}(P,Q) \geq 0 \qquad \text{and} \qquad \mathfrak{D}(P,Q) = 0 \Leftrightarrow P = Q. \tag{2.1}$$

Here $\mathcal{M}_1$ denotes the space of probability measures on some measurable space. Variational inference can be summarized in three steps:

1. Define a *variational family*, $\mathcal{Q}$: a set of probability distributions that allow for easy computation of quantities of interest. Ideally, there is at least one element of $\mathcal{Q}$ that captures salient properties of the posterior $P_{|\mathcal{D}}$.
2. Select a notion of divergence, $\mathfrak{D}$, between distributions. This is used to measure how close a candidate posterior is to the true posterior.
3. Select the distribution in the variational family that best approximates the posterior by minimizing the divergence chosen in step 2. In other words, the approximate posterior is

$$Q^* \in \arg\min_{Q \in \mathcal{Q}} \mathfrak{D}(Q, P_{|\mathcal{D}}). \tag{2.2}$$

This process is illustrated in figure 2.1. A critical aspect of variational inference is the choice of divergence, $\mathfrak{D}$. There are two considerations to bear in mind when selecting $\mathfrak{D}$. First, $\mathfrak{D}$ should encode the correct sense of closeness, in the sense that distributions that are close according to the divergence should lead to similar statistical conclusions. Second, the minimization problem in eq. (2.2) must be tractable or able to be approximated. In the Gaussian process literature, several choices of divergence have been proposed. Titsias (2009) suggested minimizing the reverse Kullback-Leibler divergence, which is by far the most common approach taken in variational Bayesian inference as it allows for the most tractable computation. Bui et al. (2017) motivated performing expectation propagation or power expectation propagation as attempting to minimize an α-divergence via a local approximation. Huggins et al. (2019) suggested using a divergence that upper bounds a Wasserstein divergence to obtain better control on the approximation of posterior moments and showed it could be locally minimized via gradient descent.

We focus on the formulation in terms of Kullback-Leibler divergence. The Kullback-Leibler (KL) divergence is defined by

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) := \int \log \frac{q(f)}{p_{|\mathcal{D}}(f)} dQ \tag{2.3}$$

where q and $p_{|\mathcal{D}}$ are densities of Q and $P_{|\mathcal{D}}$ with respect to an arbitrary measure such that they are well-defined. The Kullback-Leibler divergence has several useful properties.

[1] It is often required that a divergence behaves like a quadratic form locally. We do not require this, but the Kullback-Leibler divergence satisfies this criterion.

Proposition 2.1 (Chain rule of Kullback-Leibler Divergence, Polyanskiy and Wu, 2014, Theorem 2.2)**.** *Let (X,Σ_X) and (Y,Σ_Y) measurable spaces. Let P,Q be probability measures, on $(X\times Y,\Sigma_X\times\Sigma_Y)$, P_X,Q_X the projection of the measures onto X and $P_{Y|X},Q_{Y|X}$ the conditional measures. Then,*

$$\mathfrak{D}_{KL}(Q,P)=\mathfrak{D}_{KL}(Q_X,P_X)+\mathbb{E}_{X\sim Q_X}[\mathfrak{D}_{KL}\left(Q_{Y|X},P_{Y|X}\right)]. \tag{2.4}$$

Proposition 2.1 allows us to relate the Kullback-Leibler divergence between measures on a joint space into the Kullback-Leibler divergence between the marginal and conditional measures.

The *data-processing inequality* says that if the same transformation is applied to two random variables, the distributions of the resulting random variables cannot be farther apart than the distributions of the original random variables. This can be derived by applying the chain rule of Kullback-Leibler divergence with both orderings of the random variables.

Corollary 2.2 (Data Processing Inequality)**.** *With the same setup as proposition 2.1 and letting P_Y,Q_Y denote projection onto (Y,Σ_Y), if $Q_{Y|X}=P_{Y|X}$, Q_X–almost surely, then*

$$\mathfrak{D}_{KL}(Q_Y,P_Y)\leq\mathfrak{D}_{KL}(Q_X,P_X). \tag{2.5}$$

While these properties are useful, the primary motivation for using the Kullback-Leibler divergence is that minimization of $\mathfrak{D}_{\mathrm{KL}}\left(Q,P_{|\mathcal{D}}\right)$ can be performed without access to the posterior density or samples from the posterior. This is shown by the following manipulation

$$\begin{aligned}\mathfrak{D}_{\mathrm{KL}}\left(Q,P_{|\mathcal{D}}\right)&=\int\log\frac{q(f)}{p_{|\mathcal{D}}(f)}dQ &(2.6)\\ &=\int\log\frac{q(f)}{p(f)}dQ+\int\log\frac{p(f)}{p_{|\mathcal{D}}(f)}dQ &(2.7)\\ &=\mathfrak{D}_{\mathrm{KL}}(Q,P)-\int\log\frac{p_{|\mathcal{D}}(f)}{p(f)}dQ &(2.8)\\ &=\mathfrak{D}_{\mathrm{KL}}(Q,P)+\mathcal{L}(\theta)-\int\log\ell_{\mathcal{D}}dQ, &(2.9)\end{aligned}$$

where $\ell_{\mathcal{D}}=\ell_{\mathcal{D},\theta}$ denotes the likelihood function of the data[2] and $\mathcal{L}(\theta)=\log\int\ell_{\mathcal{D},\theta}dP_\theta$ is the marginal likelihood from eq. (1.38). The key observation in this calculation is Bayes' rule (eq. 1.4), which allows us to go from eq. (2.8) to eq. (2.9).

Taking a minimum on both sides with respect to $Q\in\mathcal{Q}$,

$$\arg\min_{Q\in\mathcal{Q}}\mathfrak{D}_{\mathrm{KL}}\left(Q,P_{|\mathcal{D}}\right)=\arg\max_{Q\in\mathcal{Q}}\underbrace{\int\log\ell_{\mathcal{D}}dQ-\mathfrak{D}_{\mathrm{KL}}(Q,P)}_{:=\underline{\mathcal{L}}(Q,\theta)}. \tag{2.10}$$

[2]Commonly this is written, $\ell_{\mathcal{D},\theta}(f)=\log p(\mathbf{y}|f,\mathbf{x},\theta)$, and for model eq. (1.3) is given in eq. (1.5).

Commonly $\underline{\mathcal{L}}(Q,\theta)$ is referred to as the *evidence lower bound* (ELBO), as it lower bounds $\mathcal{L}(\theta)$.[3] If the prior density can be evaluated, the density of Q can be evaluated, and Q can be sampled from, $\underline{\mathcal{L}}(Q,\theta)$ can be estimated unbiasedly with simple Monte Carlo methods. This allows us to select Q by performing stochastic gradient ascent to approximate the maximization eq. (2.10). In some cases, including when the prior, likelihood and approximate posterior are Gaussian $\underline{\mathcal{L}}(Q,\theta)$ is analytically tractable, further facilitating approximate solutions to the maximization problem posed in eq. (2.10).

2.1.1 Variational Characterization of the Bayes' Posterior

As the Kullback-Leibler divergence between two measures is non-negative and equal to zero if and only if the measures are the same

$$P_{|\mathcal{D}} = \arg\min_{Q\in\mathcal{M}_1} \mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) = \arg\max_{Q\in\mathcal{M}_1} \underline{\mathcal{L}}(Q,\theta), \tag{2.11}$$

where $\mathcal{M}_1$ denotes the space of all probability measures on the same measurable spaces on which the prior is defined. This yields a variational interpretation of the posterior distribution as the maximizer of eq. (2.10).

2.2 The Variational Family for Sparse Gaussian Process Regression

We return to the specific problem of Gaussian process regression. To instantiate variational inference with the Kullback-Leibler divergence, we need to specify the variational family, $\mathcal{Q}$, and a method for (approximately) optimizing over $\mathcal{Q}$. In this section, we state the variational family from a largely probabilistic perspective, roughly following the style of derivation given in Hensman et al. (2013, 2015b); Matthews et al. (2016).

2.2.1 The Variational Family

Motivated by earlier successes of Nyström based methods, Titsias (2009) defined a set of points $\mathbf{z} \subset \mathcal{X}$, $|\mathbf{z}| = M$ that are used to form the approximation. Let $\mathbf{x}^* \subset \mathcal{X}, |\mathbf{x}^*| < \infty$ be arbitrary, and let q denote the density of the variational posterior with respect to Lebesgue measure of the appropriate dimension. Then

$$q(f_{\mathbf{z}}, f_{\mathbf{x}}, f_{\mathbf{x}^*}) = q(f_{\mathbf{z}})q(f_{\mathbf{x}}|f_{\mathbf{z}})q(f_{\mathbf{x}^*}|f_{\mathbf{x}}, f_{\mathbf{z}}), \tag{2.12}$$

and

$$p_{|\mathcal{D}}(f_{\mathbf{z}}, f_{\mathbf{x}}, f_{\mathbf{x}^*}) = p_{|\mathcal{D}}(f_{\mathbf{z}})p_{|\mathcal{D}}(f_{\mathbf{x}}|f_{\mathbf{z}})p_{|\mathcal{D}}(f_{\mathbf{x}^*}|f_{\mathbf{x}}, f_{\mathbf{z}}). \tag{2.13}$$

[3] To expand on this naming choice, the log marginal likelihood is sometimes referred to as the *model evidence*.

Conditioned on the latent function values at $f(x_1), \ldots, f(x_n)$ the data contains no information about the underlying process (see eq. 1.3). Therefore,

$$p_{|\mathcal{D}}(f_{\mathbf{x}^*}|f_{\mathbf{x}}, f_{\mathbf{z}}) = p(f_{\mathbf{x}^*}|f_{\mathbf{x}}, f_{\mathbf{z}}) \tag{2.14}$$

where p is the prior conditional density. Using the chain rule of Kullback-Leibler divergence (proposition 2.1) and eq. (2.14)

$$\begin{aligned}\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) &= \mathfrak{D}_{\mathrm{KL}}\left(Q_{f_{\mathbf{x}}, f_{\mathbf{z}}}, P_{f_{\mathbf{x}}, f_{\mathbf{z}}|\mathcal{D}}\right) + \mathbb{E}_{f_{\mathbf{x}}, f_{\mathbf{z}} \sim Q_{f_{\mathbf{x}}, f_{\mathbf{z}}}}\left[\mathfrak{D}_{\mathrm{KL}}\left(Q_{|f_{\mathbf{x}}, f_{\mathbf{z}}}, P_{|f_{\mathbf{x}}, f_{\mathbf{z}}, \mathcal{D}}\right)\right] &(2.15)\\ &= \mathfrak{D}_{\mathrm{KL}}\left(Q_{f_{\mathbf{x}}, f_{\mathbf{z}}}, P_{f_{\mathbf{x}}, f_{\mathbf{z}}|\mathcal{D}}\right) + \mathbb{E}_{f_{\mathbf{x}}, f_{\mathbf{z}} \sim Q_{f_{\mathbf{x}}, f_{\mathbf{z}}}}\left[\mathfrak{D}_{\mathrm{KL}}\left(Q_{|f_{\mathbf{x}}, f_{\mathbf{z}}}, P_{|f_{\mathbf{x}}, f_{\mathbf{z}}}\right)\right]. &(2.16)\end{aligned}$$

The second term is non-negative, and equals zero if and only if $Q_{|f_{\mathbf{x}}, f_{\mathbf{z}}} = P_{|f_{\mathbf{x}}, f_{\mathbf{z}}}$ $Q_{f_{\mathbf{x}}, f_{\mathbf{z}}}$–almost surely. Hence, without any loss of generalization, we may assume $\mathcal{Q}$ only contains distributions satisfying $Q_{|f_{\mathbf{x}}, f_{\mathbf{z}}} = P_{|f_{\mathbf{x}}, f_{\mathbf{z}}}$, as if not minimization of the Kullback-Leibler divergence would prefer a distribution of this form by eq. (2.16).

It remains to specify candidate distributions for $Q_{f_{\mathbf{z}}}$ and $Q_{f_{\mathbf{x}}|f_{\mathbf{z}}}$. In order to make computations fast, a further restriction is imposed that

$$Q_{f_{\mathbf{x}}|f_{\mathbf{z}}} = P_{f_{\mathbf{x}}|f_{\mathbf{z}}}. \tag{2.17}$$

This means that *conditioned on the values of the latent process at the points* $\mathbf{z}$ *the approximate posterior is equal to the prior conditioned on the same values*. This is not true of the posterior, with the notable exception when $\mathbf{x} \subset \mathbf{z}$. In most other cases, this means $P_{|\mathcal{D}} \notin \mathcal{Q}$, and variational inference incurs some error. We can think of $f_{\mathbf{z}}$ as acting like a sufficient statistic for the observed data, and so if most of the information that the model would extract from the observations $\mathbf{y}$ can be summarized by knowing a distribution over $f_{\mathbf{z}}$, then the error introduced by this approximation is small.

Finally, $Q_{f_{\mathbf{z}}}$ is parameterized as a free-form Gaussian distribution, $Q_{f_{\mathbf{z}}} = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$. As all the conditional and marginal measures are Gaussian, the measure Q is a Gaussian measure. Q can therefore be characterized by its mean and covariance functions

$$\mu_Q(x) = k_{x\mathbf{z}} K_{\mathbf{zz}}^{-1} \boldsymbol{\mu}_{\mathbf{z}}, \tag{2.18}$$

$$k_Q(x, x') = k(x, x') - \mathbf{k}_{x\mathbf{z}} \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{z}x'} + \mathbf{k}_{x\mathbf{z}} \mathbf{K}_{\mathbf{zz}}^{-1} \boldsymbol{\Sigma}_{\mathbf{z}} \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{z}x'}. \tag{2.19}$$

The variational family is constructed by allowing $\mathbf{z}$ to range over $\mathcal{X}^M$, $\mu_{\mathbf{z}}$ to range over $\mathbb{R}^M$ and $\Sigma_{\mathbf{z}}$ to range over the space of positive definite matrices $S_{++}^M \subset \mathbb{R}^{M \times M}$.

2.2.2 The Evidence Lower Bound

We now consider the evidence lower bound arising from the variational family described in eqs. (2.18) and (2.19). We essentially follow the approach laid out in Matthews et al. (2016, Section 3). From

eq. (2.8),

$$\mathcal{L}(\theta) - \mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) = \int \log \ell_{\mathcal{D}} dQ - \mathfrak{D}_{\mathrm{KL}}\left(Q, P\right). \tag{2.20}$$

Because, $Q_{|f_{\mathbf{z}}} = P_{|f_{\mathbf{z}}}$ and using the chain rule for Kullback-Leibler divergence (proposition 2.1)

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P\right) = \mathfrak{D}_{\mathrm{KL}}\left(Q_{f_{\mathbf{z}}}, P_{f_{\mathbf{z}}}\right), \tag{2.21}$$

where $Q_{f_{\mathbf{z}}} = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{z}}, \boldsymbol{\Sigma}_{\mathbf{z}})$ and $P_{f_{\mathbf{z}}} = \mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{z},\mathbf{z}})$. This is a Kullback-Leibler divergence between M-dimensional Gaussian distributions and can be computed in $O(M^3)$.

Turning to the first term in eq. (2.20) and since the observed response variables only depend on the latent function through it values at the corresponding covariates, i.e. $(y_n|f) \overset{\mathrm{d}}{=} (y_n|f(x_n))$, there exists an ℓ such that

$$\int \log \ell_{\mathcal{D}} dQ = \sum_{n=1}^{N} \mathbb{E}_{f(x_n)\sim Q_{f_{x_n}}}[\ell(y_n, f(x_n))]. \tag{2.22}$$

In the Gaussian likelihood case that is the focus of this work (eq. 1.3),

$$\ell(y_n, f(x_n)) = -\frac{1}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\left(y_n - f(x_n)\right)^2, \tag{2.23}$$

see also eq. (1.5). The expectation of eq. (2.23) can be written in terms of the first and second moment of $f(x_n)$ under Q by expanding the quadratic form. As we can compute the first and second moment of $f(x_n)$ under Q with eq. (2.18) and eq. (2.19) with a cost of $O(M^3)$ for the first x_n and a cost of $O(M^2)$ for subsequent points, eq. (2.22) can be calculated in $O(NM^2 + M^3)$. Alternatively, the entire sum can be estimated via a simple Monte Carlo estimator formed by selecting a subset of the datapoints (Hensman et al., 2013), allowing for mini-batch updates to the variational parameters and hyperparameters in time $O(\tilde{N}M^2 + M^3)$, where $\tilde{N} < N$ is the size of a mini-batch.

Combining eq. (2.21) and eq. (2.22), eq. (2.20) becomes

$$\underline{\mathcal{L}}(Q, \theta) = \sum_{n=1}^{N} \mathbb{E}_{f(x_n)\sim Q}[\ell(y_n, f(x_n))] - \mathfrak{D}_{\mathrm{KL}}\left(Q_{f_{\mathbf{z}}}, P_{f_{\mathbf{z}}}\right). \tag{2.24}$$

Equation (2.24) can be evaluated in $O(NM^2 + M^3)$ or estimated in an unbiased fashion with Monte Carlo in $O(\tilde{N}M^2 + M^3)$.

2.3 Linear Algebra and Variational Inference in Gaussian Process Regression

One of the key insights of Titsias (2009) is that the optimization problem,

$$\max_{Q_{f_{\mathbf{z}}} \in \mathcal{M}_1} \underline{\mathcal{L}}(Q, \theta), \tag{2.25}$$

where $\mathcal{M}_1$ denotes the space of all probability measures on $\mathbb{R}^M$, can be solved analytically, leading to a "collapsed" form of the variational objective in which all the variational parameters (except for $\mathbf{z}$) can be computed analytically. Before re-deriving this result, we discuss briefly linear algebraic properties of the variational approximation, taking a similar view as Csató and Opper (2002). Once we have developed the necessary linear algebra, we find that both exact and variational Gaussian process regression can be derived by first decomposing the associated vector space into two parts, performing (finite) Bayesian linear regression in one of these parts, and leaving the other unchanged. Wild et al. (2021) takes a similar perspective on sparse variational Gaussian process regression, emphasizing connections to the kernel ridge regression literature through the same linear algebraic formalism.

2.3.1 Constructing a Hilbert Space

We follow the construction in Wahba (1990, Section 1.4). Let $F = \{f_x : x \in \mathcal{X}\}$ denote the collection of Gaussian random variables indexed by $\mathcal{X}$, with distributions given by the prior. We use the notation f_x and $f(x)$ interchangeably in the sequel, preferring f_x when we want to highlight vector space structure. F is not closed under linear combinations.[4]

To give F the structure of a vector space over $\mathbb{R}$ take

$$V := \operatorname{span} F = \left\{ \sum_{i=1}^{I} \alpha_i f_{x_i} : I \in \mathbb{N},\, f_{x_i} \in F,\, \alpha_i \in \mathbb{R} \right\}. \tag{2.26}$$

By construction V contains 0 and is closed under addition and scalar multiplication. Therefore, V is a vector space. Additionally, the prior carries with it a non-negative bilinear form defined on F by

$$\langle f_x, f_{x'} \rangle = \mathbb{E}[f_x f_{x'}] = k(x, x') \qquad \forall x, x' \in \mathcal{X}, \tag{2.27}$$

that can be extended linearly to V.

This is a non-negative bilinear form, and an inner product if k is (strictly) positive definite. By defining equivalence classes $f \sim g \Leftrightarrow \langle f - g, f - g \rangle = 0$, $V/\sim$ has the structure of an inner product space. Moving forward, we abuse notation by conflating functions and their equivalence classes under

[4] For example, for a stationary kernel that is not identically 0 and for any $x \in \mathcal{X}$ there is no x' such that $2f(x) = f(x')$ unless k is the 0 kernel. Suppose such an x' existed. The variance of $f(x')$ is equal to $k(x', x')$ by definition and equal to $4k(x, x)$ by the supposition. This implies $k(x', x') = 4k(x, x)$, which contradicts that k is stationary and not 0. Therefore, $2f(x) \notin F$.

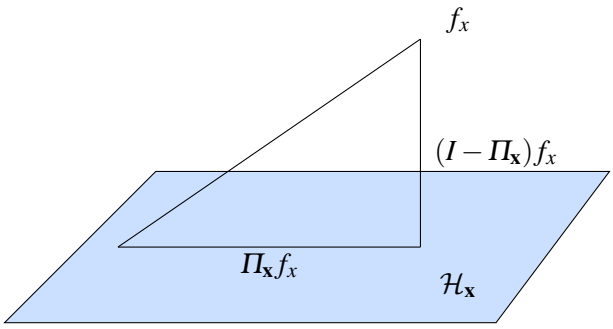


Fig. 2.2 A schematic showing the orthogonal decomposition $\mathcal{H} = \mathcal{H}_{\mathbf{x}} \oplus \mathcal{H}_{\mathbf{x}}^{\perp}$. For an arbitrary vector f_x, $\Pi_{\mathbf{x}} f_x$ is the part of f_x determined by the observed data, while $(I-\Pi_{\mathbf{x}})f_x$ is independent of the observed data.

$\sim$. Taking the completion of $V/\sim$, results in a Hilbert space $\mathcal{H} \subset L^2(\Omega)$, where Ω is the sample space over which the process is defined.

The map,

$$f_x \to k(x,\cdot) \tag{2.28}$$

is an isomorphism from $\mathcal{H}$ to the *reproducing kernel Hilbert space* associated to the kernel k, commonly referred to as Loève's isometry (Berlinet and Thomas-Agnan, 2011, Theorem 35). As a result, the same intuition can be formulated in the language of reproducing kernel Hilbert spaces. We refer the interested reader to Kanagawa et al. (2018) and Wild and Wynne (2021) for recent expositions discussing connections between the Gaussian process and reproducing kernel Hilbert space perspectives.

2.3.2 Gaussian Process Regression as (Almost) Finite Dimensional Linear Regression

Intuitively, Gaussian process regression is often motivated as Bayesian linear regression with the number and form of the features depending on the data. Indeed, if we consider the linear regression model

$$y_n = f_{\boldsymbol{\alpha}}(x_n) + \varepsilon_n,\ f_{\boldsymbol{\alpha}}(x) = \sum_{n=1}^{N} \alpha_n k(x_n, x),\ \varepsilon_n \sim \mathcal{N}(0,\sigma^2),\ \boldsymbol{\alpha} \sim \mathcal{N}(0, \mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}) \tag{2.29}$$

the posterior predictive is Gaussian with predictive mean and covariance

$$\hat{\mu}_{\text{Lin}}(x) = \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{y} \qquad \hat{k}_{\text{Lin}}(x,x') = \mathbf{k}_{x\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}\mathbf{k}_{\mathbf{x}x'} - \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x}x'}, \tag{2.30}$$

(see Rasmussen and Williams 2006, Chapter 2). Comparing eq. (1.6) and eq. (2.30), we see

$$\hat{\mu}_{\text{Lin}} = \hat{\mu} \quad \text{and} \quad \hat{k}_{\text{Lin}}(x,x') = \hat{k}(x,x') - \mathbf{k}_{x\mathbf{x}}\mathbf{K}_{\mathbf{x},\mathbf{x}}^{-1}\mathbf{k}_{\mathbf{x}x'}. \tag{2.31}$$

For any $\mathbf{a} \subset \mathcal{X}$, let $\mathcal{H}_{\mathbf{a}} := \text{span}\,\{f_a : a \in \mathbf{a}\}$ and $\Pi_{\mathbf{a}}$ denote orthogonal projection onto $\mathcal{H}_{\mathbf{a}}$. We have the orthogonal decomposition

$$\mathcal{H} = \mathcal{H}_{\mathbf{x}} \oplus \mathcal{H}_{\mathbf{x}}^{\perp}, \tag{2.32}$$

meaning that any $f \in \mathcal{H}$ can be written as $f = f_{\parallel} + f_{\perp}$, with $f_{\parallel} \in \mathcal{H}_{\mathbf{x}}$ and $f_{\perp}$ orthogonal to every $g \in \mathcal{H}_{\mathbf{x}}$. This decomposition is illustrated in figure 2.2.

$\mathcal{H}_{\mathbf{x}}^{\perp}$ denotes the subspace of all the random variables that are orthogonal (uncorrelated, hence independent) from all the $f(x_n)$. Since these random variables are independent of each $f(x_n)$ they are also independent of y_n for $1 \leq n \leq N$. Hence, this part of the space is totally uninformed by the observed data and will remain at the prior. We define the operator $\Pi_{\mathbf{x}} : \mathcal{H} \to \mathcal{H}$ as orthogonal projection onto $\mathcal{H}_{\mathbf{x}}$. We can then re-express the difference between the covariance of the Gaussian process and the covariance of the finite dimensional model from eq. (2.31) as

$$\hat{k}(x,x') - \hat{k}_{\text{Lin}}(x,x') = \langle (I - \Pi_{\mathbf{x}}) f_x, (I - \Pi_{\mathbf{x}}) f_{x'} \rangle_{\mathcal{H}}. \tag{2.33}$$

In the case $x = x'$, this simplifies to

$$\hat{k}(x,x) - \hat{k}_{\text{Lin}}(x,x) = \|(I - \Pi_{\mathbf{x}}) f_x\|_{\mathcal{H}}^2. \tag{2.34}$$

In words, the difference between the posterior predictive variance of the finite feature model and the Gaussian process at a point x is the squared distance from f_x to $\mathcal{H}_{\mathbf{x}}$, the part of the process that is informed by the data. The notion of distance used comes from the prior.

In the next section, we will show that it is not a coincidence that the posterior of the Gaussian process and finite linear regression agree on the space $\mathcal{H}_{\mathbf{x}}$ and that this can be seen from the variational characterization of the posterior (section 2.1.1).

2.3.3 Variational Gaussian Process Regression as Modified Linear Regression

We return to the problem of computing the optimal $Q_{f_{\mathbf{z}}}$, through the lens of finite dimensional Bayesian linear regression.

Write $\mathcal{H} = \mathcal{H}_{\mathbf{z}} \oplus \mathcal{H}_{\mathbf{z}}^{\perp}$. Then

$$k(x,x') - \mathbf{k}_{x\mathbf{z}} \mathbf{K}_{\mathbf{zz}}^{-1} \mathbf{k}_{\mathbf{z}x'} = \langle (I - \Pi_{\mathbf{z}}) f_x, (I - \Pi_{\mathbf{z}}) f_{x'} \rangle. \tag{2.35}$$

As $\mathcal{H}_{\mathbf{z}}^{\perp}$ is not informed by the $f_{\mathbf{z}}$ and by the assumption placed on Q eq. (2.17), the part of the process lying in $\mathcal{H}_{\mathbf{z}}^{\perp}$ must be equal to the prior. What we show in the remainder of this section is that the *optimal* variational distribution for fixed $\mathbf{z}$ can be computed by the following steps:

1. Perform Bayesian linear regression in $\mathcal{H}_{\mathbf{z}}$ (i.e. using features that span $\mathcal{H}_{\mathbf{z}}$).
2. Add the result to the prior on $\mathcal{H}_{\mathbf{z}}^{\perp}$.

We consider an M-dimensional linear regression model

$$y = f_{\boldsymbol{\alpha}}(x_n) + \varepsilon_n, \quad f_{\boldsymbol{\alpha}}(x) = \sum_{m=1}^{M} \alpha_m k(z_m, x), \quad \varepsilon_n \sim \mathcal{N}(0, \sigma^2), \quad \boldsymbol{\alpha} \sim \mathcal{N}(0, \mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}). \tag{2.36}$$

Let $P^{\text{Lin}}_{\boldsymbol{\alpha}|\mathcal{D}}$ denote the posterior distribution over of this linear model after observing the data, and $P^{\text{Lin}}_{\boldsymbol{\alpha}}$ denote the prior and $\mathcal{M}_1$ denote the space of probability measures on $\mathbb{R}^M$. Using the variational characterization of the posterior (section 2.1.1) and eq. (2.9),

$$P^{\text{Lin}}_{\boldsymbol{\alpha}|\mathcal{D}} = \arg\min_{Q_{\boldsymbol{\alpha}} \in \mathcal{M}_1} \mathfrak{D}_{\text{KL}}\left(Q_{\boldsymbol{\alpha}}, P^{\text{Lin}}_{\boldsymbol{\alpha}|\mathcal{D}}\right) \tag{2.37}$$

$$= \arg\max_{Q_{\boldsymbol{\alpha}} \in \mathcal{M}_1} \left(\sum_{n=1}^{N} \mathbb{E}[\ell(y_n, f_{\boldsymbol{\alpha}}(x_n))] - \mathfrak{D}_{\text{KL}}\left(Q_{\boldsymbol{\alpha}}, P^{\text{Lin}}_{\boldsymbol{\alpha}}\right) \right). \tag{2.38}$$

As the likelihood is Gaussian and because conditional distributions of Gaussian distributions are again Gaussian, we can restrict the optimization to only consider Gaussian measures over the weights

$$P^{\text{Lin}}_{\boldsymbol{\alpha}|\mathcal{D}} = \arg\max_{Q_{\boldsymbol{\alpha}} \in \mathcal{N}} \left(\sum_{n=1}^{N} \mathbb{E}[\ell(y_n, f_{\boldsymbol{\alpha}}(x_n))] - \mathfrak{D}_{\text{KL}}\left(Q_{\boldsymbol{\alpha}}, P_{\boldsymbol{\alpha}}\right) \right), \tag{2.39}$$

where $\mathcal{N}$ denotes the space of all Gaussian distributions on $\mathbb{R}^M$. If we assume the $k(z_m, \cdot)$ are linearly independent, then the map from the features weights to function values $\boldsymbol{\alpha} \to \{f_{\boldsymbol{\alpha}}(z_m)\}_{m=1}^M$ is a bijection on $\mathbb{R}^M$ which allows us to rewrite the optimization,

$$P^{\text{Lin}}_{f_{\mathbf{z}}|\mathcal{D}} = \arg\max_{Q_{f_{\mathbf{z}}} \in \mathcal{N}} \left(\sum_{n=1}^{N} \mathbb{E}_{f \sim Q^{\text{Lin}}_{f_{\mathbf{z}}}}[\ell(y_n, f(x_n))] - \mathfrak{D}_{\text{KL}}\left(Q_{f_{\mathbf{z}}}, P_{f_{\mathbf{z}}}\right) \right). \tag{2.40}$$

In eq. (2.40) $P^{\text{Lin}}_{f_{\mathbf{z}}|\mathcal{D}}$ is the posterior distribution of the latent function values at the points $\{z_m\}_{m=1}^M$ and $Q^{\text{Lin}}_{f_{\mathbf{z}}}$ is the measure of the stochastic process determined by the linear model which has distribution $Q_{f_{\mathbf{z}}}$ at $\{z_m\}_{m=1}^M$.

Any Gaussian distribution over weights in the linear model induces a Gaussian distribution over predictions with the same mean as a sparse Gaussian process approximation via the previously mentioned bijection and the difference between the predictive variances is given by eq. (2.35). Let $Q^{\text{GP}}_{f_{\mathbf{z}}}$ denote the sparse Gaussian process posterior determined by $Q_{f_{\mathbf{z}}}$. Define,

$$t(x_n) \coloneqq k(x_n, x_n) - \mathbf{k}_{x_n \mathbf{z}} \mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1} \mathbf{k}_{\mathbf{z} x_n}, \tag{2.41}$$

$$\sigma^2_{Q^{\text{Lin}}_{f_{\mathbf{z}}}}(x_n) \coloneqq \mathbb{E}_{f \sim Q^{\text{Lin}}_{f_{\mathbf{z}}}}[(f(x_n) - \mathbb{E}_{f \sim Q^{\text{Lin}}_{f_{\mathbf{z}}}}[f(x_n)])^2] \tag{2.42}$$

$$\sigma^2_{Q^{\text{GP}}_{f_{\mathbf{z}}}}(x_n) \coloneqq \mathbb{E}_{f \sim Q^{\text{GP}}_{f_{\mathbf{z}}}}[(f(x_n) - \mathbb{E}_{f \sim Q^{\text{GP}}_{f_{\mathbf{z}}}}[f(x_n)])^2] = \underbrace{t(x_n) + \sigma^2_{Q^{\text{Lin}}_{f_{\mathbf{z}}}}(x_n)}_{\text{c.f. eq. (2.35)}}. \tag{2.43}$$

Expanding eq. (2.40) using the form of ℓ (eq. 2.23) and letting $C = -\frac{N}{2}\log 2\pi\sigma^2$

$$\arg\max_{Q_{f_\mathbf{z}} \in \mathcal{N}} C - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbb{E}_{f\sim Q^{\text{Lin}}_{f_\mathbf{z}}}[f(x_n)])^2 - \sigma^2_{Q^{\text{Lin}}_{f_\mathbf{z}}}(x_n) - \mathfrak{D}_{\text{KL}}\left(Q_{f_\mathbf{z}}, P_{f_\mathbf{z}}\right) \tag{2.44}$$

$$= \arg\max_{Q_{f_\mathbf{z}} \in \mathcal{N}} C - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbb{E}_{f\sim Q^{\text{Lin}}_{f_\mathbf{z}}}[f(x_n)])^2 - \sigma^2_{Q^{\text{GP}}_{f_\mathbf{z}}}(x_n) - t(x_n) - \mathfrak{D}_{\text{KL}}\left(Q_{f_\mathbf{z}}, P_{f_\mathbf{z}}\right) \tag{2.45}$$

$$= \arg\max_{Q_{f_\mathbf{z}} \in \mathcal{N}} C - \frac{1}{2\sigma^2}\sum_{n=1}^{N}(y_n - \mathbb{E}_{f\sim Q^{\text{GP}}_{f_\mathbf{z}}}[f(x_n)])^2 - \sigma^2_{Q^{\text{GP}}_{f_\mathbf{z}}}(x_n) - \mathfrak{D}_{\text{KL}}\left(Q_{f_\mathbf{z}}, P_{f_\mathbf{z}}\right) \tag{2.46}$$

$$= \arg\max_{Q_{f_\mathbf{z}}} \sum_{n=1}^{N} \mathbb{E}_{f\sim Q^{\text{GP}}_{f_\mathbf{z}}}[\ell(y_n, f(x_n))] - \mathfrak{D}_{\text{KL}}\left(Q_{f_\mathbf{z}}, P_{f_\mathbf{z}}\right). \tag{2.47}$$

The final equality uses that $t(x_n)$ is independent of the choice of $Q_{f_\mathbf{z}}$, and that the predictive mean of the Gaussian process and M-dimensional models coincide (eq. 2.31). This is *precisely* the optimization problem over $\{\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z}\}$ posed in eq. (2.24) in the case of Gaussian likelihood, from which we conclude that the optimal variational distribution $Q^*_{f_\mathbf{z}}$ for sparse variational Gaussian process regression with fixed $\mathbf{z}$ is the same as the posterior distribution of the linear regression problem considered, and the variational posterior at points not in $\mathbf{z}$ differs from the finite linear regression model by the term given in eq. (2.35).

Taking all of these observations together we can write down the collapsed predictive posterior (after plugging in the optimal form of variational posterior), as

$$\mu_{Q^*}(x) = \mathbf{q}_{x\mathbf{x}}\mathbf{Q}^{-1}\mathbf{y} \quad \text{and} \quad k_{Q^*}(x,x') = k(x,x') - \mathbf{q}_{x\mathbf{x}}\mathbf{Q}^{-1}\mathbf{q}_{\mathbf{x}x'} \tag{2.48}$$

where

$$(\mathbf{q}_{x\mathbf{x}})_n = \langle \Pi_\mathbf{z} f_x, \Pi_\mathbf{z} f_{x_n}\rangle_\mathcal{H} = (\mathbf{k}_{x,\mathbf{z}}\mathbf{K}^{-1}_{\mathbf{z},\mathbf{z}}\mathbf{K}^\top_{\mathbf{x},\mathbf{z}})_n \tag{2.49}$$

$$(\mathbf{Q}_{\mathbf{x},\mathbf{x}})_{nn'} = \langle \Pi_\mathbf{z} f_{x'_n}, \Pi_\mathbf{z} f_{x_n}\rangle_\mathcal{H} = (\mathbf{K}_{\mathbf{x},\mathbf{z}}\mathbf{K}^{-1}_{\mathbf{z},\mathbf{z}}\mathbf{K}^\top_{\mathbf{x},\mathbf{z}})_{nn'} \tag{2.50}$$

and $\mathbf{Q} = \mathbf{Q}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}$. This has the same form as the posterior moments (eq. 1.6), but replacing inner products computed after projecting on to $\mathcal{H}_\mathbf{x}$ with inner products computed after projecting on to $\mathcal{H}_\mathbf{z}$.

The 'collapsed' evidence lower bound obtained from plugging in the optimal choices for $\{\boldsymbol{\mu}_\mathbf{z}, \boldsymbol{\Sigma}_\mathbf{z}\}$ in eq. (2.44) differs only from the log marginal likelihood of inference in the M-dimensional linear model (eq. 2.36) in the terms $t(x_n)$,

$$\underline{\mathcal{L}}(Q^*, \theta) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log\det(\mathbf{Q}) - \frac{1}{2}\mathbf{y}^\top\mathbf{Q}^{-1}\mathbf{y} - \frac{1}{2\sigma^2}\sum_{n=1}^{N} t(x_n) \tag{2.51}$$

$$= -\frac{N}{2}\log 2\pi - \frac{1}{2}\log\det(\mathbf{Q}) - \frac{1}{2}\mathbf{y}^\top\mathbf{Q}^{-1}\mathbf{y} - \frac{1}{2\sigma^2}\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}). \tag{2.52}$$

For simplicity moving forward, we will denote $\underline{\mathcal{L}}(Q^*, \theta) = \underline{\mathcal{L}}(\mathbf{z}, \theta)$. The evidence lower bound resembles the log marginal likelihood of Gaussian process regression (eq. 1.38), after replacing inner

products in $\mathcal{H}_{\mathbf{x}}$ with inner products in $\mathcal{H}_{\mathbf{z}}$ and with the addition of a 'penalty' depending on the distance between points in $\mathcal{H}_{\mathbf{x}}$ and the closest points in $\mathcal{H}_{\mathbf{z}}$ that comes in the form of $\text{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})$.

2.3.4 Generalized inducing points

From the above calculations and their linear algebraic interpretations, we see that the choice of inducing inputs $\mathbf{z}$ constitutes two entangled choices. First, $\mathbf{z}$ determines the subspace $\mathcal{H}_{\mathbf{z}}$, which controls all properties of the approximate posterior. The better we are able to reconstruct vectors in $\mathcal{H}_{\mathbf{x}}$ with vectors in $\mathcal{H}_{\mathbf{z}}$, the better our variational approximation. At the same time, the choice of $\mathbf{z}$ also determines the basis we use for $\mathcal{H}_{\mathbf{z}}$. This controls computational aspects of the approximation. To make predictions, we need to perform projection-like operations onto $\mathcal{H}_{\mathbf{z}}$. Projections can be computed by performing a Cholesky decomposition of $\mathbf{K}_{\mathbf{z},\mathbf{z}}$, which can be seen as performing Gram-Schmidt orthogonalization on $\{f_z : z \in \mathbf{z}\}$ in order to compute an orthogonal basis for $\mathcal{H}_{\mathbf{z}}$ (see Shawe-Taylor and Cristianini 2004, Section 5.2). For computational reasons we would ideally like to define $\mathbf{z}$ in such a way that computing an orthogonal basis is stable and inexpensive.

From both the variational and linear algebraic perspectives, we see that instead of using inducing points of the form f_z for some $z \in \mathcal{X}$, we can select any $f \in \mathcal{H}$ to use as an inducing point. In the literature these are generally referred to as "inter-domain" inducing points (Álvarez and Lawrence, 2008; Lázaro-Gredilla and Figueiras-Vidal, 2009), though we simply refer to them as generalized inducing points, or simply inducing points, as there is no conceptual difference between these and "standard" inducing points. Similarly, we will still use $\mathcal{H}_{\mathbf{z}}$ to denote the span of the generalized inducing points in which case we can think of $\mathbf{z}$ as indexing a subset of linear projections of the process instead of points in the covariate space.

Several methods have focused on the problem of a choice of inducing points that yields a basis with computational benefits (Burt et al., 2020a; Dutordoir et al., 2020; Hensman et al., 2018). Generally these methods allow many inducing points to be used at a low cost, but place constraints on the types of $\mathcal{H}_{\mathbf{z}}$ considered. With these methods $\mathcal{H}_{\mathbf{z}}$ is often not adaptive to the covariates. As a result, even if many inducing points are used so that $\mathcal{H}_{\mathbf{z}}$ is high-dimensional, the approximation may not recover the full model because there are still points in $\mathcal{H}_{\mathbf{x}}$ that are far from $\mathcal{H}_{\mathbf{z}}$. Typically, this problem becomes worse when the input domain is high-dimensional.

2.4 Diagnostics for Approximate Gaussian Process Regression

With variational Bayesian inference it is often difficult to answer the fundamental question: *Have I actually approximated the posterior well?* A line of recent work has begun to build practical diagnostic tools for variational inference to better answer this question (Huggins et al., 2020; Yao et al., 2018). In the case of Gaussian process regression with a Gaussian likelihood (eq. 1.38), one might hope somewhat more can be said with relatively simple techniques due to the additional linear-algebraic structure and because the posterior can be computed analytically. This case was previously considered in Huggins

et al. (2019) who derived an alternative objective function to the evidence lower bound for selecting inducing points that may give better control on the posterior moments.

2.4.1 The Kullback-Leibler Divergence and Bounds on Quantities of Interest

Given that variational inference minimizes the Kullback-Leibler divergence to the posterior, an intuitive way to check that inference is accurate is to see if $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) \leq \varepsilon$, where ε is an acceptable threshold. However, Huggins et al. (2020, Propositions 2.2 and 2.3) and Huggins et al. (2019) observed that a moderately large Kullback-Leibler divergence can lead to very inaccurate inference about moments of the distribution even in the case of Gaussian distributions. We now discuss the types of guarantees on practical quantities that can be obtained from $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) \leq \varepsilon$, including moments in the case when both Q and $P_{|\mathcal{D}}$ are Gaussian. The general theme is that if $\varepsilon \ll 1$, we can expect to obtain non-vacuous guarantees on the approximate posterior while if $\varepsilon \gg 1$, as observed by Huggins et al. (2019), very little can be said about the quality of approximation of moments and other quantities of interest.

Pinsker's Inequality and the Probability of Events

Pinsker's inequality allows us to control the probability of an event under the posterior in terms of the probability of the event under the approximate posterior and the Kullback-Leibler divergence between the approximate posterior and the posterior. Before recalling Pinsker's inequality, we recall the definition and several properties of the total variational distance between probability measures.

Definition 2.3 (Total Variation Distance)**.** *Let P, Q be probability measures on a common measurable space (Ω, Σ). Define the total variational distance between P and Q by,*

$$\mathfrak{D}_{TV}(P, Q) = \sup_{A \in \Sigma} |P(A) - Q(A)|. \tag{2.53}$$

From the definition, we see that total variation is symmetric, non-negative and 0 if and only if $P = Q$. It can also be verified that it obeys a triangle inequality, so total variation distance is, as the name suggests, a distance.

If we let X and X' be random variables with distributions P and Q respectively, then for any measurable $\chi : \Omega \to [0, 1]$ (Polyanskiy and Wu 2014, Theorem 6.3)

$$\left|\mathbb{E}_P[\chi(X')] - \mathbb{E}_Q[\chi(X)]\right| \leq \mathfrak{D}_{\mathrm{TV}}(P, Q). \tag{2.54}$$

The following result relates Kullback-Leibler divergence to the total variation distance.

Theorem 2.4 (Pinsker's Inequality, Polyanskiy and Wu 2014, Theorem 6.5)**.** *For any two probability measures P and P' defined on a common measurable space*

$$\mathrm{TV}(Q, P) \leq \sqrt{\tfrac{1}{2}\mathfrak{D}_{KL}(Q, P)}. \tag{2.55}$$

This bound is vacuous unless $\mathfrak{D}_{\mathrm{KL}}(Q,P) \leq 2$, since total variation distance is never larger than 1 (eq. 2.53). Combining eq. (2.54) and theorem 2.4, for any $a > 0$ and $\chi : \Omega \to [0,a]$

$$|\mathbb{E}_Q[\chi(X')] - \mathbb{E}_P[\chi(X)]| \leq a\sqrt{\tfrac{1}{2}\mathfrak{D}_{\mathrm{KL}}(Q,P)}. \tag{2.56}$$

From eq. (2.56), we conclude if the Kullback-Leibler divergence is small, the approximate posterior yields similar answers to the exact posterior about statistical queries that can be expressed as expectations of bounded measurable functions under the posterior.

Example 2.5 (Hypothesis Testing)**.** *Suppose we are using the model eq. (1.3) and are interested in the question: What is the posterior probability that an observation exceeds* 0 *at a point* x_**,* $P_{|\mathcal{D}}(f(x_*) + \varepsilon_* \geq 0)$*, where* $\varepsilon_* \sim \mathcal{N}(0,\sigma^2)$*? We might be interested in such a question as a form of Bayesian hypothesis test, or due to the need to make a binary decision in some down-stream task that depends on the sign of* $f(0)$*. From the definition of total variation distance (eq. 2.3),*

$$|P_{|\mathcal{D}}(f(x_*) + \varepsilon_* \geq 0) - Q(f(x_*) + \varepsilon_* \geq 0)| \leq \mathfrak{D}_{TV}\left(P_{|\mathcal{D}}, Q\right). \tag{2.57}$$

In the inequality, we use that ε_* *has the same distribution under both the posterior and approximate posterior and that total variation distance satisfies a data-processing inequality analogous to corollary 2.2.*[5] *Applying Pinsker's inequality (eq. 2.4)*

$$\left|P_{|\mathcal{D}}(f(x_*) + \varepsilon_* \geq 0) - Q(f(x_*) + \varepsilon_* \geq 0)\right| \leq \sqrt{\tfrac{1}{2}\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)}. \tag{2.58}$$

Concretely, if $\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq 0.1$

$$\left|P_{|\mathcal{D}}(f(x_*) + \varepsilon_* \geq 0) - Q(f(x_*) + \varepsilon_* \geq 0)\right| \leq \sqrt{0.05} < 0.23. \tag{2.59}$$

Hence, we can conclude the probability assigned by the approximate posterior and the posterior to this event differs by less than 0.23*. We note that in general this bound will be overly pessimistic. To obtain useful results, we will need very good approximations to the posterior. Luckily, in the case of sparse variational Gaussian process regression this is at times possible.*

Moment Bounds via the Kullback-Leibler Divergence

Frequently, moments of distributions (e.g. the posterior mean and variance) are reported. For Gaussian random variables, these cannot be expressed as bounded functions. However, some guarantees on the posterior mean and variance can still be obtained using the form of the Kullback-Leibler divergence between Gaussian distributions. Huggins et al. (2020) explored an alternative and more general technique for obtaining bounds on moments of distributions in terms of the Kullback-Leibler divergence via transport-entropy inequalities.

[5] This can be seen from definition 2.3 after a brief calculation. In fact, such inequalities hold for all f-divergences, see Polyanskiy and Wu (2014, Theorem 6.2)

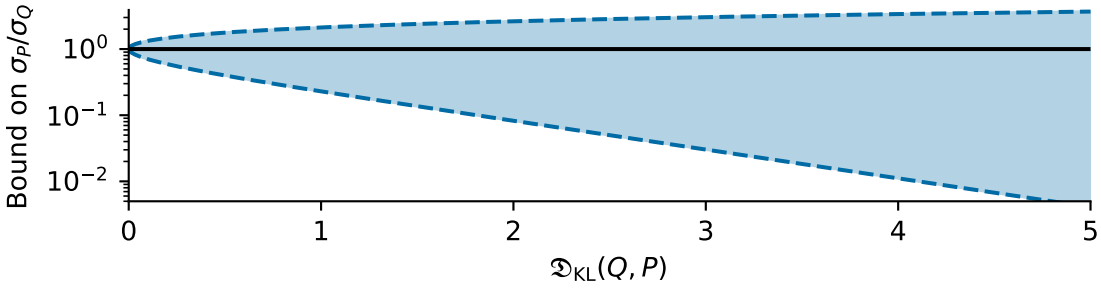


Fig. 2.3 An illustration of the interval in eq. (2.61) as a function of the KL-divergence. The y-axis is plotted on a log scale. The line $\sigma_P/\sigma_Q = 1$ is plotted in black. We see that the bound is asymmetric.

Proposition 2.6 (Kullback-Leibler Divergence, Univariate Gaussian Distributions)**.** *Let* $Q = \mathcal{N}(\mu_Q, \sigma_Q^2)$ *and* $P = \mathcal{N}(\mu_P, \sigma_P^2)$*, then*

$$\mathfrak{D}_{KL}(Q,P) = \frac{1}{2}\Big(-1 - \log\frac{\sigma_Q^2}{\sigma_P^2} + \frac{\sigma_Q^2}{\sigma_P^2} + \frac{(\mu_Q-\mu_P)^2}{\sigma_P^2}\Big). \tag{2.60}$$

Rearranging proposition 2.6 allows us to obtain bounds on the relative error between the variance of P and Q. To state a sharp form of this bound, we need to define the *Lambert W-function*.

Definition 2.7 (Lambert W-function)**.** *Any function* $W : \mathbb{C} \to \mathbb{C}$ *satisfying* $W(r) = re^r$ *for all* $r \in \mathbb{C} \setminus (-\infty, -1/e)$ *is a Lambert W-function.*

For $r \in (-1/e, \infty)$, there are two solutions for $a = ze^r$; we denote the largest of these by $W_0(r)$ (the so-called principal branch of the Lambert-W function) and the smaller solution by $W_{-1}(r)$.

Proposition 2.8. *Let* $Q = \mathcal{N}(\mu_Q, \sigma_Q^2)$ *and* $P = \mathcal{N}(\mu_P, \sigma_P^2)$*, then*

$$\frac{\sigma_Q}{\sigma_P} \in \left[\sqrt{-W_0(e^{-(1+2\mathfrak{D}_{KL}(Q,P))})}, \sqrt{-W_{-1}(-e^{-(1+2\mathfrak{D}_{KL}(Q,P))})}\right]. \tag{2.61}$$

A fortiori, if $\mathfrak{D}_{KL}(Q,P) \leq \frac{1}{10}$

$$\frac{\sigma_Q}{\sigma_P} \in \left[\sqrt{1-\sqrt{6\mathfrak{D}_{KL}(Q,P)}}, \sqrt{1+\sqrt{6\mathfrak{D}_{KL}(Q,P)}}\right]. \tag{2.62}$$

Remark 2.9. *Huggins et al. (2019) gave an example showing the left endpoint of the interval eq. (2.61) goes to* 0 *at least exponentially fast as* $\mathfrak{D}_{KL}(Q,P)$ *increases. For this bound to be useful, we therefore will need the Kullback-Leibler divergence to be small.*

Proof of proposition 2.8. Let $a = \sigma_Q^2/\sigma_P^2$. Using proposition 2.6

$$2\mathfrak{D}_{\mathrm{KL}}(Q,P) + 1 = -\log(a) + a + \tfrac{(\mu_Q-\mu_P)^2}{\sigma_P^2} \geq -\log(a) + a. \tag{2.63}$$

Exponentiating both sides and using that $x \to -1/x$ is monotone increasing,

$$\frac{\exp(a)}{a} \leq \exp(2\mathfrak{D}_{\mathrm{KL}}(Q,P)+1) \Rightarrow -a\exp(-a) \leq -\exp(-(2\mathfrak{D}_{\mathrm{KL}}(Q,P)+1)) \tag{2.64}$$

$$\Rightarrow -a\exp(-a)+\exp(-(2\mathfrak{D}_{\mathrm{KL}}(Q,P)+1)) \leq 0. \tag{2.65}$$

The left-hand side of eq. (2.65) is continuous, positive at $a=0$ and as $a \to \infty$ tends to $-\infty$. From this we conclude that the solutions to eq. (2.65) forms a closed interval with endpoints given by the equality

$$-a\exp(-a)+\exp(-(1+2\mathfrak{D}_{\mathrm{KL}}(Q,P))) = 0. \tag{2.66}$$

This equality has two real solutions,

$$a = -W_0(-\exp(-(1+2\mathfrak{D}_{\mathrm{KL}}(Q,P)))) \quad \text{and} \quad a = -W_1(-\exp(-(1+2\mathfrak{D}_{\mathrm{KL}}(Q,P)))). \tag{2.67}$$

From this, we conclude that $a \in [-W_0(-\exp(-(1+2\mathfrak{D}_{\mathrm{KL}}(Q,P)))), -W_1(-\exp(-(1+2\mathfrak{D}_{\mathrm{KL}}(Q,P))))]$ and taking square roots completes the first claim.

Under the assumption that $2\mathfrak{D}_{\mathrm{KL}}(Q,P) < \frac{1}{5}$, $a-\log(a) < 1.2$. This implies $a \in [0.493, 1.77]$. For a in this range, $a-\log(a)-1 \geq (a-1)^2/3$. Equation (2.62) follows by rearranging. □

Equation (2.62) is strictly weaker than eq. (2.61). However, it is easier to analyze for small Kullback-Leibler divergences. Particularly, eq. (2.62) shows that the ratio of variances converges to 1 with an additive error of $O(\sqrt{\mathfrak{D}_{\mathrm{KL}}(Q,P)})$ as $\mathfrak{D}_{\mathrm{KL}}(Q,P) \to 0$.

Corollary 2.10 (Bounds on Mean, univariate Gaussian)**.** *Let $Q = \mathcal{N}(\mu_Q, \sigma_Q^2)$ and $P = \mathcal{N}(\mu_P, \sigma_P^2)$, then*

$$\frac{1}{\sqrt{-W_{-1}(-e^{-(1+2\mathfrak{D}_{KL}(Q,P))})}} \frac{|\mu_P-\mu_Q|}{\sigma_Q} \leq \frac{|\mu_P-\mu_Q|}{\sigma_P} \leq \sqrt{2\mathfrak{D}_{KL}(Q,P)}. \tag{2.68}$$

Corollary 2.11. *Let $Q \sim \mathcal{GP}(\mu_Q, k_Q)$ and $P \sim \mathcal{GP}(\mu_P, k_P)$. Define $\sigma_Q(x) = \sqrt{k_Q(x,x)}$ and $\sigma_P = \sqrt{k_P(x,x)}$. Then, for all $x \in \mathcal{X}$,*

$$\frac{\sigma_Q(x)}{\sigma_P(x)} \in \left[\sqrt{-W_0(-e^{-(1+2\mathfrak{D}_{KL}(Q,P))})}, \sqrt{-W_{-1}(-e^{-(1+2\mathfrak{D}_{KL}(Q,P))})}\right], \tag{2.69}$$

$$\left|\frac{\mu_Q(x)-\mu_P(x)}{\sigma_P(x)}\right| \leq \sqrt{2\mathfrak{D}_{KL}(Q,P)} \qquad \textit{and} \tag{2.70}$$

$$\left|\frac{\mu_Q(x)-\mu_P(x)}{\sigma_Q(x)}\right| \leq \sqrt{2\mathfrak{D}_{KL}(Q,P)}\sqrt{-W_{-1}(-e^{-(1+2\mathfrak{D}_{KL}(Q,P))})}. \tag{2.71}$$

Proof. Combine proposition 2.8, corollary 2.10 and the data-processing inequality (corollary 2.2). □

Corollary 2.11 implies that if Q and P are Gaussian processes and $\mathfrak{D}_{\mathrm{KL}}(Q,P)$ is small, the mean functions of Q and P must be close (as measured by the standard deviation of both distributions), and the marginal standard deviations must also be close in a relative sense. On the other hand for

even moderately large Kullback-Leibler divergences (greater than several nats), these bounds can be hopelessly loose (Huggins et al., 2019, Proposition 3.1).

Linear Algebraic Bounds on Posterior Moments

The bounds computed in the last section provided uniform bounds on posterior moments in terms of $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)$. If we are interested in the posterior moments at a particular point, this is rather indirect, as the Kullback-Leibler divergence depends on the entirety of both processes. We can instead provide posterior checks by directly considering the form of the Gaussian process predictive variance (eq. 1.6) and using standard matrix inequalities. We refer the reader to table 1 for the matrix notation and to appendix A for properties of matrices used in this section and the remainder of the thesis. The main tool used in these bounds will be that the Nyström approximation results in a matrix that lower bounds the kernel matrix in the positive definite ordering. This can be seen by considering the matrix,

$$\mathbf{K}_{+} = \begin{pmatrix} \mathbf{K}_{\mathbf{x},\mathbf{x}} & \mathbf{K}_{\mathbf{x},\mathbf{z}} \\ \mathbf{K}_{\mathbf{x},\mathbf{z}}^{\top} & \mathbf{K}_{\mathbf{z},\mathbf{z}}. \end{pmatrix}. \tag{2.72}$$

Because $\mathbf{K}_{+}$ is formed by evaluating the kernel function, and the kernel is positive definite semi-definite, $\mathbf{K}_{+}$ is positive semi-definite. From proposition A.15, the *Schur complement* $\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}$ is positive definite, so $\mathbf{K}_{\mathbf{x},\mathbf{x}} \succ \mathbf{Q}_{\mathbf{x},\mathbf{x}}$.

We begin by quoting bounds on the moments from Davies (2015).[6]

Proposition 2.12 (Davies, 2015, Equations 94 and 95)**.** *Define* $\sigma_P(x) = k(x,x) - \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x}x}$. *Suppose* $\mathbf{z} \subset \mathbf{x}$ *and let* $\overline{\mathbf{z}} = \mathbf{x} \setminus \mathbf{z}$. *Then*

$$\max(0, k(x,x) - \mathbf{k}_{x\mathbf{z}}(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{\mathbf{z}x} - \tfrac{1}{\sigma^2}\|\mathbf{k}_{x\overline{\mathbf{z}}} - \mathbf{k}_{x\mathbf{z}}(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{\mathbf{z}\overline{\mathbf{z}}}\|_2^2) \tag{2.73}$$

$$\leq \sigma_P(x) \leq k(x,x) - \mathbf{k}_{x\mathbf{z}}(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{\mathbf{z}x}. \tag{2.74}$$

The upper bound in proposition 2.12 is a subset-of-data approximation, which will be slow to converge in many instances when variational inference succeeds. For example, if all the function values at the observed data are heavily correlated under the prior, a handful of $\mathbf{z}$ will recover the posterior using variational methods, but not with a subset-of-data approach. In this instance the bounds in proposition 2.12 will be very slow to converge. We therefore derive the following bounds built on the machinery of Nyström approximation.

Proposition 2.13. *Define* $\sigma_P(x) = k(x,x) - \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x}x}$. *Let* $\mathbf{Q}_{\mathbf{x},\mathbf{x}} = \mathbf{K}_{\mathbf{x},\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^{\top}$ *and* $\mathbf{Q} = \mathbf{Q}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}$ *Then*

$$\max(0, k(x,x) - \mathbf{k}_{x\mathbf{x}}\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x}) \leq \sigma_P(x) \leq k(x,x) - \mathbf{k}_{x\mathbf{x}}(\mathbf{Q} + \mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)\mathbf{I})^{-1}\mathbf{k}_{\mathbf{x}x}. \tag{2.75}$$

[6] A square appears to be missing in equation 95 on term of Davies' lower bound involving a norm, which we include.

Proof. Observe that $\mathbf{Q} \prec \mathbf{K} \prec \mathbf{Q} + \operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})\mathbf{I}$ and use that $\lambda \to -1/\lambda$ is matrix monotone (i.e. it preserves the Loewner order, Proposition A.19). □

Remark 2.14. *The computational cost of the bounds in proposition 2.13 are higher than the cost of making predictions in variational Gaussian process regression. In particular, predictions in variational Gaussian process regression can be made in $O(M^2)$ after initial cost $O(NM^2)$, whereas the cost of these bounds is $O(NM^2)$ for each new prediction.*

The variance of variational Gaussian process regression (eq. 2.48) is neither an upper nor lower bound on the posterior variance. We advocate for the use of the variational predictive variance for inference, and using proposition 2.13 to ensure that the posterior variance cannot differ from the variational approximation by a large amount.

Proposition 2.15. *Recall for all $x \in \mathcal{X}$, $\hat{\mu}(x) = \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{y}$.*

$$|\hat{\mu}(x) - \mathbf{k}_{x\mathbf{x}}\mathbf{Q}^{-1}\mathbf{y}| \leq \frac{\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\sigma^2}\|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x}\|\|\mathbf{y}\|. \tag{2.76}$$

Proof. Factoring gives

$$\mathbf{k}_{x\mathbf{x}}(\mathbf{Q}^{-1} - \mathbf{K}^{-1})\mathbf{y} = \mathbf{k}_{x\mathbf{x}}\mathbf{Q}^{-1}(\mathbf{K} - \mathbf{Q})\mathbf{K}^{-1}\mathbf{y}. \tag{2.77}$$

From the Cauchy-Schwarz inequality and the definition of the operator norm,

$$\mathbf{k}_{x\mathbf{x}}\mathbf{Q}^{-1}(\mathbf{K} - \mathbf{Q})\mathbf{K}^{-1}\mathbf{y} \leq \|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x}\|_2\|\mathbf{K} - \mathbf{Q}\|_{\text{op}}\|\mathbf{K}^{-1}\mathbf{y}\|_2. \tag{2.78}$$

Observe $\|\mathbf{K} - \mathbf{Q}\|_{\text{op}} \leq \operatorname{tr}(\mathbf{K} - \mathbf{Q})$ and $\|\mathbf{K}^{-1}\mathbf{y}\|_2 \leq \|\mathbf{K}^{-1}\|_{\text{op}}\|\mathbf{y}\|_2 \leq \frac{1}{\sigma^2}\|\mathbf{y}\|_2$. □

For proposition 2.13 and proposition 2.15 to be useful, $\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})$ must be small, or $\|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x}\|_2$ must be small. We will see in the next section that the former is a sufficient condition for $\mathfrak{D}_{\text{KL}}\left(Q, P_{|\mathcal{D}}\right)$ to be small, while the latter occurs if the point x is far from all the observed data.

2.4.2 Upper Bounds on the Kullback-Leibler Divergence for Variational Gaussian Process Regression

Unfortunately, even though we minimize $\mathfrak{D}_{\text{KL}}\left(Q, P_{|\mathcal{D}}\right)$ in variational inference, we are almost never able to actually compute it. If we could, we could also evaluate the log marginal likelihood directly using eq. (2.9). In the case of Gaussian process regression, a straight-forward evaluation of $\mathfrak{D}_{\text{KL}}\left(Q, P_{|\mathcal{D}}\right)$ would involve evaluating the posterior distribution over $f_{\mathbf{x}}, f_{\mathbf{z}}|\mathcal{D}$ which is prohibitive in instances where variational Gaussian process regression is applied. However, Titsias (2014) showed that we can provide upper bounds on the log marginal likelihood that can be computed in $O(NM^2)$, the same cost as the evidence lower bound (eq. 2.52). We now present a refinement of this upper bound, derived in work done with Artem Artemev and Mark van der Wilk (Artemev et al., 2021).

Recall

$$\mathcal{L}(\theta) = c - \frac{1}{2}\underbrace{\log\det(\mathbf{K})}_{\text{log det.}} - \frac{1}{2}\underbrace{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}_{\text{quadratic}}. \tag{2.79}$$

Observe that the log marginal likelihood has two terms that are expensive to evaluate, and we can upper bound each term separately. The key ideas to obtain the desired bounds are: 1. $\mathbf{Q} \prec \mathbf{K}$ (proposition A.15) 2. We can evaluate $\operatorname{tr}(\mathbf{K})$ in $O(N)$ time.

Proposition 2.16. *For* $\mathbf{K} = \mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}$, $\mathbf{Q} = \mathbf{Q}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}$ *and* $\mathbf{Q}_{\mathbf{x},\mathbf{x}} \prec \mathbf{K}_{\mathbf{x},\mathbf{x}}$, *then*

$$\log\det\mathbf{K} \geq \log\det\mathbf{Q} + \log\left(1 + \frac{\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2}\right) \tag{2.80}$$

where $\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}})$ *denotes the largest eigenvalue of* $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$.

The only quantity in this bound that is not calculated already in variational inference is $\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}})$. This can be calculated (to machine precision) in $O(NM^2)$ by computing the square of the largest singular value of $\mathbf{L}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}$, where $\mathbf{L}$ is a right Cholesky factor of $\mathbf{K}_{\mathbf{z},\mathbf{z}}$. Alternatively, it can be upper bounded by $\operatorname{tr}(\mathbf{Q}_{\mathbf{x},\mathbf{x}})$. Hence, the total computational cost of this lower bound is $O(NM^2)$.

Proof of proposition 2.16. Let $\lambda_n(\mathbf{K})$ denote the eigenvalues of $\mathbf{K}$ and $\lambda_n(\mathbf{Q})$ denote the eigenvalues of $\mathbf{Q}$. As $\mathbf{K} \succ \mathbf{Q}$ proposition A.19 implies $\lambda_n(\mathbf{K}) - \lambda_n(\mathbf{Q}) := e_n \geq 0$. By proposition A.9, $\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}) = \operatorname{tr}(\mathbf{K} - \mathbf{Q}) = \sum_{n=1}^N e_n$. Writing the determinant in terms of eigenvalues (proposition A.13):

$$\log\det\mathbf{K} = \sum\log\left(e_n + \lambda_n(\mathbf{Q})\right) = \log|\mathbf{Q}| + \sum\log\left(1 + \frac{e_n}{\lambda_n(\mathbf{Q})}\right). \tag{2.81}$$

Rewriting the second term on the right-hand side as the log of a product and using that $e_n \geq 0$,

$$\sum_{n=1}^N \log\left(1 + \frac{e_n}{\lambda_n(\mathbf{Q})}\right) = \log\prod_{n=1}^N\left(1 + \frac{e_n}{\lambda_n(\mathbf{Q})}\right) \geq \log\left(1 + \sum_{n=1}^N \frac{e_n}{\lambda_n(\mathbf{Q})}\right). \tag{2.82}$$

The inequality comes from keeping only the terms that are linear in the second argument after expanding the product. Because $\lambda_n(\mathbf{Q}) \leq \lambda_1(\mathbf{Q}) = \lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2$,

$$\log\det\mathbf{K} \geq \log\det\mathbf{Q} + \log\left(1 + \frac{\sum_{n=1}^N e_n}{\lambda_1(\mathbf{Q})}\right) = \log\det\mathbf{Q} + \log\left(1 + \frac{\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2}\right). \tag{2.83}$$

□

Proposition 2.17 (Upper Bound on Quadratic Term, Titsias, 2014)**.**

$$\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} \geq \mathbf{y}^\top(\mathbf{Q} + \operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})\,\mathbf{I})^{-1}\mathbf{y}. \tag{2.84}$$

Proof. This follows from $\mathbf{K} \prec \mathbf{Q} + \operatorname{tr}(\mathbf{K} - \mathbf{Q})\,\mathbf{I}$ and that $\lambda \to -1/\lambda$ preserves the Loewner order. □

Proposition 2.18 (A Posteriori Upper bound on Kullback divergence for sparse Gaussian process regression)**.** *Let* $T = \operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})$

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{T}{2}\left(\frac{1}{\sigma^2} - \frac{1}{T}\log\left(1 + \frac{T}{\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2}\right) + y^\top \mathbf{Q}^{-1}(\mathbf{Q} + T\mathbf{I})^{-1}\mathbf{y}\right). \tag{2.85}$$

Proof. $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) = \underline{\mathcal{L}}(\theta) - \underline{\mathcal{L}}(\mathbf{z}, \theta)$ (eq. 2.94). Combine proposition 2.16 and proposition 2.17 to upper bound $\underline{\mathcal{L}}(\theta)$, subtract eq. (2.52), and rearrange the quadratic form in the difference using $(\mathbf{A}+\mathbf{B})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{A}+\mathbf{B})^{-1}$. □

The upper bound in proposition 2.18 can be computed in $O(NM^2)$, making it a practical diagnostic for variational inference in Gaussian process regression. Together with corollary 2.11, we can obtain uniform guarantees on the posterior mean and variance of the approximate posterior relative to the exact posterior and, using eq. (2.56), on the probability of events under the approximate posterior relative to the true posterior. For any of these bounds to be useful, we will need the upper bound on the Kullback-Leibler divergence to be quite small.

2.4.3 Example: Assessing Variational Inference on the Mauna Loa CO_2 Dataset

We now provide an example of approximate inference and corresponding diagnostics with fixed model hyperparameters. We find that the diagnostic tools in this instance can guarantee that the approximate posterior is a reasonable approximation to the posterior. However, obtaining these guarantees generally requires many more inducing points than obtaining a high-quality approximate posterior.

The dataset considered consists of monthly average CO_2 readings dating between 1958 and 2021 taken on Mauna Loa (Tans and Keeling, 2022). The response variable is the monthly average CO_2 in parts per million (PPM). The covariate is time. There are 771 observations in the dataset.

We follow a simplified version of the kernel selection process discussed in Rasmussen and Williams (2006, Chapter 5), and consider a kernel of the form,

$$k(x,x') = \theta_1 \exp\left(-\frac{(x-x')^2}{2\theta_2}\right) + \theta_3 \exp\left(-\frac{(x-x')^2}{2\theta_4} - \frac{0.5\sin^2(\pi|x-x'|)}{\theta_5}\right). \tag{2.86}$$

This kernel is a sum of a squared exponential kernel, intended to model long term trends, as well as a quasi-periodic kernel, with a period of one year that can model short-scale variation in the data as well as seasonal trends. A more detailed description of a modeling philosophy for this problem, as well as a more involved kernel that may be more appropriate, is given in Rasmussen and Williams (2006, Chapter 5). We select $\{\{\theta_i\}_{i=1}^5, \sigma^2\}$ via maximum marginal likelihood, and defer discussion of approximate maximum marginal likelihood to the next section.

Care is needed in extrapolating with this model; sufficiently far from the data CO_2 levels will be entirely uncorrelated with the observed data and the posterior will return to the empirical data mean. The use of a linear, or other non-stationary kernel may be appropriate to alleviate this and model longer term increasing trends, although for long term trends sustained linear growth is also unrealistic.

We consider this example only as an example to illustrate approximate Gaussian process inference diagnostics and defer to those with domain-expertise in climate modeling for more realistic predictions of CO_2 concentration trends.

We consider the following simple questions: 1. What is the probability (under the model) that CO_2 levels in January 2030 exceed 440 parts per million (PPM)? 2. Construct a 95% Bayesian credible interval for CO_2 levels in January 2030.

In the following example inducing points are selected via algorithm 1 (chapter 3). This algorithm is equivalent to taking $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$ to be an incomplete pivoted Cholesky decomposition of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$, as in Fine and Scheinberg (2001).

A Hypothesis Testing Approach

We consider 3 different answers to the first question. The correct answer is given by computing,

$$P_{|\mathcal{D}}(f(x_*) + \varepsilon_* \geq 440) = 0.146, \tag{2.87}$$

with $x_* = 2030$ and $\varepsilon_* \sim \mathcal{N}(0, \sigma^2)$. The natural approximate Bayesian analogue is given by,

$$Q(f(x_*) + \varepsilon_* \geq 440), \tag{2.88}$$

which we compute for several choices of M. Finally, the interval

$$\left[Q(f(x_*) + \varepsilon_* \geq 440) - \sqrt{\tfrac{1}{2}\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)}, Q(f(x_*) + \varepsilon_* \geq 440) + \sqrt{\tfrac{1}{2}\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)}\right] \tag{2.89}$$

is *guaranteed* to contain the probability in eq. (2.87) by Pinsker's inequality (theorem 2.4). The upper bound on the Kullback-Leibler divergence in proposition 2.18 can be used in eq. (2.89) to construct a larger, but faster to compute interval.

Figure 2.4 shows the posterior probability, the probability under the approximate posterior and the interval from eq. (2.89), computed using the upper bound on $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)$ from proposition 2.18. With smaller M, variational Gaussian process regression can overestimate or underestimate the probability, but converges to the correct value up to high precision for $M > 150$. On the other hand, the interval given in eq. (2.89) is not useful until $M > 250$. For larger M, it quickly converges to the posterior probability.

Moment Bounds and Credible Intervals

Let $\sigma_P(x)$ denote the posterior variance at a point $x \in \mathcal{X}$ (eq. 1.6) and $\sigma_Q(x)$ the variational approximation to this quantity given by eq. (2.48). The interval

$$I_{P_{|\mathcal{D}}} = [\hat{\mu}(x^*) - 1.96(\sigma_P(x^*) + \sigma^2), \hat{\mu}(x^*) + 1.96(\sigma_P(x^*) + \sigma^2)] = [435, 441] \tag{2.90}$$

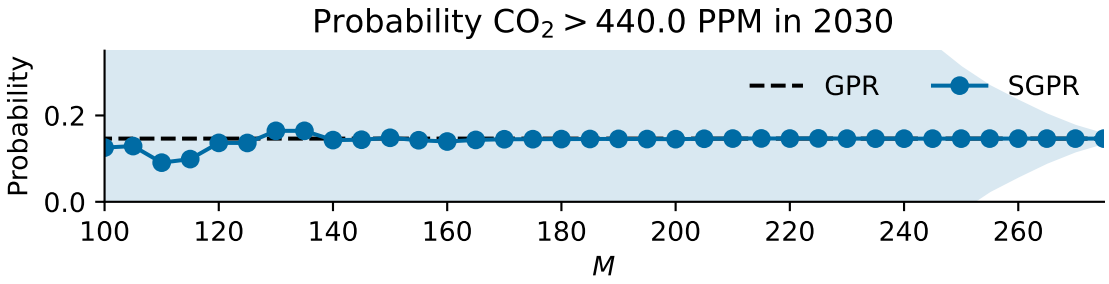


Fig. 2.4 The probability CO_2 levels exceed 440 parts per million under the posterior is shown by the black dashed line. The blue line and dots show the probability assigned to the same event by the approximate posterior for various M. There is originally some discrepancy, but for $M > 140$, the two probabilities are indistinguishable. The blue shaded region shows an interval guaranteed to contain the posterior probability (proposition 2.18 and eq. 2.89). This converges slower than the approximate posterior probability, but eventually guarantees the probability under the posterior is contained in a narrow interval.

is a 95% credible region under the posterior for emissions in the year $x^* = 2030$ by simply taking the interval. The constant 1.96 comes from inverting a Gaussian cumulative density function. This interval has the variational Bayesian analogue

$$I_Q = [\hat{\mu}_Q(x^*) - 1.96(\sigma_Q(x^*) + \sigma^2), \hat{\mu}_Q(x^*) + 1.96(\sigma_Q(x^*) + \sigma^2)]. \tag{2.91}$$

Adding upper bounds on the standard deviation (eq. 2.13) to upper and lower bounds on the mean (eq. 2.15) produces the interval

$$\begin{aligned} I_{\text{inflated}} = \Big[&\mathbf{k}_{x^*\mathbf{x}}\mathbf{Q}^{-1}\mathbf{y} - \tfrac{\operatorname{tr}(\mathbf{K}-\mathbf{Q})}{\sigma^2}\|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x^*}\|\|\mathbf{y}\| - 1.96\sqrt{k(x^*,x^*) - k_{x^*\mathbf{x}}(\mathbf{Q} + \operatorname{tr}(\mathbf{K}-\mathbf{Q})\mathbf{I})^{-1}\mathbf{k}_{\mathbf{x}x^*} + \sigma^2}, \\ &\mathbf{k}_{x^*\mathbf{x}}\mathbf{Q}^{-1}\mathbf{y} + \tfrac{\operatorname{tr}(\mathbf{K}-\mathbf{Q})}{\sigma^2}\|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x^*}\|\|\mathbf{y}\| + 1.96\sqrt{k(x^*,x^*) - k_{x^*\mathbf{x}}(\mathbf{Q} + \operatorname{tr}(\mathbf{K}-\mathbf{Q})\mathbf{I})^{-1}\mathbf{k}_{\mathbf{x}x^*} + \sigma^2}\Big], \end{aligned} \tag{2.92}$$

which is guaranteed to contain $I_{P_{|\mathcal{D}}}$. On the other hand, adding lower bounds on the standard deviation (eq. 2.13) to lower and upper bounds on the mean (eq. 2.15) produces the interval,

$$\begin{aligned} I_{\text{deflated}} = \Big[&\mathbf{k}_{x^*\mathbf{x}}\mathbf{Q}^{-1}\mathbf{y} + \tfrac{\operatorname{tr}(\mathbf{K}-\mathbf{Q})}{\sigma^2}\|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x^*}\|\|\mathbf{y}\| - 1.96\sqrt{k(x^*,x^*) - \mathbf{k}_{x^*\mathbf{x}}\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x^*} + \sigma^2}, \\ &\mathbf{k}_{x^*\mathbf{x}}\mathbf{Q}^{-1}\mathbf{y} - \tfrac{\operatorname{tr}(\mathbf{K}-\mathbf{Q})}{\sigma^2}\|\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x^*}\|\|\mathbf{y}\| + 1.96\sqrt{k(x^*,x^*) - \mathbf{k}_{x^*\mathbf{x}}\mathbf{Q}^{-1}\mathbf{k}_{\mathbf{x}x^*} + \sigma^2}\Big]. \end{aligned} \tag{2.93}$$

I_{deflated} is contained in $I_{P_{|\mathcal{D}}}$ and may be empty. In summary $I_{\text{deflated}} \subset I_{P_{|\mathcal{D}}} \subset I_{\text{inflated}}$. Moreover, both of the bounding intervals can be computed in $O(NM^2)$, and if $\mathbf{x} \subset \mathbf{z}$ the inclusions become equalities. For small M, the error bound on the predictive mean from proposition 2.15 can be very loose. We therefore take the minimum of it and the bound on the mean from corollary 2.11 using the upper bound on σ_P given by proposition 2.13 and modify eqs. (2.92) and (2.93) accordingly.

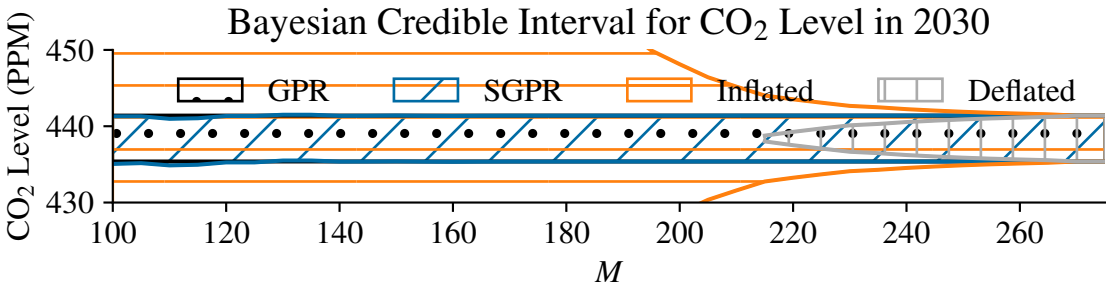


Fig. 2.5 A 95% Bayesian credible interval for CO_2 concentrations in 2030 is constructed. $I_{P_{|\mathcal{D}}}$ is shown in black, while I_Q is shown for various numbers of inducing points in blue. I_{inflated}, which is guaranteed to contain the posterior credible interval is shown in orange. I_{deflated}, which is guaranteed to be contained in the posterior credible interval is shown in gray. The variational approximation results in a similar credible interval for all $M > 100$ to the posterior. The intervals guaranteed to contain and be contained in the credible interval converge more slowly, though provide strong guarantees for $M > 240$.

Figure 2.5 plots 95% credible intervals for the posterior $I_{P_{|\mathcal{D}}} = [435, 441]$, as well as for the approximate posterior given by variational inference for various values of M. We see good agreement (within 1 part per million on both sides) between the credible region computed with the variational posterior (eq. 2.91) and the posterior (eq. 2.90) for all $M \geq 100$. I_{inflated} (eq. 2.92) only becomes reasonably narrow for $M > 200$ (for example, for $M = 210$ we have $I_{\text{inflated}} = [431, 445]$). I_{deflated} is empty for $M < 215$, but for larger M expands to a reasonable approximation of $I_{P_{|\mathcal{D}}}$.

2.4.4 Summary of Diagnostics For Approximate Gaussian Process Regression

Due to the linear-algebraic structure of Gaussian process regression, there are more tools available to assess the quality of variational inference than are typically available in variational Bayesian inference. Bounds can be derived indirectly, via first bounding the Kullback-Leibler divergence to the posterior using linear algebra (proposition 2.18), then using information theoretic properties (theorem 2.4) or the form of the Gaussian Kullback-Leibler divergence (corollary 2.11). Alternatively, direct bounds can be obtained through relatively mechanical linear algebra calculations (propositions 2.13 and 2.15). When many inducing points are used these bounds provide strong guarantees. However, for reasonably small numbers of inducing points, which are often used in practice for computational reasons, the guarantees given by this approach can be vacuous even in cases where variational inference captures salient properties of the posterior.

2.5 Model Selection with the Evidence Lower Bound

To this point, our discussion has focused on approximate inference in Gaussian process regression with fixed hyperparameters. We now turn to the problem of approximate maximum marginal likelihood model selection.

2.5.1 Model Selection and Variational Bayesian Inference

As well as providing a framework for approximate inference, variational inference with the Kullback-Leibler divergence also yields an efficiently computable approximation to model selection via maximum marginal likelihood. From eq. (2.9)

$$\underline{\mathcal{L}}(Q,\theta) = \mathcal{L}(\theta) - \mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right), \tag{2.94}$$

and so $\underline{\mathcal{L}}(Q,\theta)$ is a lower bound on the log marginal likelihood that is tight if $Q \approx P_{|\mathcal{D}}$. Parameters can be selected using the evidence lower bound as a surrogate to the log marginal likelihood, i.e. by solving the maximization problem,

$$\theta_{\mathrm{ELBO}} \in \arg\max_{\theta\in\Theta} \max_{Q\in\mathcal{Q}} \underline{\mathcal{L}}(Q,\theta). \tag{2.95}$$

We refer to approximations to the maximization in eq. (2.95) as evidence lower bound maximization. In practice, the maximization problem can be (approximately) solved, via first (approximately) solving the maximization over Q and then (approximately) solving the maximization over θ and repeating this until convergence in a procedure known as variational expectation maximization (Attias, 1999). Alternatively joint optimization over all variables can be performed to simultaneously select Q and θ. We should not expect $\theta_{\mathrm{ELBO}} = \theta_{\mathrm{MML}}$ unless the posterior corresponding to the model with hyperparameters θ_{MML} is in the variational family. Generally, evidence lower bound maximization leads to bias in model selection (Bauer et al., 2016; Turner and Sahani, 2011). Recently, progress has been made in obtaining consistency guarantees and rates of convergence for selecting between mixture models via maximization of the evidence lower bound (Chérief-Abdellatif, 2019). However, we are not aware of consistency results for evidence lower bound maximization in Gaussian process regression, and cannot reasonably hope for consistency in instances in which maximum marginal likelihood itself is not consistent (see section 1.2.4). We typically expect the bias introduced by evidence lower bound maximization to be small if the variational family contains good approximations to the posterior associated to the optimal hyperparameters. Particularly, if $\min_{Q\in\mathcal{Q}} \mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D};\theta_{\mathrm{MML}}}\right) \leq \varepsilon$, where $P_{|\mathcal{D};\theta_{\mathrm{MML}}}$ denotes the posterior

distribution associated to the maximum marginal likelihood value choice of hyperparameters, then

$$\begin{aligned}\mathcal{L}(\theta_{\text{ELBO}}) &\geq \max_{Q\in\mathcal{Q}} \mathcal{L}(Q, \theta_{\text{ELBO}}) && (2.96)\\ &\geq \max_{Q\in\mathcal{Q}} \mathcal{L}(Q, \theta_{\text{MML}}) && (2.97)\\ &= \mathcal{L}(\theta_{\text{MML}}) - \min_{Q\in\mathcal{Q}} \mathfrak{D}_{\text{KL}}\left(Q, P_{|\mathcal{D};\theta_{\text{MML}}}\right) && (2.98)\\ &\geq \mathcal{L}(\theta_{\text{MML}}) - \varepsilon. && (2.99)\end{aligned}$$

Such a bound requires that the variational approximation is accurate for the maximum likelihood parameters, which are unknown. Additionally, this sequence of inequalities is only informative about parameters obtained if we are able to solve the optimization problem eq. (2.95) globally, which is not usually the case. However, the intuition that accurate approximations to the posterior lead to better model selection is still often useful.

Empirically, maximizing the evidence lower bound generally leads to overestimation of the noise variance (Bauer et al., 2016; Titsias, 2009), which is consistent with the fact that upper bounds on the Kullback-Leibler divergence scale linearly in the inverse noise variance (eq. 2.18). Similarly, one tends to overestimate kernel lengthscale parameters, and underestimate the signal-to-noise ratio; all of these biases lead to models that are easier to approximate with sparse variational inference. As an example of biases introduced, we return to the case of estimating a single scale parameter, discussed in section 1.2.4 in the case of model selection via maximum marginal likelihood.

Evidence Lower Bound Maximization for a Scale Parameter Assume fixed covariates and

$$y_n = f(x_n) + \varepsilon_n, \quad \varepsilon_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \theta_0\sigma^2), \quad f \sim \mathcal{GP}(0, \theta_0 k) \tag{2.100}$$

for some unknown $\theta_0 > 0$. Setting derivatives to 0, one can show that the maximum evidence lower bound solution is

$$\theta_{\text{ELBO}} = \tfrac{1}{N}\mathbf{y}^\top\mathbf{Q}^{-1}\mathbf{y} = \theta_{\text{MML}} + \tfrac{1}{N}\mathbf{y}^\top(\mathbf{Q}^{-1} - \mathbf{K}^{-1})\mathbf{y}. \tag{2.101}$$

Because $\mathbf{K} \succ \mathbf{Q}$ the second term is non-negative. Unlike eq. (1.41), θ_{ELBO} is biased and will overestimate the scale parameter on average. Performing a change of variables gives

$$\theta_{\text{ELBO}} = \theta_{\text{MML}} + \frac{\theta_0}{N}\mathbf{w}^\top(\mathbf{K}^{1/2}\mathbf{Q}^{-1}\mathbf{K}^{1/2} - \mathbf{I})\mathbf{w}, \tag{2.102}$$

with $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The distribution of eq. (2.102) is generalized χ^2. Unlike in the maximum marginal likelihood case, the properties of this distribution depends on the covariates as well as the inducing variables through the matrix $\mathbf{Q}^{-1} - \mathbf{K}^{-1}$, which is small if the variational approximation is good. In figure 2.6, we run the same simulation as in figure 1.5 with $N = 100$, this time plotting histograms

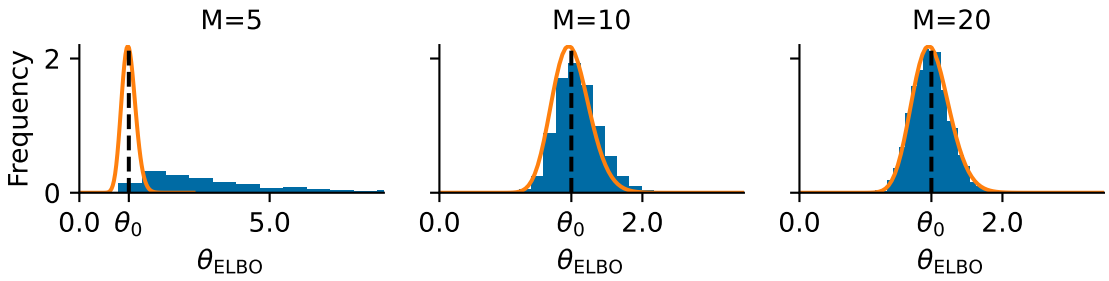


Fig. 2.6 A simulation generating data according to eq. (2.100) with a squared exponential kernel and $N = 100$ datapoints. The density function of θ_{MML} is shown in orange, while the histogram represents (normalized) counts of θ_{ELBO} with 2000 simulations. For small M, there is a large systematic bias and the distribution has a strong positive skew. For larger M the distribution resembles the χ^2-distribution of the maximum likelihood estimate. Note that the orange density curve is the same in all 3 plots, but the x axis is changed to make the blue histogram more visible.

corresponding to θ_{ELBO}. The positive bias in θ_{ELBO} is large when very few inducing points are used, but becomes small once sufficiently many are used to form a good approximation to the posterior.

2.5.2 Assessing the Quality of Hyperparameter Selection

Unfortunately, the behavior of model selection with the evidence lower bound in place of the marginal likelihood seems far more difficult to diagnose than posterior inference with fixed hyperparameters. Since the log marginal likelihood is non-convex and can be very flat or multi-modal, we cannot reasonably hope to obtain a posteriori guarantees telling us that optimization of the evidence lower bound will yield similar hyperparameter estimates as maximum marginal likelihood. Additionally, even if global optimization can be performed, the degree of bias introduced depends on the tightness of the evidence lower bound at θ_{MML}, which is unknown.

Any rigorous, a posteriori guarantees on optimization of the evidence lower bound appear to be a challenging task. However, we can assess an upper bound on the Kullback-Leibler divergence throughout training, and hope that because the evidence lower bound is close to the log marginal likelihood, optimization will lead to similar solutions. Some caution is needed in drawing conclusions from the Kullback-Leibler divergence being small; in particular, a common failure mode of evidence lower bound maximization when insufficiently many inducing points are used is to select a very large noise variance and model the observed data entirely as noise. This results in a small Kullback-Leibler divergence to the posterior for the model considered. This problem can be resolved by adding more inducing points, but cannot always be detected by monitoring the Kullback-Leibler divergence during training.

We return to the Mauna Loa example discussed in the previous section, this time selecting model hyperparameters with evidence lower bound maximization. We plot both the actual Kullback-Leibler

divergence and the computationally efficient upper bound (proposition 2.18) against the number of iterations of training for several M in figure 2.7. 150 inducing points suffice to obtain a reasonable approximation to the marginal likelihood throughout training, and once 200 inducing points are used, the evidence lower bound is essentially indistinguishable from the marginal likelihood throughout training, leading to essentially identical hyperparameter selection. On the other hand, obtaining reasonably tight upper bounds on the log marginal likelihood and Kullback-Leibler divergence throughout training requires at least 225 inducing points, and depending on the quality of approximation desired, perhaps more. In general, we expect this number to be heavily dependent on of the dataset, initialization of parameters and class of models considered.

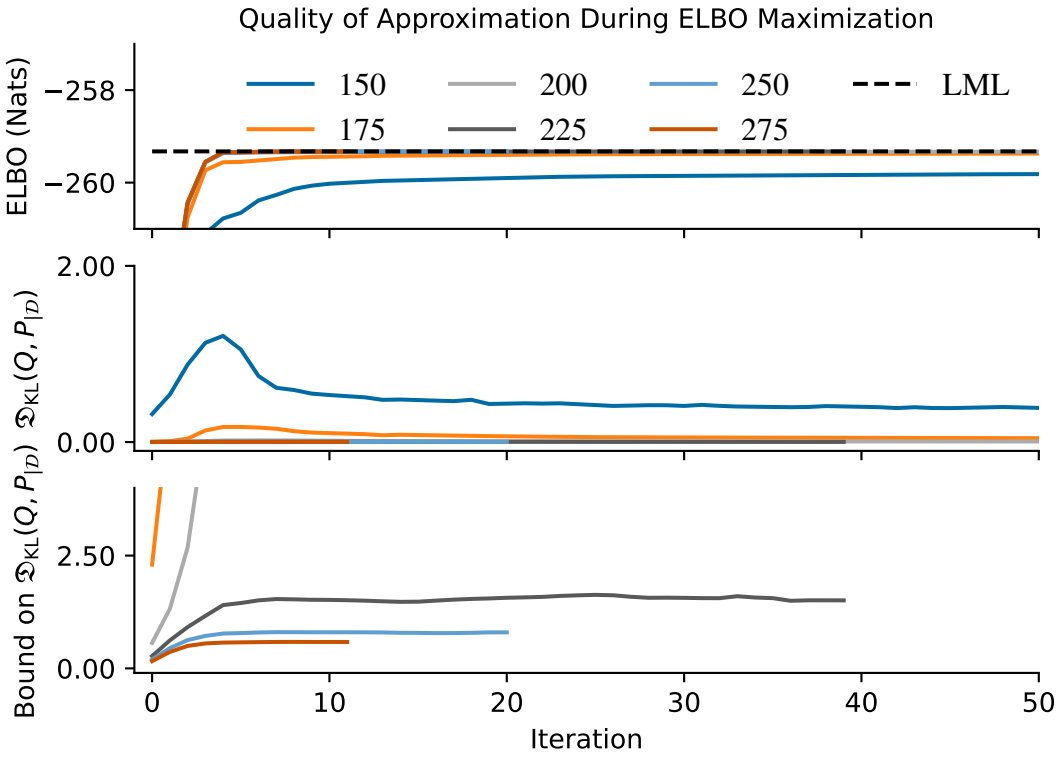


Fig. 2.7 The evidence lower bound (top), Kullback-Leibler divergence to the posterior (center) and the efficiently computable upper bound on the Kullback-Leibler divergence (proposition 2.18, bottom) plotted against iteration of evidence lower bound maximization using L-BFGS. We see that upper bounds on $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)$ can be quite pessimistic, potentially leading to using more inducing points than are actually needed.

2.6 Numerical Issues and Computation for Variational Gaussian Process Regression

We have implicitly assumed that variational Gaussian process regression can be implemented in both an efficient and numerically accurate way. In practice, most existing implementations of variational Gaussian process regression are less numerically stable than implementations of Gaussian process regression, leading to issues both in practical performance and in assessing the quality of inference. We now discuss the details of the computation done in the variational version of workflow 2.

2.6.1 Computation of the Evidence Lower Bound

An application of Woodbury's lemma (proposition A.16) and the matrix determinant lemma (proposition A.17) allows us to rewrite the evidence lower bound (eq. 2.52) as,

$$\begin{aligned}\underline{\mathcal{L}}(\mathbf{z},\theta) = C - \frac{N}{2}\log(\sigma^2) - \frac{1}{2}\log\det(\mathbf{I}+\frac{1}{\sigma^2}\mathbf{A}\mathbf{A}^\top) - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}\right) + \frac{1}{2}\mathrm{tr}\left(\mathbf{A}^\top\mathbf{A}\right) \\ - \frac{\|\mathbf{y}\|^2}{2\sigma^2} + \frac{1}{2\sigma^2}\mathbf{y}^\top\mathbf{A}(\mathbf{I}+\frac{1}{\sigma^2}\mathbf{A}\mathbf{A}^\top)^{-1}\mathbf{A}^\top\mathbf{y} \end{aligned} \tag{2.103}$$

where $\mathbf{A} = \mathbf{L}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^\top$, $\mathbf{L}\mathbf{L}^\top = \mathbf{K}_{\mathbf{z},\mathbf{z}}$ and $C = -\frac{N}{2}\log 2\pi$. Equation (2.103) can be implemented in $O(NM^2)$ via Cholesky decomposition of $\mathbf{K}_{\mathbf{z},\mathbf{z}}$, a triangular back-solve to compute $\mathbf{A}$, matrix-matrix multiplication to compute $\mathbf{A}\mathbf{A}^\top$, an additional Cholesky decomposition of $(\mathbf{I}+\frac{1}{\sigma^2}\mathbf{A}\mathbf{A}^\top)^{-1}$, and a series of matrix-vector operations that have a small computation cost. Similar considerations apply to proposition 2.18.

Where the matrix $\mathbf{K}$ that must be inverted in Gaussian process regression is generally reasonably well-conditioned as its smallest eigenvalue is bounded below by σ^2, the matrix $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ can be arbitrarily poorly conditioned. This can lead to numerical errors and potential failure of the Cholesky decomposition.

Jitter and the Evidence Lower Bound A common numerical 'trick' to avoid failures in the Cholesky decomposition consists of replacing $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ with $\mathbf{K}_{\mathbf{z},\mathbf{z}} + \varepsilon\mathbf{I}$ for a small ε (e.g. `1e-6`) in the computation of eq. (2.52), which improves the condition number of $\mathbf{K}_{\mathbf{z},\mathbf{z}}$, alleviating issues with the Cholesky decomposition failing during optimization of the evidence lower bound. Since $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ may be very poorly conditioned this strategy should be approached with caution as small perturbations to $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ can lead to large changes in solutions to systems of equations involving $\mathbf{K}_{\mathbf{z},\mathbf{z}}$.

Proposition 2.19 (Effect of jitter on upper and lower bounds)**.** *Let $\underline{\mathcal{L}}_\varepsilon(\mathbf{z},\theta)$ denote the evidence lower bound computed with jitter $\varepsilon \geq 0$ added to $\mathbf{K}_{\mathbf{z},\mathbf{z}}$, that is*

$$\underline{\mathcal{L}}_\varepsilon(\mathbf{z},\theta) = -C - \frac{1}{2}\log\det(\mathbf{Q}_{\mathbf{x},\mathbf{x}}(\varepsilon)+\sigma^2\mathbf{I}) - \frac{1}{2}\mathbf{y}^\top(\mathbf{Q}_{\mathbf{x},\mathbf{x}}(\varepsilon)+\sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{2.104}$$

$$- \frac{1}{2\sigma^2}\mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}(\varepsilon)\right) \tag{2.105}$$

with $\mathbf{Q}_{\mathbf{x},\mathbf{x}}(\varepsilon) = \mathbf{K}_{\mathbf{x},\mathbf{z}}(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \varepsilon\mathbf{I})^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^{\top}$ *and* $C = -\frac{N}{2}\log 2\pi$. *Then* $\underline{\mathcal{L}}_{\varepsilon}(\mathbf{z},\theta)$ *is monotonically decreasing in* ε. *If* $\overline{D}_{\varepsilon}(\mathbf{z},\theta)$ *denotes the upper bound on the Kullback-Leibler divergence then* $\overline{D}_{\varepsilon}(\mathbf{z},\theta)$ *is monotonically increasing in* ε.

Proof. Upon observing that for $\varepsilon' \geq \varepsilon \geq 0$, we have $\mathbf{Q}_{\mathbf{x},\mathbf{x}}(\varepsilon') \preceq \mathbf{Q}_{\mathbf{x},\mathbf{x}}(\varepsilon)$ (see Horn and Johnson 2012, Theorem 7.2.2, a) both statements follow from properties of the Loewner order on semi-definite matrices (see proposition A.19). □

Titsias (2008) argued for the use of proposition 2.19 during evidence lower bound maximization on the basis of performing variational Gaussian process regression in an augmented model. Under this interpretation, ε is a variational parameter, whose optimal value is 0. As Titsias (2008) observed and proposition 2.19 shows, during optimization $\varepsilon \to 0$. This results in numerical instability unless a lower bound is enforced on ε. We therefore take the pragmatic perspective that ε is not a variational parameter, and should be taken as close to 0 as possible without the Cholesky decomposition failing. Taking even relatively small ε can have a noticeable impact on the evidence lower bound: several nats is not uncommon depending on the hyperparameter settings. This presents a practical issue in evaluating the diagnostics discussed in the section 2.4, which often require $\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) \ll 1$ to lead to useful conclusions.

Chapter 3

Convergence Properties of Variational Gaussian Process Regression

When introducing Gaussian process approximations in chapter 1, one of the central questions we asked was: *Will it work*? A more precise formulation of this question in the context of sparse variational Gaussian process regression is: *How many inducing points are needed to accurately approximate the posterior?* In this chapter we prove that for large datasets under relatively standard assumptions, many fewer inducing points are needed than there are datapoints to accurately approximate the posterior.

Motivating Examples We give three toy examples to illustrate that any reasonable answer to the question of how many inducing points are needed depends both on the model (prior and likelihood) and dataset considered.

Example 3.1. *Consider the case of a squared exponential kernel with lengthscale* $\frac{1}{\sqrt{2}}$ *and variance* 1. *Suppose* $x_n = n$ *for* $1 \leq n \leq N$, *which might occur when studying time series. The resulting kernel matrix is*

$$\mathbf{K}_{\mathbf{x},\mathbf{x}} = \begin{pmatrix} 1 & e^{-1} & \dots & e^{-N^2} \\ e^{-1} & 1 & \dots & e^{-(N-1)^2} \\ \vdots & \ddots & \ddots & \vdots \\ e^{-N^2} & e^{-(N-1)^2} & \dots & 1 \end{pmatrix}. \tag{3.1}$$

In this model, we cannot expect sparse variational inference to work unless $N \approx M$, *because each observation contains new information that cannot be compactly summarized by knowing what the posterior function does at a fixed collection of points (since the domain keeps increasing in size). More formally, the matrix in eq.* (3.1) *is diagonally dominant. By Gershgorin's circle theorem (Horn and Johnson, 2012, Theorem 6.1.1), the smallest eigenvalue of* $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ *is lower bounded by* $1 - 2\sum_{n=1}^{\infty} e^{-n^2} > 0.21$. *No approximation based on low-rank methods, such as* $\mathbf{Q}_{\mathbf{x},\mathbf{x}} \approx \mathbf{K}_{\mathbf{x},\mathbf{x}}$, *can be accurate since* $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ *is not approximately low-rank.*

Example 3.2. *Consider the same dataset as in the previous example, but with the linear kernel, $k(x,x') = xx'$. The resulting covariance matrix is*

$$\mathbf{K}_{\mathbf{x},\mathbf{x}} = \begin{pmatrix} 1 & 2 & \cdots & N \\ 2 & 4 & \cdots & 2N \\ \vdots & \ddots & \ddots & \vdots \\ N & 2N & \dots & N^2 \end{pmatrix} = \mathbf{v}\mathbf{v}^\top, \tag{3.2}$$

with $\mathbf{v} = [1,2,\cdots,N]^\top$. If we place a single inducing point at any point $x \in \mathbb{R}$, $x \neq 0$, then we recover exact inference in this model. This occurs since the function values at all non-zero inputs are perfectly correlated. Note also that $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ has rank 1.

Example 3.3. *Suppose that $x_1 = x_2 = \cdots = x_N = x$. Then for any kernel k*

$$\mathbf{K}_{\mathbf{x},\mathbf{x}} = k(x,x) \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \ddots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} = k(x,x)\mathbf{1}\mathbf{1}^\top. \tag{3.3}$$

If we place a single inducing point at x, we will recover the exact posterior.

In general, we do not expect a single covariate will be repeated many times, as in example 3.3, but approximately the same behavior occurs when many covariates are very close. From these examples, we distill the following explanation for when we should expect variational Gaussian process regression to succeed: *variational Gaussian process regression can be successful with many fewer inducing points than datapoints if 1. The model is simple, or 2. The data is very concentrated in a small region.*[1]

Structure of Chapter The entire chapter seeks to address the question: *How many inducing points are needed to accurately approximate the posterior?* The first three sections focus on slight variations of this question:

- *How large can the Kullback-Leibler divergence to the posterior be, and how does this depend on the distribution of the data and how we place inducing points (section 3.1)?*
- *How many inducing points suffice for commonly used kernels (cf. section 1.2.1) to make upper bounds on the Kullback-Leibler divergence small (section 3.2)?*
- *How many inducing points are needed so that the Kullback-Leibler divergence is not necessarily large (section 3.3)?*

[1] If the kernel is non-stationary or $\mathcal{X}$ is not a metric space, we can replace 'concentrated' with 'correlated under the prior', in which case the two conditions are somewhat redundant. From the Hilbert space view discussed in section 2.3, we can think of this as requiring there to exist a low-dimensional subspace to play the role of $\mathcal{H}_\mathbf{z}$ such that projecting from $\mathcal{H}_\mathbf{x}$ onto this space does not lose much information.

In summary, the first two sections consider upper bounds, while the third consider lower bounds.

Section 3.4 discusses related work, both prior and subsequent to the publication of the work discussed in this chapter. Section 3.5 gives a summary of the results presented and discusses direction for future research motivated by the results presented.

Assumptions for Results Before delving into the proof of results, we clarify the assumptions we will make about the model and data-generating process.

Assumption 1. *The covariates x_n are independently and identically distributed according to ρ', a probability measure on $\mathcal{X}$.*

Assumption 2. *There exists a measure ρ and $C_{\rho,\rho'} \geq 1$ such that $\sup_{x\in\mathcal{X}} r'(x)/r(x) < C_{\rho,\rho'}$, where r is the density of ρ and r' is the density of ρ' from assumption 1 such that the kernel k satisfies Mercer's theorem with respect to ρ on the diagonal (theorem 1.1).*[2] *In particular, $k(x,x) = \sum_{m=1}^{\infty} \lambda_m \phi_m(x)^2$, ρ-almost everywhere, with $\sum_{m=1}^{\infty} \lambda_m < \infty$ and $\{\phi_m\}_{m=1}^{\infty}$ orthonormal in $L^2(\mathcal{X},\rho)$.*

Assumption 3.a. *There exists a $C_y > 0$ such that $\mathbb{E}[\|\mathbf{y}\|^2|\mathbf{x}] \leq C_y N$ for all $N \in \mathbb{N}$.*

Assumption 1 excludes cases like example 3.1 where the spread of the data increases over time, as the x_n are not identically distributed in that case. Assumption 2 allows us to make asymptotic statements: the bounds derived will depend on the operator eigenvalues λ_m. However, as an intermediate step we prove bounds depending on the empirical eigenvalues of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$, $\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})$, that hold without this assumption. Assumption 3.a is relatively mild, and is more general than the assumption that response variables are uniformly bounded. We will also derive bounds under the more restrictive assumption,

Assumption 3.b. *For all $N \in \mathbb{N}$, $\mathbf{y}|\mathbf{x} \sim \mathcal{N}(0,\mathbf{K})$.*

Assumption 3.b assumes the model (eq. 1.3) corresponds to the actual data-generating process.

Bibliographic Notes A preliminary version of the results in this chapter were presented in Burt et al. (2019), a version similar in scope to what is presented here was given in Burt et al. (2020b). The basic proof strategy for the upper bounds was presented in the Master of Philosophy dissertation Burt (2018, Section 4) and a result was given for inducing features defined with respect to the Mercer decomposition of the kernel, assuming inducing inputs are initialized with an impractically expensive scheme, or assuming access to oracle optimization of inducing inputs. The primary contributions over work in Burt (2018) are: the proof structure has been clarified; the average case analysis under assumption 3.b is new; results on upper bounds based on fast methods for initializing inducing points has been added, which allows for reasonably practical algorithms to be run with the claimed guarantees; the addition of lower bounds to the analysis (section 3.3).

This work was done in collaboration with Carl E. Rasmussen and Mark van der Wilk.

[2] More generally we require $\rho' \ll \rho$ and $\|\frac{d\rho'}{d\rho}\|_{L^\infty(\mathcal{X},\rho)} < C_{\rho,\rho'}$. The proof immediately generalizes, and we avoid this technicality only to limit the measure-theoretic overhead.

3.1 Upper bounds on the Kullback-Leibler Divergence to the Posterior

In this section, we prove several related bounds. They have a form similar to the following bound:

Theorem 3.4 (Form of Upper Bounds on Kullback-Leibler Divergence, Expectation)**.** *Suppose assumption 1, assumption 2 and either assumption 3.a or assumption 3.b. Select M inducing points from the training data according to some algorithm that can be computed in $O(N\text{polylog}(N)\text{poly}(M))$. Then,*

$$\mathbb{E}[\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)] \leq U(N, M, C_{\rho,\rho'}, \Lambda_{h(M)}), \tag{3.4}$$

where h is (essentially) linear in M and $\Lambda_{h(M)} = \sum_{m=h(M)+1}^{\infty} \lambda_m$, with λ_m the eigenvalues of $T_{k,\rho}$.

Alternatively, we show results of the form,

Theorem 3.5 (Form of Upper Bounds on Kullback-Leibler Divergence, Fixed Probability)**.** *Suppose assumption 1, assumption 2 and either assumption 3.a or assumption 3.b. Select M inducing points from the training data according to some algorithm that can be computed in $O(N\text{polylog}(N)\text{poly}(M))$. Then, with probability $1-\delta$*

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq U(N, M, C_{\rho,\rho'}, \Lambda_{h(M)}, \delta), \tag{3.5}$$

where h is (essentially) linear in M and $\Lambda_{h(M)} = \sum_{m=h(M)+1}^{\infty} \lambda_m$, with λ_m the eigenvalues of $T_{k,\rho}$.

The upper bound U increases linearly or quadratically in N, its first argument, will increase linearly or be constant in its second argument and will be linear in its third argument. The reason such bounds are useful is that the third argument is itself a function of M, and is rapidly decreasing in M if the model is sufficiently simple (see the discussion regarding approximate dimensionality and Mercer's theorem in section 1.2.1).

The proofs of upper bounds on the Kullback-Leibler divergence of the form in theorem 3.4 and theorem 3.5 follow three steps. First, we prove a simplified version of the a posteriori bound on the Kullback-Leibler divergence to the posterior (proposition 2.18). The bound is strictly weaker than the one already proven, but easier to analyze. Second, we discuss methods for selecting inducing points $\mathbf{z}$. In order to obtain bounds, we need the method for selecting inducing points to lead to a bound on $\text{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)$ and $\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\text{op}}$. We leverage existing literature on column-sampling for the Nyström approximation to this end. The resulting bounds depend on the eigenvalues of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. The final step of the proof relates these back to the eigenvalues of $T_{k,\rho}$ using a minor generalization of a lemma from Shawe-Taylor et al. (2005) presented in Burt (2018).

3.1.1 Simplified A Posteriori Upper Bounds on the Kullback-Leibler Divergence

The starting point for our analysis will be an upper bound on the Kullback-Leibler divergence. This bound will again rely on matrix properties, and we refer the reader to table 1 in the preamble for a summary of notation used and appendix A for a review of matrix properties used.

Proposition 3.6. *For any $M, N \geq 0$, $\mathbf{z} \in \mathcal{X}^M, \mathbf{x} \in \mathcal{X}^N, \mathbf{y} \in \mathbb{R}^N$ for Q a Gaussian process with mean and covariance eq. (2.48),*

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{1}{2\sigma^2}\left(T + \frac{\zeta \|\mathbf{y}\|^2}{\zeta + \sigma^2}\right) \leq \frac{1}{2\sigma^2}\left(T + \frac{T\|\mathbf{y}\|^2}{T + \sigma^2}\right) \tag{3.6}$$

with $\zeta = \|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\text{op}}$ and $T = \text{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)$.

Proof. The proof is a simplified version of the proof of proposition 2.18 after noting that $\mathbf{Q} \prec \mathbf{K} \prec \mathbf{Q} + \zeta\mathbf{I} \prec \mathbf{Q} + T\mathbf{I}$, and applying the definition of operator norm (definition A.6) to bound the quadratic term in $\mathbf{y}$ that arises. The second upper bound was shown in Burt (2018, Lemma 4.1.2). □

The term involving $\|\mathbf{y}\|^2$ will be generally the largest term in this bound, and may be quite pessimistic for many datasets. With assumption 3.b, we arrive at a sharper upper bound on the expected Kullback-Leibler divergence. Before stating this bound, we prove the following lemma, which is a special case of a Hölder inequality for Schatten norms (see Tao 2012, Exercise 1.3.26).

Lemma 3.7. *Let $\mathbf{A} \in S_+^N$ and $\mathbf{B} \in \mathbb{R}^{N\times N}$. Then,* $\text{tr}\left(\mathbf{AB}\right) \leq \|\mathbf{B}\|_{\text{op}}\text{tr}\left(\mathbf{A}\right)$.

Proof. We follow the proof given in Bach (2022). As $\mathbf{A} \in S_+^N$, we can eigendecompose $\mathbf{A}$ and write $\mathbf{A} = \sum_{n=1}^N \lambda_n(\mathbf{A})\mathbf{u}_n\mathbf{u}_n^\top$ with $\lambda_n(\mathbf{A}) \geq 0$ and $\|\mathbf{u}_n\|_2 = 1$. Then,

$$\text{tr}\left(\mathbf{AB}\right) = \sum_{n=1}^N \lambda_n(\mathbf{A})\text{tr}\left(\mathbf{u}_n\mathbf{u}_n^\top\mathbf{B}\right) = \sum_{n=1}^N \lambda_n(\mathbf{A})\mathbf{u}_n^\top\mathbf{B}\mathbf{u}_n. \tag{3.7}$$

In the second equality we use the cyclic property of trace proposition A.10. For each n, we have,

$$\mathbf{u}_n^\top\mathbf{B}\mathbf{u}_n \leq \|\mathbf{B}\|_{\text{op}}\|\mathbf{u}_n\|^2 = \|\mathbf{B}\|_{\text{op}}, \tag{3.8}$$

which together with the non-negativity of the $\lambda_n(\mathbf{A})$ completes the proof. □

We will additionally require the formula for the Kullback-Leibler divergence between multivariate Gaussian measures.

Proposition 3.8 (Kullback-Leibler Divergence, Multivariate Gaussian)**.** *Let $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathbb{R}^N$ and $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in S_{++}^N$. Then*

$$\mathfrak{D}_{KL}\left(\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1), \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)\right) = \frac{1}{2}\left(\text{tr}\left(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\right) + \log\frac{\det\boldsymbol{\Sigma}_2}{\det\boldsymbol{\Sigma}_1} + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top\boldsymbol{\Sigma}_2^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) - N\right). \tag{3.9}$$

To prove the desired upper bound, we now recognize that for any fixed $\mathbf{y}$ the Kullback-Leibler divergence to the posterior arising in sparse variational Gaussian process regression closely resembles a density ratio. Under assumption 3.b the average Kullback-Leibler divergence is therefore itself related to a Kullback-Leibler divergence between finite dimensional Gaussian distributions.

Proposition 3.9. *With the same setup as proposition 3.6, but additionally assuming assumption 3.b and that* $\mathbf{z}|\mathbf{x}$ *is independent of* $\mathbf{y}|\mathbf{x}$*,*

$$\frac{T}{2\sigma^2} \leq \mathbb{E}[\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{z},\mathbf{x}] \leq \frac{T}{\sigma^2}. \tag{3.10}$$

Proof. Combining eq. (1.38) and eq. (2.10)

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) = \mathcal{L}(\theta) - \underline{\mathcal{L}}(\mathbf{z},\theta) = \frac{T}{2\sigma^2} + \log\frac{\mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{K})}{\mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{Q})}. \tag{3.11}$$

Taking conditional expectations on both sides and recognizing a Kullback-Leibler divergence on the right-hand side,

$$\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{z},\mathbf{x}] = \frac{T}{2\sigma^2} + \mathfrak{D}_{\mathrm{KL}}\left(\mathcal{N}(\mathbf{0},\mathbf{K}), \mathcal{N}(\mathbf{0},\mathbf{Q})\right). \tag{3.12}$$

The claimed lower bound follows from non-negativity of the Kullback-Leibler divergence. From proposition 3.8

$$\begin{aligned}
\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{z},\mathbf{x}] &= \frac{T}{2\sigma^2} - \frac{N}{2} + \frac{1}{2}\log\frac{\det\mathbf{Q}}{\det\mathbf{K}} + \frac{1}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}\mathbf{K}\right) && (3.13)\\
&\leq \frac{T}{2\sigma^2} - \frac{N}{2} + \mathrm{tr}\left(\mathbf{Q}^{-1}\mathbf{K}\right) && (3.14)\\
&= \frac{T}{2\sigma^2} + \frac{1}{2}\mathrm{tr}\left(\mathbf{Q}^{-1}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})\right). && (3.15)
\end{aligned}$$

Apply lemma 3.7 using that $\|\mathbf{Q}^{-1}\|_{\mathrm{op}} \leq \frac{1}{\sigma^2}$ to conclude $\mathrm{tr}\left(\mathbf{Q}^{-1}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})\right) \leq \frac{T}{\sigma^2}$. □

From propositions 3.6 and 3.9, in order to upper bound the Kullback-Leibler divergence to the posterior, it suffices to upper bound $\mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)$. Bounds on this quantity depend on the method for selecting inducing points, as well as the location of covariates. In the next section, we address the problem of selecting inducing points.

3.1.2 Selecting Inducing Points

Inducing point selection has been widely-studied in the context of kernel methods where a Nyström approximation is leveraged, including sparse Gaussian process regression and kernel ridge regression. Kernel ridge regression is equivalent to prediction with the Gaussian process mean, and we therefore expect methods that work well for kernel ridge regression to also apply to Gaussian process regression with minimal modifications (see Wild and Wynne, 2021 for a comparison between sparse variational Gaussian process regression and Nyström based kernel ridge regression). One of the most commonly used methods in practice for selecting inducing inputs consists of sampling a subset of cardinality M uniformly at random from the covariates without replacement. Bounds on the quality of the resulting matrix approximation, and the downstream kernel ridge regression predictor have been studied in this

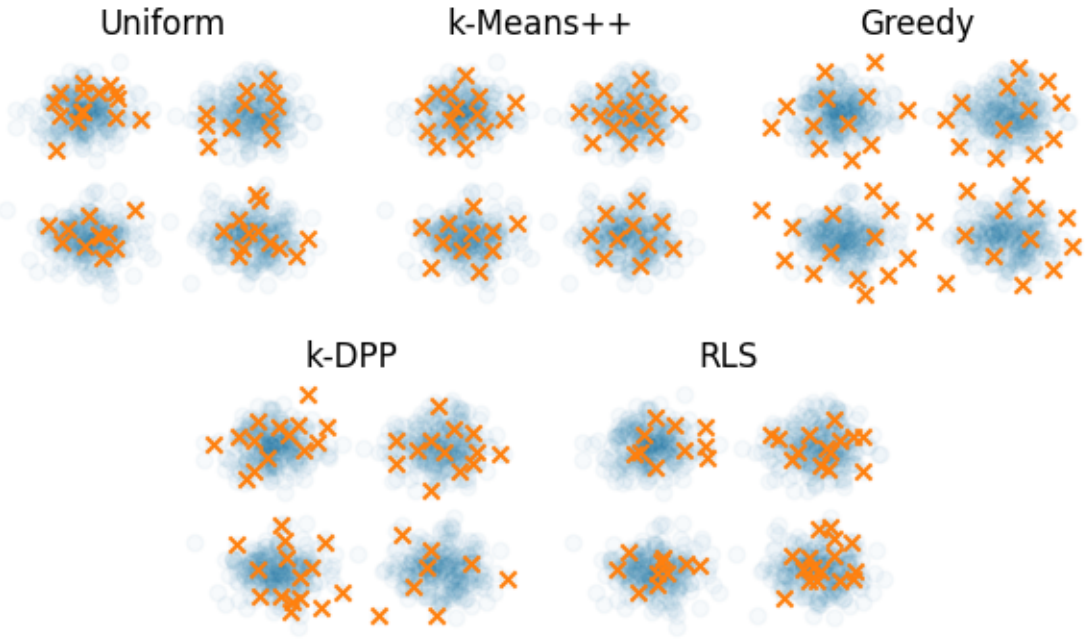


Fig. 3.1 A comparison of several methods for initializing inducing points. There are $N = 1000$ covariates (blue circles) drawn from a Gaussian mixture model with 4 clusters and $M = 50$ inducing points (orange $\times's$). From top left to bottom right inducing points are selected: uniformly as a subset of fixed cardinality from the covariates; as the centers from k-means++ run on the covariates; following algorithm 1; following algorithm 2 running 50000 iterations of MCMC to approximate M-DPP sampling (we use the standard terminology k-DPP in the figure, but in the sequel use M-DPP to avoid confusion with the kernel); using the recursive approximation to leverage score sampling from Musco and Musco (2017, Algorithm 3). Generally, it is beneficial to have inducing inputs be over-dispersed relative to a uniform subset of the covariates.

case (Bach, 2013; Gittens and Mahoney, 2016) and depend heavily on assumptions about the covariate distribution and resulting kernel matrix.

Two methods from the kernel ridge regression literature generally result in sharper upper bounds and are therefore of particular interest to us: sampling from an *M-determinantal point process* (*M*-DPP) (Li et al., 2016a),[3] and *ridge leverage score* (RLS) sampling (Alaoui and Mahoney, 2015; Calandriello et al., 2017; Rudi et al., 2015). Calandriello et al. (2019) used ridge leverage scores to show convergence of the approximation eq. (2.18) and eq. (2.19) to the corresponding posterior moments. We give a longer comparison between the bounds presented in this work and in Calandriello et al. (2019) in section 3.4.

Minimizing the Trace and Operator Norm Error

Before discussing placement of inducing points in detail, we consider the set of inducing points that minimize the upper bounds in propositions 3.6 and 3.9. Reasoning about the best-case inducing points with respect to these bounds is greatly simplified by a well-known characterization of optimal fixed-rank matrix approximation.

A matrix norm, $\|\cdot\|$, is called *unitarily invariant* if for any matrix $\mathbf{A}$ and any orthogonal matrices $\mathbf{U}$, and $\mathbf{V}$, $\|\mathbf{A}\| = \|\mathbf{U}\mathbf{A}\mathbf{V}^\top\|$. From the existence of the singular value decomposition, and since the operator norm is equal to the largest singular value, the operator norm is unitarily invariant. While the trace is not a norm on matrices, for positive definite matrices, the trace coincides with the *Schatten 1-norm*, which is the ℓ^1-norm of the singular values of the matrix. Since this norm depends only on the singular values of the matrix, it is also unitarily invariant.

Theorem 3.10 (Eckart-Young-Mirsky Theorem, Eckart and Young, 1936)**.** *Let $\|\cdot\|$ be a unitarily invariant norm on $\mathbb{R}^{N\times K}$. Suppose $\mathbf{A}$ has singular value decomposition, $\mathbf{A} = \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \mathbf{v}_r^\top$ with $\sigma_{r+1} \geq \sigma_r > 0$ for all r. If $M \leq R$ then the matrix $\mathbf{A}^{(\mathrm{M})} = \sum_{r=1}^{M} \sigma_r \mathbf{u}_r \mathbf{v}_r^\top$ satisfies*

$$\|\mathbf{A} - \mathbf{A}^{(\mathrm{M})}\| \leq \|\mathbf{A} - \mathbf{B}\| \quad \text{for all } \mathbf{B} \in \mathbb{R}^{N\times K} \text{ with rank at most } M. \tag{3.16}$$

In words, this means that the truncated-singular value decomposition leads to the optimal fixed-rank approximation of a matrix with respect to any unitarily invariant norm. For symmetric positive semi-definite matrices, as the singular value decomposition and eigendecomposition coincide, this means a truncated eigendecomposition is optimal. See Li and Strang (2020) for an elementary proof of theorem 3.10.

Motivated by this result, we consider the eigendecomposition[4] $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = \mathbf{K}_{\mathbf{x},\mathbf{x}}$, where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ is an $N \times N$ orthogonal matrix whose columns are the eigenvectors of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ and

$$\mathbf{\Lambda} = \operatorname{diag}(\lambda_1(\mathbf{K}_{\mathbf{x},\mathbf{x}}), \ldots, \lambda_N(\mathbf{K}_{\mathbf{x},\mathbf{x}})) \tag{3.17}$$

[3] The standard terminology is k-DPP. We use M as this determines the number of inducing points and to avoid confusion with the kernel function.

[4] Because $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ is symmetric, positive definite, this is the same as the singular value decomposition

is a diagonal matrix of eigenvalues of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ sorted in descending order. Define $\mathbf{K}_{\mathbf{x},\mathbf{x}}^{(\mathrm{M})} = \mathbf{U}^{(\mathrm{M})}\boldsymbol{\Lambda}^{(\mathrm{M})}\mathbf{U}^{(M)\top}$, where $\mathbf{U}^{(\mathrm{M})}$ is an $N \times M$ matrix containing the first M columns of $\mathbf{U}$ and $\boldsymbol{\Lambda}^{(\mathrm{M})}$ is an $M \times M$ diagonal matrix with entries, $\lambda_1(\mathbf{K}_{\mathbf{x},\mathbf{x}}), \ldots, \lambda_M(\mathbf{K}_{\mathbf{x},\mathbf{x}})$, i.e. $\mathbf{K}_{\mathbf{x},\mathbf{x}}^{(\mathrm{M})}$ is the rank-M truncated eigendecomposition of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. As any subset of M inducing variables will lead to a rank-M matrix $\mathbf{Q}_{\mathbf{x},\mathbf{x}} \prec \mathbf{K}_{\mathbf{x},\mathbf{x}}$ by positive semi-definiteness of the kernel and proposition A.15, theorem 3.10 implies that

$$\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\mathrm{op}} \geq \|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{K}_{\mathbf{x},\mathbf{x}}^{(\mathrm{M})}\|_{\mathrm{op}} = \lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}}), \quad \text{and} \tag{3.18}$$

$$\mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right) \geq \mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{K}_{\mathbf{x},\mathbf{x}}^{(\mathrm{M})}\right) = \sum_{m=M+1}^{N} \lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}). \tag{3.19}$$

We construct a set of inducing points that achieves the lower bounds in eq. (3.18) and eq. (3.19).

Construction of Eigenvector Inducing Points Consider the random variable defined as linear combinations of the random variables associated to evaluating the latent function at each observed input location, with weights coming from the m^{th} eigenvector of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$, i.e.

$$f_m = \frac{1}{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})} \sum_{n=1}^{N} u_{n,m} f(x_n) \in \mathcal{H}. \tag{3.20}$$

Ferrari-Trecate et al. (1999) considered the use of these features to define a finite-dimensional linear regression model that approximates a Gaussian process model. Calculating the covariance between inducing points,

$$\mathbb{E}[f_m f_{m'}] = \frac{1}{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})\lambda_{m'}(\mathbf{K}_{\mathbf{x},\mathbf{x}})} \sum_{n=1}^{N}\sum_{n'=1}^{N} u_{m,n}u_{m',n'}\mathbb{E}[f(x_n)f(x_{n'})] \tag{3.21}$$

$$= \frac{1}{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})\lambda_{m'}(\mathbf{K}_{\mathbf{x},\mathbf{x}})} \sum_{n=1}^{N}\sum_{n'=1}^{N} u_{m,n}u_{m',n'}k(x_n, x'_n). \tag{3.22}$$

The final two sums are the quadratic form, $\mathbf{u}_m^\top \mathbf{K}_{\mathbf{x},\mathbf{x}}\mathbf{u}_{m'}$. As $\mathbf{u}_m$ is an eigenvector of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ and $\mathbf{u}_m$ is orthogonal to $\mathbf{u}'_m$ unless $m = m'$, this simplifies to

$$\mathbb{E}[f_m f_{m'}] = \begin{cases} 1/\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) & m = m' \\ 0 & \text{otherwise.} \end{cases} \tag{3.23}$$

A similar calculation shows

$$\mathbb{E}[f_m f(x_n)] = \mathbf{u}_{m,n}. \tag{3.24}$$

From eq. (3.23) $\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1} = \boldsymbol{\Lambda}^{(\mathrm{M})}$ and from eq. (3.24) $\mathbf{K}_{\mathbf{x},\mathbf{z}} = \mathbf{U}^{(\mathrm{M})}$ for these features. Therefore, $\mathbf{Q}_{\mathbf{x},\mathbf{x}} = \mathbf{K}_{\mathbf{x},\mathbf{x}}^{(\mathrm{M})}$. By theorem 3.10 these inducing points minimize the upper bounds in propositions 3.6 and 3.9 among all sets of inducing points with cardinality M. Substituting the resulting bounds into

proposition 3.6

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{1}{2\sigma^2}\left(\sum_{m=M+1}^{\infty} \lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \frac{\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}})\|\mathbf{y}\|^2}{\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \sigma^2}\right), \tag{3.25}$$

which begins to resemble the form of theorem 3.4 up to replacing matrix eigenvalues with operator eigenvalues. Unfortunately, computing the matrices $\mathbf{K}_{\mathbf{x},\mathbf{z}}$ and $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ in this case involves computing the first M eigenvalues and eigenvectors of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$, which in many instances has a computational cost comparable to directly computing properties of the posterior, and scales at least quadratically in N. Zhu et al. (1997) and Burt (2018) considered an operator-theoretic analogue of these features that uses the eigenvalues and eigenfunctions of $T_{k,\rho}$ to define inducing points. Generally, the covariance between these features and latent function values at the data cannot be computed in closed-form and these features are therefore of limited applicability.

M-Determinantal Point Processes

We return to more practical methods for inducing point selection. In order to derive non-trivial upper bounds on $\mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)$ and $\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\mathrm{op}}$, we need a sufficiently good method for placing inducing points. When using differentiable kernel functions, many practitioners select the locations of the inducing points with gradient-based methods by maximizing the evidence lower bound (eq. 2.52). As this is a high-dimensional, non-convex optimization algorithm, directly analyzing the result of this procedure seems a challenging task. Instead, in this section we assume M inducing inputs are sampled from the covariates according to an approximate *M-determinantal point process* (M-DPP) (Kulesza and Taskar, 2011) and use known bounds on $\mathbb{E}[\mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)|\mathbf{x}]$ for this sampling method (Belabbas and Wolfe, 2009). If this scheme is used as an initialization prior to a gradient-based optimization of the evidence lower bound with respect to the inducing inputs with the hyperparameters fixed, the bounds on the Kullback-Leibler divergence still apply so long as the optimizer does not decrease the evidence lower bound.

Given a matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}} \in S_+^N$, an M-determinantal point process (Kulesza and Taskar, 2011) with kernel matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ defines a discrete probability distribution over subsets of the N columns of S_+^N, with positive probability only assigned to subsets of cardinality M. The probability of any subset of cardinality M is proportional to the determinant of the principal sub-matrix formed by selecting those columns and the corresponding rows, that is for any set $\mathbf{z}$ of M columns of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$[5]

$$\Pr(\mathbf{z}) = \frac{\det(\mathbf{K}_{\mathbf{z},\mathbf{z}})}{\sum_{|\mathbf{z}'|=M}\det(\mathbf{K}_{\mathbf{z}',\mathbf{z}'})}, \tag{3.26}$$

[5]For discussion of M determinantal point process, we conflate the indices of columns of the matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ and data points x_n via the map $n \to x_n$.

where $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ is the principal sub-matrix of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ with columns formed by $\mathbf{z}$. For a thorough introduction to determinantal point processes, as well as an implementation of many sampling methods, see Gautier et al. (2019).

The determinant of $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ corresponds to the volume of the parallelepiped in $\mathbb{R}^M$ formed by the columns in $\mathbf{z}$, so M-determinantal point processes introduce strong negative correlations between points sampled: sampling two points close together would make the volume of this parallelepiped collapse. This leads to samples that are more dispersed than subsets selected uniformly from the covariates (see figure 3.1). The tendency to select points that are less correlated means that it is likely a representative subset of points will be selected, and suffices to establish bounds on the error of a Nyström approximation.

Lemma 3.11 (Belabbas and Wolfe, 2009, Theorem 1)**.** *Let $\mathbf{K}_{\mathbf{x},\mathbf{x}} \in S_+^N$ with eigenvalues $\lambda_1(\mathbf{K}_{\mathbf{x},\mathbf{x}}) \geq \cdots \geq \lambda_N(\mathbf{K}_{\mathbf{x},\mathbf{x}}) \geq 0$. Suppose a set of points is sampled according to an M-determinantal point process with kernel matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. Define the (random) matrix $\mathbf{Q}_{\mathbf{x},\mathbf{x}} = \mathbf{K}_{\mathbf{x},\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^{\top}$ where $\mathbf{K}_{\mathbf{z},\mathbf{z}}$ is the $M \times M$ principal sub-matrix of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ with columns in $\mathbf{z}$ and $\mathbf{K}_{\mathbf{x},\mathbf{z}}$ is the $N \times M$ matrix with rows $\mathbf{z}$. Then,*

$$\mathbb{E}[\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}) \,|\mathbf{x}] \leq (M+1) \sum_{m=M+1}^{N} \lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}). \tag{3.27}$$

Lemma 3.11 tells us that using an M-determinantal point process to choose inducing inputs makes $\mathbb{E}[\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}) \,|\mathbf{x}]$ relatively close to its optimal value of $\sum_{m=M+1}^{N} \lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})$.

Sampling from an M-Determinantal Point Process The next important question to address is whether an M-determinantal point process can be sampled from with sufficiently low computational complexity for this to be a practical method for selecting inducing points. Naively computing the probability distribution over all $\binom{N}{M}$ subsets of size M via eq. (3.26) is prohibitively expensive. Kulesza and Taskar (2011) gave an algorithm that runs in polynomial time and yields exact samples from an M-determinantal point process. Unfortunately, this algorithm involves computing an eigendecomposition of the $N \times N$ kernel matrix ($\mathbf{K}_{\mathbf{x},\mathbf{x}}$ in our case), which is computationally prohibitive as it is at least as expensive as computing the posterior and log marginal likelihood.

Dereziński et al. (2019) gave an algorithm for obtaining a sample from an M-determinantal point process in time that is polynomial in M and nearly-linear in N. However, the degree of the polynomial in M is high. We instead consider an approximate algorithm and make use of a simple corollary of lemma 3.11.

Corollary 3.12. *Let μ denote an M-determinantal point process with kernel matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. Let μ' denote a measure on subsets of columns of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ with cardinality M then*

$$\mathbb{E}_{\mu'}[\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}) \,|\mathbf{x}] \leq \mathfrak{D}_{TV}(\mu, \mu') \operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + (M+1) \sum_{m=M+1}^{N} \lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}). \tag{3.28}$$

Algorithm 1 Greedy Initialization of Inducing points

Input: Training inputs $\mathbf{x} = \{x_n\}_{n=1}^N$, number of points to choose, M, kernel k.
Returns: A set of inducing inputs of cardinality M, $\mathbf{z}_{\text{greedy}}$.
$\mathbf{z} = \emptyset$
for $1 \leq m \leq M$ **do**
  Select $x_m \in \arg\max_{x \in \mathbf{x}} k(x,x) - \mathbf{k}_{x,\mathbf{z}} \mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1} \mathbf{k}_{\mathbf{z},x}$
  $\mathbf{z} = \mathbf{z} \cup \{x_m\}$
end for
Return: $\mathbf{z}$

Algorithm 2 Markov Chain Monte Carlo algorithm for approximately sampling from an M-determinantal point process (Anari et al., 2016)

Input: Training inputs $\mathbf{x} = \{x_n\}_{n=1}^N$, number of points to choose, M, kernel k, T number of steps of MCMC to run.
Returns: An (approximate) sample from an M-determinantal point process with kernel matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ formed by evaluating k at $\mathbf{x}$.
Compute $\mathbf{z}_0$ via algorithm 1.
for $1 \leq t \leq T$ **do**
  Sample n uniformly from $\mathbf{z}_t$ and n' uniformly from $\mathbf{x} \setminus \mathbf{z}_t$. Define $\mathbf{z}' = \mathbf{z}_t \setminus \{n\} \cup \{n'\}$,
  Compute $p_{n \to n'} := \frac{1}{2} \min\{1, \det(\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}}) / \det(\mathbf{K}_{\mathbf{z}_t,\mathbf{z}_t})\}$
  With probability $p_{n \to n'}$, $\mathbf{z}_{t+1} = \mathbf{z}'$ otherwise, $\mathbf{z}_{t+1} = \mathbf{z}_t$
end for
Return: $\mathbf{z}_T$

Proof. Combine lemma 3.11, triangle inequality and eq. (2.54) using that $\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}) \in [0, \operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}})]$. □

Remark 3.13. *From the definition of the trace as the sum of the diagonal entries, assumption 1 and changing the order of summation and integration* $\mathbb{E}[\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}})] = N \mathbb{E}_{x \sim \rho'}[k(x,x)]$.

Corollary 3.12 shows that sufficiently accurate approximate sampling from an M-determinantal point process only has a small effect on the quality of the resulting approximation $\mathbf{Q}_{\mathbf{x},\mathbf{x}} \approx \mathbf{K}_{\mathbf{x},\mathbf{x}}$. High-quality approximate samples can be drawn using a simple Markov Chain Monte Carlo (MCMC) algorithm described in Anari et al. (2016) and outlined in algorithm 2. This MCMC algorithm is well-studied in the context of M-determinantal point processes and their generalizations, and is known to be rapidly mixing (Anari et al., 2016; Hermon and Salez, 2019).

Lemma 3.14 (Hermon and Salez (2019), Corollary 1)**.** *Let μ be an M-determinantal point process with $N \times N$ kernel matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. Fix $\varepsilon \in (0,1)$. Then algorithm 2 produces a sample from a distribution μ' satisfying*

$$\mathfrak{D}_{TV}(\mu, \mu') \leq \varepsilon \tag{3.29}$$

in not more than $T(\varepsilon) = 2MN\left(\log\log\left(\frac{1}{\mu(\mathbf{z}_0)}\right) + \log\frac{2}{\varepsilon^2}\right)$ *iterations, where* $\mathbf{z}_0$ *is the subset of columns at which the Markov chain is initialized.*

In other words, the MCMC algorithm mixes quickly, so long as the initial set of inducing points, $\mathbf{z}_0$ does not have extremely small prior probability under the M-determinantal point process. As total variation distance to the stationary distribution is monotonically decreasing in the number of Markov transitions (Robert and Casella, 1999, Proposition 6.52), if the chain is run for more iterations, the bound still holds. If μ is an M-determinantal point process on $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ and μ' satisfies $\mathfrak{D}_{\mathrm{TV}}(\mu,\mu') \leq \varepsilon$, we will refer to an μ' as an *ε-approximate M-determinantal point process on* $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ and a sample from μ' as an ε-approximate sample.

The per step cost of algorithm 2 is dominated by computing the acceptance ratio, which can be performed in $O(M^2)$, by iteratively updating a Cholesky or QR factorization of the matrix associated to the current set of columns. This makes the total cost of obtaining an ε-approximate sample $O\left(NM^3\log\log\left(1/\mu(\mathbf{z}_{\text{greedy}})\right) + NM^3\log 2/\varepsilon^2\right)$, where $\mathbf{z}_{\text{greedy}}$ denotes the set of columns selected by greedily maximizing the determinant of the sub-matrix.

The greedy algorithm (algorithm 1) runs in $O(NM^2)$ and is known to have an approximation ratio of not more than $1/M!^2$ to the maximum probability subset (Anari et al., 2016; Çivril and Magdon-Ismail, 2009). Because the maximum probability subset is more probable than the average probability of a subset

$$\mu(\mathbf{z}_{\text{greedy}}) \geq \left(M!^2\binom{N}{M}\right)^{-1} \geq (MN)^{-M}, \tag{3.30}$$

giving an overall complexity of not more than $O\left(NM^3(\log\log N + \log M + \log 1/\varepsilon^2)\right)$ for sampling an ε-approximate M-determinantal point process with algorithm 2. Since ε shows up inside a logarithm, we can take ε to be large (polynomial in N) while only increasing the run-time by a factor that scales logarithmically in N.

Directly using algorithm 1 for selecting $\mathbf{z}$ to form a Nyström approximation is equivalent to running an incomplete Cholesky decomposition with pivoting and has been considered in Fine and Scheinberg (2001) and Foster et al. (2009) for direct kernel approximation and as a preconditioner in Gardner et al. (2018). Inducing points selected according to algorithm 1 and algorithm 2 with a squared exponential kernel are shown in figure 3.1 (top right and bottom left respectively).

Ridge Leverage Scores

While using an M-determinantal point process for inducing inputs selection results in upper bounds on $\mathbb{E}[\mathrm{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})\,|\mathbf{x}]$, this method has a significant drawback: the computational cost of running the MCMC algorithm to obtain approximate samples dominates the cost of sparse inference. Ridge leverage score (RLS) sampling offers an alternative that runs in $O(NM^2)$, while retaining strong theoretical guarantees on the quality of the resulting approximation. In this section, we discuss ridge leverage score sampling as well an algorithm of Musco and Musco (2017) that allows for efficient approximations to

ridge leverage score sampling. Calandriello et al. (2019) applied a ridge leverage score based algorithm for the Nyström approximation in a Gaussian process bandit problem and obtained upper bounds on the error in the posterior mean and marginal variance induced by the resulting approximation.

The *ω-ridge leverage score* of a point $x_n \in \mathbf{x}$ of a Gaussian process regressor, which we denote by $\ell^{\omega}(x_n)$ is defined as $1/\omega$ times the posterior variance at x_n of the process with noise variance ω, i.e.

$$\ell^{\omega}(x_n) = \frac{1}{\omega}\left(k(x_n,x_n) - \mathbf{k}_{x_n,\mathbf{x}}\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x},x_n}\right).$$

Ridge leverage score sampling uses these scores as an importance distribution for selecting which points to include in sparse kernel methods. Intuitively, inputs at which there is high posterior uncertainty must be 'far' from other observed inputs, and therefore informative.

Computing the ridge leverage scores exactly is too computationally expensive, as it is equivalent to computing the posterior variance at each training input. However, practical approximate versions of leverage sampling algorithms that retain strong theoretical guarantees have been developed. Generally, the idea of such algorithms is to overestimate the ridge leverage scores, ensuring that the probability of inclusion of any point is at least as high as it would be if the exact leverage score was used. This comes at the cost of sampling more points in the approximation to obtain the same accuracy guarantees.

We consider the application of Algorithm 3 in Musco and Musco (2017) to the problem of selecting inducing inputs for sparse variational Gaussian process regression. This algorithm comes with the following bounds on the quality of the resulting Nyström approximation.

Lemma 3.15 (Musco and Musco 2017, Theorem 14, Appendix D)**.** *Given $\mathbf{x} \in \mathcal{X}^N$ and a kernel k, let $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ denote the $N \times N$ covariance matrix associated to $\mathbf{x}$ and k. Fix $\delta \in (0, \frac{1}{32})$ and $S \in \mathbb{N}$. There exists a universal constant c and algorithm with run time $O(NM^2)$ and memory complexity $O(NM)$ that with probability $1-3\delta$ returns $M \leq cS\log(S/\delta)$ columns of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ such that the resulting Nyström approximation, $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$, satisfies*

$$\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\mathrm{op}} \leq \frac{1}{S}\sum_{m=S+1}^{N} \lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}).$$

Remark 3.16. *For consistency of presentation with the result on M-determinantal point process initialization, we have opted to consider the 'fixed sample size' variant of the recursive ridge leverage algorithm from Musco and Musco (2017). In this variant, the ω-value in the ω-ridge leverage score which this sampling scheme approximates is determined by the choice of S. Practically it may be preferable to use Musco and Musco (2017, Algorithm 2) which is adaptive in the sense that it automatically determines the number of inducing points to achieve a specific quality of matrix approximation.*

Remark 3.17. *In section 3.1.2 M was fixed and the quality of the resulting approximation was random, in the algorithm discussed above M is additionally random, though its size is controlled in terms of S with high probability.*

3.1.3 Matrix and Operator Eigenvalues

In the previous sections, the results on the quality of approximation depended on the eigenvalues of $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ and therefore hold conditionally on the covariates $\mathbf{x}$, that is for a specific dataset. In this section, we will treat $\mathbf{x}$ as a random variable in order to determine what happens for a 'typical' dataset.

Under assumption 1 and assumption 2 with $\rho = \rho'$ and in the limit as the amount of data tends to infinity, under regularity conditions on the kernel k the matrix $\frac{1}{N}\mathbf{K}_{\mathbf{x},\mathbf{x}}$ behaves like the operator $T_{k,\rho}$ defined in (1.26) (Koltchinskii and Giné, 2000). For finite sample sizes, the large eigenvalues of $\frac{1}{N}\mathbf{K}_{\mathbf{x},\mathbf{x}}$ tend to overestimate the corresponding eigenvalues of $T_{k,\rho}$ and the small eigenvalues of $\frac{1}{N}\mathbf{K}_{\mathbf{x},\mathbf{x}}$ tend to underestimate the small eigenvalues of $T_{k,\rho}$.

Lemma 3.18 (Shawe-Taylor et al., 2005, Proposition 4, Burt, 2018, Lemma 4.1.2)**.** *Suppose assumption 1 and assumption 2 hold. Let $\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})$ denote the m^{th} largest eigenvalue of the (random) matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. Let λ_m denote the m^{th} largest eigenvalue of $T_{k,\rho}$. Then, for any $M, 1 \leq M < N$,*

$$\frac{1}{N}\mathbb{E}_{\rho'}\left[\sum_{m=M+1}^{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})\right] \leq C_{\rho,\rho'}\sum_{m=M+1}^{\infty}\lambda_m. \tag{3.31}$$

Proof. The idea is to combine Mercer's theorem and the Eckart-Young-Mirsky theorem (theorem 3.10) for the trace, together with a candidate rank-M approximation to $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. We therefore begin by expanding the kernel, along its diagonal, with respect to Mercer's theorem with measure ρ

$$[\mathbf{K}_{\mathbf{x},\mathbf{x}}]_{i,j} = \sum_{m=1}^{M}\lambda_m\phi_m(x_i)\phi_m(x_j) + \sum_{m=M+1}^{\infty}\lambda_m\phi_m(x_i)\phi_m(x_j). \tag{3.32}$$

Define the matrix

$$[\mathbf{\Phi}(\mathbf{x})]_{i,j} = \sum_{m=1}^{M}\lambda_m\phi_m(x_i)\phi_m(x_j). \tag{3.33}$$

For any $\mathbf{x}$, $\mathbf{\Phi}(\mathbf{x}) \prec \mathbf{K}_{\mathbf{x},\mathbf{x}}$, and $\mathbf{\Phi}(\mathbf{x})$ is rank-M. Hence, by the Eckart-Young-Mirsky theorem (theorem 3.10)

$$\sum_{m=M+1}^{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) = \operatorname{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{K}_{\mathbf{x},\mathbf{x}}^{(\mathrm{M})}\right) \leq \operatorname{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{\Phi}(\mathbf{x})\right). \tag{3.34}$$

Taking an expectation on the right-hand side,

$$\mathbb{E}_{\rho'}[\mathrm{tr}\,(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \boldsymbol{\Phi}(\mathbf{x}))] = \sum_{n=1}^{N}\sum_{m=M+1}^{\infty} \lambda_m \mathbb{E}_{\rho'}[\phi_m(x_n)^2] \tag{3.35}$$

$$= \sum_{n=1}^{N}\sum_{m=M+1}^{\infty} \lambda_m \mathbb{E}_{\rho}[\phi_m(x_n)^2 \tfrac{r'(x_n)}{r(x_n)}] \tag{3.36}$$

$$\leq C_{\rho,\rho'} \sum_{n=1}^{N}\sum_{m=M+1}^{\infty} \lambda_m = C_{\rho,\rho'} N \sum_{m=M+1}^{\infty} \lambda_m. \tag{3.37}$$

The expectation and sum can be interchanged in the first line by Tonelli's theorem since $\phi_m(x_n)$ is non-negative. In the second line, we have used that multiplying by the density ratio allows us to convert between expectations under ρ and ρ'. In the final line, we have used Hölder's inequality, followed by that ϕ_m is a unit vector in $L^2(\mathcal{X},\rho)$ implying ϕ_m^2 is a unit vector in $L^1(\mathcal{X},\rho)$. □

3.1.4 A Priori Bounds on the Kullback-Leibler Divergence

We now turn to bounds on the Kullback-Leibler divergence that hold either with fixed probability or in expectation. In order to prove bounds in expectation, we apply the law of iterated expectation

$$\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)] = \mathbb{E}[\mathbb{E}[\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{x},\mathbf{z}]|\mathbf{x}]]. \tag{3.38}$$

The exact statement of the result and proof of the result now depends on whether we make assumption 3.b or assumption 3.a, and whether inducing points are initialized with an approximate M-determinantal point process or ridge leverage score sampling. We state the resulting theorems in each case below, and provide two representative proofs. The other cases follow the same proof strategy with minor modifications.

Several times, we will make use of Markov's inequality,

Lemma 3.19 (Markov's Inequality)**.** *Let A be a non-negative random variable with finite expectation, then for any $\delta \in (0,1)$, with probability $1-\delta$*

$$A \leq \mathbb{E}[A]/\delta. \tag{3.39}$$

Theorem 3.20. *Suppose assumptions 1 and 2 and assumption 3.a hold. Additionally suppose*

$$\mathbb{E}_{x\sim\rho'}[k(x,x)] \leq v. \tag{3.40}$$

Sample M inducing points from the training data according to an ε-approximate M-determinantal point process with kernel matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$. Then,

$$\mathbb{E}[\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)] \leq \frac{N}{2\sigma^2}\left(1 + \frac{C_y N}{\sigma^2}\right)\left((M+1)C_{\rho,\rho'} \sum_{m=M+1}^{\infty} \lambda_m + v\varepsilon\right). \tag{3.41}$$

Corollary 3.21. *Under the assumptions of theorem 3.20, with probability at least* $1-\delta$,

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{N}{2\delta\sigma^2}\left(1+\frac{C_y N}{\sigma^2}\right)\left((M+1)C_{\rho,\rho'}\sum_{m=M+1}^{\infty}\lambda_m + v\varepsilon\right). \tag{3.42}$$

Proof of theorem 3.20 and corollary 3.21. Beginning from the law of iterated expectation, applying proposition 3.6 and assumption 3.a

$$\begin{aligned}
\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)] &= \mathbb{E}[\mathbb{E}[\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{x}, \mathbf{z}]|\mathbf{x}]] && (3.43)\\
&\leq \frac{1}{2\sigma^2}\mathbb{E}\left[\mathbb{E}\left[T + \frac{T}{T+\sigma^2}\mathbb{E}[\|\mathbf{y}\|^2|\mathbf{x}, \mathbf{z}]|\mathbf{x}\right]\right] && (3.44)\\
&\leq \frac{1}{2\sigma^2}\left(1+\frac{C_y N}{\sigma^2}\right)\mathbb{E}[\mathbb{E}[T|\mathbf{x}]] && (3.45)
\end{aligned}$$

with $T = \mathrm{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\right)$. From corollary 3.12 and lemma 3.18

$$\mathbb{E}[\mathbb{E}[T|\mathbf{x}]] \leq (M+1)\mathbb{E}_{\rho'}\left[\sum_{m=M+1}^{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})\right] + Nv\varepsilon \leq (M+1)NC_{\rho,\rho'}\sum_{m=M+1}^{N}\lambda_m + Nv\varepsilon. \tag{3.46}$$

Corollary 3.21 follows via non-negativity of the Kullback-Leibler divergence and Markov's inequality (lemma 3.19). □

Alternatively, if we suppose the model is correctly specified (assumption 3.b), we arrive at a stronger result.

Theorem 3.22. *With the same assumptions as theorem 3.20, but assumption 3.b in place of assumption 3.a*

$$\mathbb{E}[\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)] \leq \frac{N}{\sigma^2}\left((M+1)C_{\rho,\rho'}\sum_{m=M+1}^{\infty}\lambda_m + v\varepsilon\right). \tag{3.47}$$

Corollary 3.23. *Under the assumptions of theorem 3.22, with probability at least* $1-\delta$,

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{N}{\delta\sigma^2}\left((M+1)C_{\rho,\rho'}\sum_{m=M+1}^{\infty}\lambda_m + v\varepsilon\right). \tag{3.48}$$

Proof of theorem 3.22 and corollary 3.23. The proof is identical to the proof of theorem 3.20 and corollary 3.21 after substituting proposition 3.9 for proposition 3.6 to upper bound $\mathbb{E}[\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{x}, \mathbf{z}]$. □

We now consider the case when ridge leverage score sampling is used to select inducing inputs. In this case, we only provide bounds that hold with fixed probability.

Theorem 3.24. *Suppose assumptions 1 and 2 and assumption 3.a hold. If inducing points are initialized using Musco and Musco (2017, Algorithm 3) with fixed* $\delta \in (0, 1/32)$ *and* $S \in \mathbb{N}$, *then there exists a*

universal constant c such that with probability $1-5\delta$, we have $M < cS\log(S/\delta)$ and

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{N^2}{2S\delta\sigma^2}\left(1+\frac{C_y}{\delta\sigma^2}\right)\left(C_{\rho,\rho'}\sum_{m=S+1}^{\infty}\lambda_m\right). \tag{3.49}$$

Proof. Consider the events

$$\mathcal{E}_1^{\delta} := \left\{\|\mathbf{y}\|^2 \leq \frac{C_y N}{\delta}\right\} \tag{3.50}$$

$$\mathcal{E}_2^{\delta,S} := \{M \leq cS\log(S/\delta)\} \cup \left\{\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\text{op}} \leq \frac{1}{S}\sum_{m=S+1}^{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})\right\} \tag{3.51}$$

$$\mathcal{E}_3^{\delta,S} := \left\{\sum_{m=S+1}^{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) \leq \frac{N}{\delta}\sum_{m=S+1}^{\infty}\lambda_m\right\}. \tag{3.52}$$

By assumption 3.a and Markov's inequality $\Pr(\mathcal{E}_1^{\delta}) \geq 1-\delta$. By lemma 3.15 for any S, $\Pr(\mathcal{E}_2^{\delta,S}) \geq 1-3\delta$. Finally, by lemma 3.18 and Markov's inequality $\Pr(\mathcal{E}_3^{\delta,S}) \geq 1-\delta$. Hence, by the union bound for any S,

$$\Pr(\mathcal{E}_1^{\delta} \cap \mathcal{E}_2^{\delta,S} \cap \mathcal{E}_3^{\delta,S}) \geq 1-5\delta. \tag{3.53}$$

Now suppose $\mathcal{E}_1^{\delta} \cap \mathcal{E}_2^{\delta,S} \cap \mathcal{E}_3^{\delta,S}$, then

$$\mathfrak{D}_{\text{KL}}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{1}{2\sigma^2}\left(T + \frac{\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\text{op}} C_y N}{\delta\sigma^2}\right) \tag{3.54}$$

$$\leq \frac{N\|\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}\|_{\text{op}}}{2\sigma^2}\left(1+\frac{C_y}{\sigma^2\delta}\right) \tag{3.55}$$

$$\leq \frac{N\sum_{m=S+1}^{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{2S\sigma^2}\left(1+\frac{C_y}{\sigma^2\delta}\right) \tag{3.56}$$

$$\leq \frac{N^2\sum_{m=S+1}^{\infty}\lambda_m}{2S\sigma^2\delta}\left(1+\frac{C_y}{\sigma^2\delta}\right). \tag{3.57}$$

□

Theorem 3.25. *Take the same assumptions on $\mathbf{x}$ and $\mathbf{y}|\mathbf{x}$ as in theorem 3.22. Suppose inducing points are initialized using Musco and Musco (2017, Algorithm 3) with fixed $\delta \in (0, 1/32)$ and $S \in \mathbb{N}$. There exists a universal constant c such that with probability $1-5\delta$, we have $M < cS\log(S/\delta)$ and*

$$\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \leq \frac{N^2}{S\delta^2\sigma^2}\left(C_{\rho,\rho'}\sum_{m=S+1}^{\infty}\lambda_m\right). \tag{3.58}$$

Proof. The proof is nearly identical to the proof of theorem 3.25 using proposition 3.9 in place of proposition 3.6 and adjusting the definition of $\mathcal{E}_1^{\delta}$ accordingly. □

Remark 3.26. *Shawe-Taylor et al. (2005, Theorem 7) gives a high probability bound on $\mathcal{E}_3^{\delta,S}$ with a better dependence on δ by an application of McDiarmid's inequality. This could be used in place of one*

Table 3.1 Upper bound on average effective dimension (see eq. (3.61) for a definition) for common kernels with respect to a uniform measure on the unit square. The bound also holds if covariates are distributed according to a measure with compact support and bounded density.

Kernel	Effective dimension
Linear	$\leq D$
Matérn	$O(N^{\frac{D}{D+2\nu}})$
RBF	$O((\log N)^D)$

of the applications of Markov's inequality (lemma 3.19) to improve the dependence of theorem 3.24 and theorem 3.25 on δ.

Are these bounds useful?

Having established probabilistic upper bounds on the Kullback-Leibler divergence resulting from variational approximation, a simple question is: *do these bounds offer any insight into the efficacy of variational inference?* If in order for the upper bounds to be small, we need to take $M = N$, then they are not useful, as it is not hard to see that if $\mathbf{z} = \mathbf{x}$ then exact inference is recovered. In the next section, we discuss bounds on the eigenvalues of $T_{k,\rho}$ for common kernels (section 1.2.1) and input distributions. These bounds show that for many models and datasets satisfying assumptions 1 and 2 and assumption 3.a or assumption 3.b, the upper bounds in theorems 3.20, 3.22, 3.24 and 3.25 imply that the Kullback-Leibler divergence is small with $M \ll N$ inducing points.

3.2 Number of Inducing Points for Common Kernels

In this section, we consider the kernels discussed in section 1.2.1, and the number of inducing points required to ensure the upper bound on the Kullback-Leibler divergence from section 3.1.4 is small. Before delving into the specific bounds, we briefly discuss a reasonable conjecture supported by results in the kernel ridge regression literature, which we will see generally leads to the conclusion that fewer inducing points suffice than our bounds prove.

3.2.1 Effective Dimension

An intuitive answer to the number of inducing points needed for a model and dataset is *the number of inducing points needed to accurately approximate the posterior corresponds to the number of features that affect the posterior.* By the representer theorem (Kimeldorf and Wahba, 1970) any reasonable answer to this question should be less than N.

If $\sigma^2 = 0$, then N is a reasonable answer both to the number of features that affect the posterior and to the number of inducing points required. For $\sigma^2 = 0$, the posterior mean interpolates the points (x_n, y_n) and the posterior has zero variance at each x_n. Unless several $f(x_n)$ are linearly dependent, using fewer than N inducing points will result in a positive variance under the approximate posterior at one of

the observed data points, and so the Kullback-Leibler divergence to the posterior will be infinite. That the approximate posterior will have positive variance at some x_n can be seen from section 2.3, since the posterior variance is at least as large as the orthogonal projection onto the space orthogonal to the $f(z_n)$ (eq. 2.35). By the assumption that the $f(x_n)$ are linearly independent, for at least one $f(x_n)$ this projection is non-zero (by a dimensionality argument).

Additionally, a sensible notion of effective dimension must be smaller than

$$\dim \operatorname{span} \{f(x_n)\}_{n=1}^N = \dim(\mathcal{H}_{\mathbf{x}}), \tag{3.59}$$

which generalizes the earlier observation that it should be less than N. To summarize, letting d_{eff} denote such a notion we expect,

$$d_{\text{eff}} \leq \dim(\mathcal{H}_{\mathbf{x}}) \leq N \quad \text{and} \quad \sigma^2 \to 0 \Longrightarrow d_{\text{eff}} \to \dim(\mathcal{H}_{\mathbf{x}}) \quad \text{and} \quad \sigma^2 \to \infty \Longrightarrow d_{\text{eff}} \to 0. \tag{3.60}$$

The kernel ridge regression and online learning literature has shown that a particularly fruitful definition of the effective dimension is given by

$$d_{\text{eff}} = \operatorname{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}}(\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2 \mathbf{I})^{-1}\right) = \sum_{n=1}^{N} \frac{\lambda_n(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\lambda_n(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \sigma^2}. \tag{3.61}$$

The general conclusion of a line of research in kernel ridge regression starting with Alaoui and Mahoney (2015) is that using a Nyström approximation with d_{eff} inducing points (up to logarithmic factors) results in optimal regret for kernel ridge regression, see for example Rudi et al. (2015). Table 3.1 summarizes an upper bound on the average effective dimension of the kernels considered, under assumption 1 and with ρ having compact support.

Splitting the sum, for any $0 \leq M_0 \leq N$

$$\sum_{n=1}^{N} \frac{\lambda_n(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\lambda_n(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \sigma^2} \leq M_0 + \frac{1}{\sigma^2} \sum_{n=M_0+1}^{N} \lambda_n(\mathbf{K}_{\mathbf{x},\mathbf{x}}). \tag{3.62}$$

Equation (3.62) can be combined with lemma 3.18 and results on the eigenvalues for specific kernels discussed later in this section to show the bounds in table 3.1.

In order for the upper bounds on the Kullback-Leibler divergence from the previous section to be small, M must be such that $\frac{1}{\sigma^2}\sum_{m=M}^{N} \lambda_m$ is small. This requires at least as many inducing points as the effective dimension of the problem, which can be seen from eq. (3.62). The extent to which it requires more depends both on how small we need $\sum_{m=M}^{N} \lambda_m$ to be (e.g. $O(N^{-1}M^{-1})$), and how rapidly the eigenvalues of $T_{k,\rho}$ decay. At least some of the pessimism of our upper bounds relative to other results that suggest taking M to be close to the effective dimensions suffices is due to our choice to ensure the Kullback-Leibler divergence is small. Indeed, subsequent to our work, Nieman et al. (2021) showed under frequentist assumptions that if the Kullback-Leibler divergence is $o(N)$, the approximate posterior still contracts to a true, sufficiently smooth function used to generate the data.

While upper bounding the Kullback-Leibler divergence appears to be pessimistic, it has the advantage of implying guarantees if inducing points are first initialized with a scheme and then moved by maximizing the evidence lower bound (so long as line search is used to ensure monotonicity of the optimization). Additionally, it implies bounds on posterior moments and the probability of events under the true posterior (section 2.4) which cannot be concluded directly if the Kullback-Leibler divergence is $o(N)$. It is possible however that this type of error bound could be established indirectly for sufficiently large N via the arguments in Nieman et al. (2021).

3.2.2 Sums of Eigenvalues for Common Kernels

In order to instantiate the bounds in section 3.1.4 we recall several results on the sums of eigenvalues for the squared exponential and Matérn kernel.

Squared Exponential Kernel and Gaussian Covariates A minor modification of the argument in Seeger et al. (2008, Appendix 2), gives the following lemma for the squared exponential kernel and Gaussian covariates.

Lemma 3.27. *Let k be a squared exponential kernel (eq. 1.11) on $\mathbb{R}^D$ and ρ a standard (multivariate) Gaussian distribution. Then*

$$\sum_{m=M+1}^{\infty} \lambda_m = O(M \exp(-\alpha M^{1/D})), \tag{3.63}$$

where $\alpha = -\log B$, and B is as in eq. (1.22).

Proof. The proof of this proposition is nearly identical to an argument in Seeger et al. (2008). As the kernel is separable and the covariate distribution is isotropic each eigenvalue is a product of the eigenvalues of the kernel defined along each dimension. Taking eq. (1.22) into account each eigenvalue is of the form,

$$\lambda_m = (1/(2A))^{D/2} B^{m'}, \tag{3.64}$$

for some integer m' with a, A and B defined as in the eq. (1.22). Note that m and m' are only equal if $D = 1$. The number of times each eigenvalue with m' in the exponent is repeated is equal to the number of ways to write m' as a sum of D non-negative integers. By counting the multiplicity of each eigenvalue, Seeger et al. (2008, Appendix II) arrived at the bound

$$\lambda_{m+D-1} \leq (1/(2A))^{D/2} B^{m^{1/D}}. \tag{3.65}$$

Define $\tilde{M} = M - D + 1$, then for $M > D - 1$,

$$\sum_{m=M+1}^{\infty} \lambda_m \leq \left(\frac{1}{2A}\right)^{\frac{D}{2}} \sum_{m=\tilde{M}+1}^{\infty} B^{m^{1/D}} \tag{3.66}$$

$$\leq \left(\frac{1}{2A}\right)^{\frac{D}{2}} \int_{s=\tilde{M}}^{\infty} B^{s^{1/D}} ds \tag{3.67}$$

$$= \left(\frac{1}{2A}\right)^{\frac{D}{2}} D\alpha^{-D} \int_{t=\alpha\tilde{M}^{1/D}}^{\infty} \exp(-t) t^{D-1} dt \tag{3.68}$$

$$= \left(\frac{1}{2A}\right)^{\frac{D}{2}} D\alpha^{-D} \Gamma(D, \alpha(M-D+1)^{1/D}), \tag{3.69}$$

where in the second to last line we make the substitution $t = \alpha s^{1/D}$ and in the final line we recognized the integral as an incomplete Γ-function. From Gradshteyn and Ryzhik (2014, 8.352) for integer D and $r > 0$,

$$\Gamma(D, r) = (D-1)! e^{-r} \sum_{k=0}^{D-1} \frac{r^k}{k!} \leq D! e^{-r} \max_{0 \leq k \leq D-1} \frac{r^k}{k!}. \tag{3.70}$$

The supremum is achieved for $k = D - 1$ for fixed D and r sufficiently large, which in the case $r = \alpha(M + D - 1)^{1/D}$ (eq. 3.69) is the case for all $M \geq \frac{1}{\alpha} D^D + D - 1$. Under this assumption

$$\Gamma(D, r) \leq D! e^{-r} \frac{r^{D-1}}{(D-1)!} = D e^{-r} r^{D-1}. \tag{3.71}$$

Using this bound in eq. (3.66)

$$\sum_{m=M+1}^{\infty} \lambda_m \leq \left(\frac{1}{2A}\right)^{\frac{D}{2}} D^2 \alpha^{-D} \exp(-\alpha(M-D+1)^{1/D})(\alpha(M-D+1)^{1/D})^{D-1} \tag{3.72}$$

$$\leq \left(\frac{1}{2A}\right)^{\frac{D}{2}} \frac{D^2(M-D+1)}{\alpha} \exp(-\alpha(M-D)^{1/D}) = O(M \exp(-\alpha M^{1/D})). \tag{3.73}$$

□

We now state a corollary of theorem 3.20.

Corollary 3.28. *Let k be a squared exponential kernel on $\mathbb{R}^D$ and ρ' a Gaussian measure or a measure that is supported on a compact set with bounded density. Suppose assumptions 1 and 2 and assumption 3.b holds. For any $\gamma > 0$, and $\delta \in (0,1)$ there exists a constant C, which depends on the parameters of the kernel, the properties of ρ', and D such that for all N if inducing points are initialized with an ε-approximate M-determinantal point process (algorithm 2) with $\varepsilon = \frac{\gamma\sigma^2\delta}{N\sigma_k^2}$ and $M = C\log(\frac{N}{\delta\gamma\sigma^2})^D$ then with probability $1-\delta$, $\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) < \gamma$. Moreover, the computational cost of this algorithm is $O(N\log(\frac{N\sigma_k^2}{\delta\gamma\sigma^2})^{3D+1})$.*

A similar result follows from theorem 3.20 under assumption 3.a with a different choice of ε and a different leading constant C.

Matérn Kernels Widom (1963, Theorem 2) gave a relatively complete asymptotic characterization of the eigenvalues of $T_{k,\rho}$ for many stationary kernels and if ρ is compactly supported with a bounded density. For Widom's Theorem to apply, we need the following properties of the spectral density from Bochner's theorem (eq. 1.34), s, to hold:

1. For all $i \in \{1, \cdots, D\}$, fixing all $\omega^{(j)}, j \neq i$, there exists an $\omega_0^{(i)} \in \mathbb{R}$ such that $s(\omega)$ is monotonically increasing as a function of $\omega^{(i)}$ for all $\omega^{(i)} < \omega_0^{(i)}$ and is monotonically decreasing as a function of $\omega^{(i)}$ for $\omega^{(i)} \geq \omega_0^{(i)}$.
2. Let $\{\xi_i\}_{i=1}^{\infty}, \{\eta_i\}_{i=1}^{\infty}$, be sequences in $\mathbb{R}^D$ such that $\lim_f i \to \infty \frac{\|\eta_i - \xi_i\|}{\|\eta_i\|} = 0$ and $\lim_{i\to\infty} \|\xi_i\| = \infty$, then $\lim_{i\to\infty} \frac{|s(\xi_i)|}{|s(\eta_i)|} = 1$.
3. Let $\{\xi_i\}_{i=1}^{\infty}, \{\eta_i\}_{i=1}^{\infty}$, be sequences in $\mathbb{R}^D$ such that $\lim_{i\to\infty} \|\xi_i\|, \|\eta_i\| = \infty$ and $\lim_{i\to\infty} \frac{\|\xi_i\|}{\|\eta_i\|} = 0$, then $\lim_{i\to\infty} \frac{|s(\xi_i)|}{|s(\eta_i)|} = 0$.

If the spectral density of the kernel satisfies these conditions, and ρ is compactly supported with bounded density, r, the number of eigenvalues of $T_{k,\rho}$ greater than ε is asymptotically equivalent (as $\varepsilon \to 0$) to the volume of the region in $\mathbb{R}^D \times \mathbb{R}^D$ such that $r(x)s(\omega) > \varepsilon$. A precise statement of the result can be found in Widom (1963), and more discussion of the result is given in Seeger et al. (2008). Because of the second condition, Widom's theorem does not apply to kernels with rapidly decaying spectral densities, such as the squared exponential kernel (though more stationary kernels are analyzed in Widom, 1964 for uniformly distributed covariates). However, the Matérn kernel is known to satisfy the three conditions for any smoothness parameter ν.

Seeger et al. (2008) gave the following corollary, which we will use.

Lemma 3.29 (Seeger et al. (2008), Theorem 2)**.** *Let k be an isotropic kernel. Suppose k satisfies the criteria of Widom's theorem, the distribution, ρ, is supported inside a ball of radius R around the origin, and is bounded above by τ, then*

$$\lambda_m \leq \tau(2\pi)^D s\left(\frac{2\Gamma(D/2+1)^{2/D}}{R} m^{1/D}\right)(1+o(1)). \tag{3.74}$$

As the decay of the spectral density is the Fourier transform of the kernel, and because the decay of a Fourier transform is determined by the smoothness of the original function, we can interpret Widom's Theorem and lemma 3.29 as saying the eigenvalues of $T_{k,\rho}$ decay rapidly if the kernel is smooth.

Table 3.2 Number of inducing points and computational cost for Matérn kernel and squared exponential kernel on compact domains using an approximate M-determinantal point process to initialize inducing points. We assume $D < 2\nu$, otherwise the bounds are vacuous for Matérn kernels.

Kernel	$\mathbf{y}\|\mathbf{x}$	Theorem	M	Computation
Sq. Exp.	Assumption 3.b	Theorem 3.22	$O((\log N)^D)$	$O(N(\log N)^{3D+1})$
Sq. Exp.	Assumption 3.a	Theorem 3.20	$O((\log N)^D)$	$O(N(\log N)^{3D+1})$
Matérn ν	Assumption 3.b	Theorem 3.22	$O(N^{\frac{D}{2\nu-D}})$	$O(N^{\frac{2\nu+2D}{2\nu-D}}(\log N)^2)$
Matérn ν	Assumption 3.a	Theorem 3.20	$O(N^{\frac{2D}{2\nu-D}})$	$O(N^{\frac{2\nu+5D}{2\nu-D}}(\log N)^2)$

Table 3.3 Number of inducing points and computational cost for Matérn kernel and squared exponential kernel on compact domains using an approximate ridge leverage scores to initialize inducing points.

Kernel	$\mathbf{y}\|\mathbf{x}$	Theorem	M	Computation
Sq. Exp.	Assumption 3.b	Theorem 3.25	$O((\log N)^D \log\log N)$	$O(N(\log N)^{2D}(\log\log N)^2)$
Sq. Exp.	Assumption 3.a	Theorem 3.24	$O((\log N)^D \log\log N)$	$O(N(\log N)^{2D}(\log\log N)^2)$
Matérn ν	Assumption 3.b	Theorem 3.25	$O(N^{\frac{2D}{2\nu+D}}\log(N))$	$O(N^{\frac{2\nu+5D}{2\nu+D}}(\log N)^2)$
Matérn ν	Assumption 3.a	Theorem 3.24	$O(N^{\frac{2D}{2\nu+D}}\log(N))$	$O(N^{\frac{2\nu+5D}{2\nu+D}}(\log N)^2)$

Lemma 3.30. *Let k be a Matérn kernel with smoothness parameter ν on $\mathbb{R}^D$ and ρ a compactly supported measure with bounded density. Then for M sufficiently large,*

$$\sum_{m=M+1}^{\infty} \lambda_m = O(M^{-\frac{2\nu}{D}}). \tag{3.75}$$

Proof. From lemma 3.29 there exists an M_0 such that for all $m \geq M_0, \lambda_m \leq Cm^{-(2\nu+D)/D}$. For $M \geq M_0$

$$\sum_{m=M+1}^{\infty} \lambda_m \leq C \int_{m=M+1}^{\infty} x^{-\frac{2\nu+D}{D}} dx \leq C'(M+1)^{-\frac{2\nu}{D}}. \tag{3.76}$$

□

Corollary 3.31. *Let k be a Matérn kernel with smoothness parameter ν on $\mathbb{R}^D$ such that $2\nu \geq D$. Suppose assumptions 1 and 2 and assumption 3.b holds with ρ' a measure that is supported on a compact set with bounded density. For any $\gamma > 0$, and $\delta \in (0,1)$ there exists a constant C, which depends on the parameters of the kernel, the properties of ρ' and D, if inducing points are initialized with an ε-approximate M-determinantal point process (algorithm 2) with $\varepsilon = \frac{\gamma\sigma^2\delta}{2N\sigma_k^2}$ and $M = C(\frac{N}{\gamma\delta\sigma^2})^{D/(2\nu-D)}$ then with probability $1-\delta$, $\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) < \varepsilon$.*

The number of inducing points needed, and the computational cost of the algorithm are summarized in tables 3.2 and 3.3 for the approximate M-determinantal point process and ridge leverage score sampling methods respectively under different sets of assumptions.

3.3 Lower Bounds on the Kullback-Leibler Divergence

To this point we have focused on the questions *how large can the Kullback-Leibler divergence be* and *how many inducing points are sufficient to ensure it is small*? In this section we approach the inverse questions, *how small can the Kullback-Leibler divergence be* and *how many inducing points are necessary for it to be small*?

3.3.1 Several Lower Bounds on the Kullback-Leibler Divergence

Some subtlety emerges when considering lower bounds. From a frequentist perspective a natural quantity to consider is the MaxMin bound

$$\max_{\substack{\mathbf{y}\in\mathbb{R}^N \\ \|\mathbf{y}\|\leq\|\mathbf{y}'\|}} \min_{\mathbf{z}\in\mathcal{X}^M} \mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right), \tag{3.77}$$

which asks how large the Kullback-Leibler divergence can be if an adversary constructs $\mathbf{y}$ subject to norm constraints (alternatively one could assume $\mathbf{y}$ is a noisy realization of a true, sufficiently regular function), then the practitioner optimally selects $\mathbf{z}$ for this $\mathbf{y}$. A more Bayesian quantity to study would be

$$\mathbb{E}[\min_{\mathbf{z}\in\mathcal{X}^M} \mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{x}], \tag{3.78}$$

under assumption 3.b, which assumes that practitioner optimally selects $\mathbf{z}$ depending on $\mathbf{y}$. Both of these techniques involve understanding the optimal inducing point location for a specific $\mathbf{y}$, which we do not resolve here. The following lemma considers several forms of bounds that avoid this difficulty.

Lemma 3.32. *Given a kernel k, likelihood model with variance σ^2 and covariates $\mathbf{x}\in\mathcal{X}^N$ and any collection of inducing points $\mathbf{z}\in\mathcal{X}^M$*

$$\min_{\mathbf{z}\in\mathcal{X}^M} \mathbb{E}[\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)|\mathbf{x}, \mathbf{z}] \geq \tfrac{1}{2}\sum_{m=M+1}^{\infty} \frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2} \tag{3.79}$$

$$\min_{\mathbf{z}\in\mathcal{X}^M} \max_{\substack{\mathbf{y}\in\mathbb{R}^N \\ \|\mathbf{y}\|=\|\mathbf{y}'\|}} \mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \geq \frac{\|\mathbf{y}\|^2}{2\sigma^2}\frac{\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}})+\sigma^2} + \tfrac{1}{2}\sum_{m=M+1}^{N} \frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2} - \log\left(1+\frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2}\right) \tag{3.80}$$

$$\min_{\substack{\mathbf{y}\in\mathbb{R}^N \\ \|\mathbf{y}\|=\|\mathbf{y}'\|}} \min_{\mathbf{z}\in\mathcal{X}^M} \mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) \geq \tfrac{1}{2}\sum_{m=M+1}^{N} \frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2} - \log\left(1+\frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2}\right) \tag{3.81}$$

By the MinMax inequality, eq. (3.80) is an upper bound on eq. (3.77) and may not be a lower bound on optimal inducing point selection. Similarly, eq. (3.79) is an upper bound on eq. (3.78). They can be used to derive lower bounds for the performance of any inducing point method that selects inducing points independently of the response variables, including uniform sub-sampling of covariates, M-determinantal point process selection, ridge leverage score sampling and k-means++, but excluding

evidence lower bound maximization. On the other hand, the MinMin problem lower bounds both eq. (3.78) and eq. (3.79). We therefore analyze this quantity later in this chapter in order to obtain lower bounds on optimal inducing point selection, although it necessarily provides the loosest lower bounds.

Proof of lemma 3.32. Equation (3.79) follows from the lower bound provided in proposition 3.9 combined with the Eckart-Young-Mirsky theorem (theorem 3.10) and that $\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}}$ is positive definite so that the trace coincides with the Schatten 1-norm which is unitarily invariant.

Recall the form of the Kullback-Leibler divergence,

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) = \mathcal{L}(\theta) - \underline{\mathcal{L}}(\mathbf{z}, \theta) \tag{3.82}$$

$$= \frac{1}{2}\left(\mathbf{y}^\top(\mathbf{Q}^{-1} - \mathbf{K}^{-1})\mathbf{y} + \frac{\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{2\sigma^2} - \frac{1}{2}\log\frac{\det\mathbf{K}}{\det\mathbf{Q}}\right) \tag{3.83}$$

For Equation (3.81), since $\mathbf{Q}_{\mathbf{x},\mathbf{x}} \prec \mathbf{K}_{\mathbf{x},\mathbf{x}}$, the term

$$\frac{1}{2}\mathbf{y}^\top(\mathbf{Q}^{-1} - \mathbf{K}^{-1})\mathbf{y} \geq 0. \tag{3.84}$$

It remains to lower bound

$$\frac{\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{2\sigma^2} - \frac{1}{2}\log\frac{\det\mathbf{K}}{\det\mathbf{Q}} = \frac{1}{2}\sum_{m=1}^{N}\frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) - \lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\sigma^2} - \log\left(1 + \frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) - \lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2}\right) \tag{3.85}$$

$$\geq \frac{1}{2}\sum_{m=1}^{N}\frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) - \lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\sigma^2} - \log\left(1 + \frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) - \lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\sigma^2}\right), \tag{3.86}$$

where the last inequality uses that $\lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) \geq 0$ for all m. Since $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$ is rank M, for all $m \geq M+1$, $\lambda_m(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) = 0$. Also, the function $a - \log(1+a) \geq 0$ and equals 0 if $a = 0$. Hence,

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) \geq \frac{1}{2}\sum_{m=M+1}^{N}\frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2} - \log\left(1 + \frac{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})}{\sigma^2}\right). \tag{3.87}$$

To prove eq. (3.80) consider

$$\max_{\|\mathbf{y}'\| = \|\mathbf{y}\|} \mathbf{y}^\top(\mathbf{Q}^{-1} - \mathbf{K}^{-1})\mathbf{y} = \|\mathbf{Q}^{-1} - \mathbf{K}^{-1}\|_{\mathrm{op}}\|\mathbf{y}\|^2 \tag{3.88}$$

by the definition of operator norm. We now bound the operator norm below.

As the eigenvalues of the inverse of a positive definite matrix are the reciprocal of the eigenvalues of the original matrix, and since $\mathbf{Q}_{\mathbf{x},\mathbf{x}}$ is rank M, we know that the largest $N - M$ eigenvalues of $\mathbf{Q}^{-1}$ are equal to σ^2. Let V denote the span of the corresponding eigenvectors, so $\dim(V) \geq N - M$. On the other hand, the eigenvalues of $\mathbf{K}^{-1}$ take the form $\frac{1}{\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \sigma^2}$, the smallest $M+1$ of which are less than

or equal to $\frac{1}{\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}})+\sigma^2}$. Call this $M+1$ dimensional subspace W. Then,

$$\dim(V \cap W) = \dim(V) + \dim(W) - \dim(V+W) \tag{3.89}$$

$$\geq M+1+N-M-N = 1, \tag{3.90}$$

so there exists a $\mathbf{v} \in V \cap W$, $\|\mathbf{v}\| = 1$. Then

$$\|\mathbf{Q}^{-1} - \mathbf{K}^{-1}\|_{\text{op}} \geq \mathbf{v}^\top(\mathbf{Q}^{-1} - \mathbf{K}^{-1})\mathbf{v} \geq \frac{1}{\sigma^2} - \frac{1}{\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \sigma^2}, \tag{3.91}$$

and eq. (3.80) follows. □

3.3.2 Lower Bounds and Operator Eigenvalues

As mentioned in the previous subsection, we focus on determining how large M must be so that eq. (3.81) is large, as this implies lower bounds on eq. (3.77) and eq. (3.78), and is hence agnostic to the method for selecting inducing points. We note that Nieman et al. (2021) showed posterior consistency so long as the Kullback-Leibler divergence is $o(N)$, so our lower bounds should be interpreted narrowly in the context of necessary conditions for the Kullback-Leibler divergence to the posterior not to be large, and not necessarily as a diagnostic for other statistical properties of the variational posterior.

A sufficient condition for the bound eq. (3.81) to tend to infinity is that $\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}})$ tends to infinity with N. This follows from bounding the sum in eq. (3.81) below by the largest term and noting that $\lim_{a\to\infty} a - \log(1+a) = \infty$. The main technical obstacle to obtaining lower bounds on the number of inducing points needed to lower bound the Kullback-Leibler divergence asymptotically, and hence to obtain bounds on the number of inducing points needed, is to relate the matrix eigenvalue $\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}})$ back to $N\lambda_{M+1}$, the (scaled) eigenvalue of $T_{k,\rho}$.

Theorem 3.33 (Braun, 2006, Theorem 4)**.** *Suppose assumption 1 and assumption 2 hold with $\rho = \rho'$. Suppose additionally $k(x,x) \leq v$ for all $x \in \mathcal{X}$. Fix $\delta \in (0,1)$. Then for all m, with probability at least $1-\delta$,*

$$|\lambda_m - \tfrac{1}{N}\lambda_m(\mathbf{K}_{\mathbf{x},\mathbf{x}})| \leq \inf_{1\leq r\leq N} \lambda_m r\sqrt{\frac{r(r+1)v}{\lambda_r N\delta}} + \sum_{s=r}^{\infty}\lambda_s + \sqrt{\frac{2v\sum_{s=r+1}^{\infty}\lambda_s}{N\delta}} \tag{3.92}$$

$$= O\left(\lambda_m r^2 \lambda_r^{-1/2} N^{-1/2}\delta^{-1/2} + \sum_{s=r+1}^{\infty}\lambda_s + \sqrt{\frac{\sum_{s=r+1}^{\infty}\lambda_s}{N\delta}}\right). \tag{3.93}$$

From theorem 3.33 and eq. (3.81), we can derive a set of sufficient conditions for the Kullback-Leibler divergence to be large.

Theorem 3.34. *Suppose assumption 1 and assumption 2 hold with $\rho = \rho'$. Suppose additionally $k(x,x) \leq v$ for all $x \in \mathcal{X}$. Fix $\delta \in (0,1)$. Suppose $M = M(N)$ is such that $\lim \lambda_{M+1} N \to \infty$ and that there exists and $1 \leq r \leq N$ such that*

i. $\frac{(r+1)^4 v}{\lambda_r N \delta} \leq \frac{1}{16}$,
ii. $\sum_{s=r+1}^{\infty} \lambda_s \leq \frac{\lambda_{M+1}}{4}$, *and*
iii. $\sum_{s=r+1}^{\infty} \lambda_s \leq \frac{N\lambda_{M+1}^2 \delta}{8v}$.

Then with probability at least $1-\delta$, $\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) = \Omega(\lambda_{M+1}N)$.

Proof. By theorem 3.33 and by assumptions *i.*, *ii.*, *iii.*, we have with probability at least $1-\delta$

$$|\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}}) - N\lambda_{M+1}| \leq \tfrac{3}{4}N\lambda_{M+1}. \tag{3.94}$$

By the reverse triangle inequality, $\lambda_{M+1}(\mathbf{K}_{\mathbf{x},\mathbf{x}}) \geq \frac{N\lambda_{M+1}}{4}$. Using eq. (3.81) with probability at least $1-\delta$,

$$\mathfrak{D}_{\mathrm{KL}}\left(Q, P_{|\mathcal{D}}\right) \geq \frac{N}{4\sigma^2}\lambda_{M+1} - \log\left(1 + N\frac{\lambda_{M+1}}{4\sigma^2}\right) = \Omega(N\lambda_{M+1}). \tag{3.95}$$

The second inequality uses that $a - \log(1+a) = \Omega(a)$ as $a \to \infty$ and the assumption that $\lambda_{M+1}N \to \infty$. □

Remark 3.35. *Condition ii. in theorem 3.34 implies $r \geq M$. On the other hand, i. implies that $\lambda_r N$ must tend to infinity at least as fast as r^4, so r cannot be chosen to be too large relative to N.*

We now consider the specific examples of the squared exponential kernel with Gaussian covariates and the Matérn kernel with uniform covariates and determine a bound on the number of inducing points needed to ensure the Kullback-Leibler divergence to the posterior is $O(1)$ based on theorem 3.34.

Squared Exponential Kernel and Gaussian Covariates

Without loss of generality, we assume that the lengthscale matrix is the identity and that the covariates measure ρ is isotropic, as scaling these will only affect constants in the bounds presented.

Proposition 3.36. *Suppose k is an isotropic SE-kernel in D dimensions with lengthscale ℓ and variance v. Suppose the training covariates are independently identically distributed according to an isotropic Gaussian measure, ρ, on $\mathbb{R}^D$ with covariance matrix $\beta^2\mathbf{I}$. For any $r \in \mathbb{N}$, we have*

$$\lambda_r \geq \left(\frac{1}{2A}\right)^{D/2} B^{Dr^{1/D}}, \tag{3.96}$$

where λ_r denotes the r^{th} largest eigenvalue of the operator $T_{k,\rho}$ and A and B are defined in eq. (1.22).

Proof. Recall (eq. 3.64) the eigenvalues $T_{k,\rho}$ in this case are of the form,

$$\lambda_r = \left(\frac{1}{2A}\right)^{D/2} B^s \tag{3.97}$$

where the number of times each eigenvalue is repeated is equal to the number of ways to write s as a sum of D non-negative integers, where the order of the summands matters, which is $\binom{s+D-1}{D-1}$. The number of eigenvalues greater than $1/(2A)^{D/2}B^s$ is

$$\sum_{t=1}^{s}\binom{t+D-1}{D-1}=\binom{s+D}{D}. \tag{3.98}$$

The equality follows from observing that the right-hand side is equal to the number of way to write s as a sum of $D+1$ non-negative integers. For each of these representations, the first D integers sum to some $t \leq s$, and once these are fixed there is a unique choice for the final integer. This is equivalent to the left-hand side. We therefore conclude $\lambda_{\binom{s+D}{D}} = 1/(2A)^{D/2}B^s$. Define

$$\tilde{r}=\min_{s\in\{0\}\cup\mathbb{N}}\left\{\binom{s+D}{D}:\binom{s+D}{D}>r\right\}, \tag{3.99}$$

and let $\tilde{s}$ be the value of s achieving this minimum. Then,

$$\lambda_r=\lambda_{\tilde{r}}=\left(\frac{1}{2A}\right)^{D/2}B^{\tilde{s}} \quad \text{and} \quad \binom{\tilde{s}}{D}\leq\binom{\tilde{s}-1+D}{D}\leq r.$$

The first inequality on the right-hand side uses that the number of ways to choose a fixed number of elements from a set is increasing in the cardinality of the set and the second inequality uses the minimality of $\tilde{s}$. Using the lower bound, $\left(\frac{\tilde{s}}{D}\right)^D \leq \binom{\tilde{s}}{D}$, we obtain $\tilde{s}\leq Dr^{1/D}$, completing the proof of the lower bound. □

This leads us to the following corollary of theorem 3.34:

Proposition 3.37 (Lower bound on number of features for squared exponential kernel and isotropic Gaussian)**.** *Let k be a squared exponential kernel. Suppose N covariates are sampled independently and identically from an isotropic Gaussian density. Define $M(N)$ to be any function of N such that $\lim_{N\to\infty}M(N)/(\log N)^D = 0$; i.e. $M(N)=o((\log N)^D)$. Suppose inference is performed using any set of inducing inputs, Z such that $|Z|=M(N)$. Then for any $\mathbf{y}\in\mathbb{R}^N$, for any $\varepsilon\in(0,1/(4D))$ and for any $\delta\in(0,1)$, with probability at least $1-\delta$, $\mathfrak{D}_{KL}\left(Q,P_{|\mathcal{D}}\right)=\Omega(N^{1-\varepsilon})$.*

Remark 3.38. *We assume δ is independent of N in this statement, though it could be modified to allow δ to tend to 0 with N at a slow rate with minor modifications.*

Proof of proposition 3.37. The proof follows from choosing

$$r=\left\lceil\left(\frac{1}{4\alpha D}\log N\right)^D\right\rceil \quad \text{and} \quad M+1=\left\lfloor\left(\frac{\varepsilon}{\alpha D}\log N\right)^D\right\rfloor \tag{3.100}$$

with $\alpha = -\log B$. Below, by an abuse of notation we use C to denote any factor independent of N. By proposition 3.36

$$\lambda_r \geq CN^{-1/4} \quad \text{and} \quad \lambda_{M+1} \geq CN^{-\varepsilon}. \tag{3.101}$$

By lemma 3.27

$$\sum_{s=r+1}^{\infty} \lambda_s \leq C(\log N)^D N^{-1/(4D)}. \tag{3.102}$$

The result follows from checking the conditions in theorem 3.34 under the assumption that N is sufficiently large. □

Comparing corollary 3.28 and proposition 3.37, we conclude that in the case of the squared exponential kernel and Gaussian covariates:

A necessary and sufficient (assuming a good initialization is used) condition for $\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right)$ to be small is that the number of inducing points is $O((\log N)^D)$.

Matérn Kernel and Uniform Covariates

Widom's Theorem (Widom, 1963, Theorem 2.1) implies that the eigenvalues of the operator associated to the Matérn kernel with smoothness parameter ν and covariates uniformly distributed in the unit cube has eigenvalues satisfying $C_1 m^{\frac{-(2\nu+D)}{D}} \leq \lambda_m \leq C_2 m^{\frac{-(2\nu+D)}{D}}$ for some constant C_1 and C_2 independent of m, i.e. $\lambda_m = \Theta(m^{\frac{-(2\nu+D)}{D}})$.[6]

Proposition 3.39. *Suppose assumption 1 and assumption 2 hold with $\rho = \rho'$ and $T_{k,\rho}$ has eigenvalues satisfying $C_1 m^{-\eta} \leq \lambda_m \leq C_2 m^{-\eta}$ for all $m \geq 1$, some $\eta > 1$ and constants $C_1, C_2 > 0$. Suppose inference is performed using any set of inducing inputs Z such that $|Z| = M(N)$ with $M(N)$ any function such that $M = O(N^{\zeta})$ for some $\zeta \in (0, \frac{\eta-1}{\eta(4+\eta)})$. Fix any $\delta \in (0,1)$, then for N sufficiently large with probability at least $1-\delta$, $\mathfrak{D}_{KL}\left(Q, P_{|\mathcal{D}}\right) = \Omega(N^{1-\eta\zeta})$.*

Proof. Take $r = N^{\gamma}$ for some $\gamma \in (0, \frac{1}{4+\eta})$. Then,

$$\lambda_r \geq CN^{-\eta\gamma}, \tag{3.103}$$

so

$$(r+1)^4 v \lambda_r^{-1} N^{-1} \delta^{-1} \leq CN^{(4+\eta)\gamma - 1}. \tag{3.104}$$

[6] See Seeger et al. (2008) for more details on the derivation of this from Widom's Theorem.

This is less than $1/16$ for large enough N by our assumption on $\gamma \leq 1/(4+\eta)$ and so condition *i.* of theorem 3.34 is satisfied. Also,

$$\sum_{s=r}^{\infty} \lambda_s \leq CN^{\gamma(1-\eta)}. \tag{3.105}$$

The latter two conditions then hold if $M = N^{\zeta}$ for some $\zeta \in \gamma(\eta-1)/\eta$ and the result follows. □

In the case of D-dimensional Matérn kernels and a uniform covariate distribution, $\eta = \frac{2\nu+D}{D}$. By choosing ζ as large as possible, proposition 3.39 implies that for an arbitrary $\varepsilon > 0$, the Kullback-Leibler divergence is lower bounded by an increasing function of N if fewer than $\Omega\left(N^{\frac{2\nu D}{(2\nu+5D)(2\nu+D)}-\varepsilon}\right)$ inducing variables are used. This lower bound on the number of inducing variables becomes vacuous (i.e. the exponent tends to 0) as $\eta \to 1$ from above, meaning it is not useful when applied to rough kernels that we expect are extremely difficult to approximate. This is almost certainly an artifact of the analysis. There is a large gap between the upper bounds and lower bound, particularly when η is near 1 (i.e. for non-smooth kernels). The gap between the bounds is in part introduced by needing to choose M so that the error term from theorem 3.33 remains lower order and in part from using eq. (3.81) and only bounding a single eigenvalue below, instead of using eq. (3.80). To address this second issue, it would be interesting to instead consider either eq. (3.77) or eq. (3.78). An analysis of these likely requires a different approach than any of the lower bounds presented in lemma 3.32 to account for the interaction between $\mathbf{y}$ and the optimal choice of $\mathbf{z}$.

3.4 Related Work

We now describe several lines of research, both prior to and after the publication of Burt et al. (2019) and Burt et al. (2020b) that are closely related to the question of how many inducing points are needed to perform accurate sparse variational Gaussian process regression.

3.4.1 Prior and Concurrent Work

Guarantees on Nyström Approximations The bounds provided build on prior work, predominantly in the kernel ridge regression community, on the quality of Nyström approximation. In particular, one of the primary motivations for the development of fast approximate ridge leverage score algorithms, such as the one used in lemma 3.15 from Musco and Musco (2017), has been to provide generalization bounds for scalable approximate kernel ridge regression. As mentioned in section 3.1.2, because the mean of the Gaussian process posterior is the kernel ridge regression predictor, it is unsurprising that techniques developed for kernel ridge regression can be successfully applied to variational Gaussian process regression. In this line of work Bach (2013) provided generalization bounds for uniform random selection of columns, which were later improved by Alaoui and Mahoney (2015) using a ridge leverage approach. Gittens and Mahoney (2016) provides an extensive discussion of methods for selecting points to form Nyström approximations and the corresponding guarantees. Fast versions of approximate ridge

leverage sampling were developed in Calandriello et al. (2017) and Musco and Musco (2017) among other works, leading to practical algorithms with guarantees. Concurrent to Burt et al. (2019), and prior to our consideration of ridge leverage score sampling as a method for inducing point selection, Calandriello et al. (2019) provided guarantees for the marginal mean and variance of approximate Gaussian process regression using a ridge leverage score approach in an online learning problem. The conclusion from this analysis was that the number of inducing points should be roughly as large as the effective dimension (eq. 3.61).

From the perspective of approximation quality, our bounds differ from those in Calandriello et al. (2019) in essentially two ways. First, we consider the Kullback-Leibler divergence as a criterion for a successful approximation, which is stricter than the marginal moments (see corollary 2.11). The second main difference is our bounds suggest more inducing points are needed, which is perhaps unsurprising considering the Kullback-Leibler divergence is more stringent than marginal approximation of the first two moments (see also discussion around eq. 3.62). Of course, in many applications, such as the kernelized bandit setting considered in Calandriello et al. (2019), marginal approximation of the first two moments is sufficient, and ensuring the Kullback-Leibler divergence is small is not necessary, in which case the results of Calandriello et al. (2019) are likely more relevant.

The use of M-determinantal point processes for Nyström sampling has also been investigated in the context of kernel ridge regression (Li et al., 2016a), leading to guarantees on the approximation error. Dereziński et al. (2019) gave an algorithm for sampling an M-determinantal point process in time that is nearly-linear in N and polynomial in M. This was refined in Calandriello et al. (2020), who reduced the degree of the polynomial factor in M, and gave an algorithm that does not require considering all N columns while still providing an exact sample from an M-determinantal point process. The computational dependence on M in this algorithm is still high, $O(NM^{6.5} + M^{9.5})$. However, future advances in M-determinantal point process sampling could make exact sampling from an M-determinantal point process a feasible approach for inducing point placement in sparse variational Gaussian processes regression.

Inducing Point Selection in Gaussian Process Regression Within the Gaussian process community, the problem of how to select inducing inputs is well-studied. Zhu et al. (1997) and Ferrari-Trecate et al. (1999) considered finite dimensional approximations to Gaussian process regression using spectral properties of the kernel operator and matrix respectively. The approximation considered in Ferrari-Trecate et al. (1999) makes use of the same features as described in section 3.1.2, while Zhu et al. (1997) uses an analogous construction replacing the matrix eigenvectors and eigenvalues with the operator eigenvectors and eigenvalues coming from Mercer's Theorem. Zhu et al. (1997) and Ferrari-Trecate et al. (1999) consider the rate of decrease in mean square test error under the prior when adding additional features. While of theoretical interest, these methods are less generally applicable than inducing point methods as they require performing expensive matrix operations.

Many well-motivated heuristics for selecting inducing points have been proposed in the Gaussian process literature. Among the notable methods proposed include approximately minimizing $\operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})$ (Smola and Schölkopf, 2000), approximating the information gain of including a data point in the poste-

rior (Seeger et al., 2003), or using the k-means centers of the input distribution (Hensman et al., 2013, 2015b). Algorithm 1 results in essentially the same approximation as a pivoted Cholesky approximation, considered for kernel principal component analysis in Fine and Scheinberg (2001) and in Gaussian process regression by Foster et al. (2009). For kernels with a constant diagonal (e.g. stationary kernels) it coincides with the algorithm in figure 15 of Davies (2015). We do not provide an empirical comparison of the various methods for initializing inducing points; a comparison of a handful of methods on several datasets is carried out in Burt et al. (2020b).

3.4.2 Subsequent Work

There has been subsequent interest on guarantees for approximate Gaussian process regression with Nyström approximation in the machine learning community. Nieman et al. (2021) established contraction rates for sparse approximations under the (frequentist) assumption that the data consists of noisy observations of a true, sufficiently smooth function. Moreover, they show that the Kullback-Leibler divergence need not vanish for this contraction to occur. The number of inducing points needed for convergence is comparable to the number of inducing points needed in Calandriello et al. (2019), and is in general smaller than the number we show suffices for the Kullback-Leibler divergence to be small. Under slightly different assumptions (essentially an online learning setup as opposed to independent, random covariates), Vakili et al. (2022) established posterior contraction. Galy-Fajou and Opper (2020) proposed selecting inducing points by adding a candidate data point to the previously selected inducing points if it has a small covariance to all the previously selected inducing points. While the method they propose is admirably simple and likely fast and practical, it is unclear how to reach the conclusion that any sublinear number of inducing points leads to a reasonable bound on the error from the upper bound presented in Galy-Fajou and Opper (2020, Theorem 2).

3.5 Summary and Future Directions

In this chapter we sought answers to the question: *how many inducing points are necessary and sufficient for sparse variational Gaussian process approximation to be accurate*? We decided to use the Kullback-Leibler divergence between the approximate posterior and prior as our definition of an accurate approximation and derived upper and lower bounds on the number of inducing points for this to be small. We now discuss several open questions related to these results which we believe are of interest, before concluding with a few words of caution on the interpretation of the results presented.

3.5.1 Future Directions

We suggest two directions of research that would provide substantive and valuable extensions of the results presented.

Sparse Variational Gaussian Process Regression with Mini-batch Training For particularly large datasets, sparse Gaussian process regression is often performed without partially collapsing the evidence lower bound as in Titsias (2009) to compute the optimal posterior distribution over $Q_{f_\mathbf{z}}$, and the posterior mean and variance of this distribution are instead maximized with (natural) gradient descent using mini-batches of data. It would be of interest to determine sensible schemes for initializing inducing points in this setting that do not require a full pass over the training examples.

Additionally, we suspect interesting questions can be asked regarding the optimization over the parameters $\{\boldsymbol{\mu}_{f_\mathbf{z}}, \boldsymbol{\Sigma}_{f_\mathbf{z}}\}$. For example, one might reasonably hope that given an approximate solution to the optimal variational parameters for one set of hyperparameters, then few steps of stochastic gradient descent suffice to update these to new hyperparameters. It would be especially interesting if one could show that it is not necessary to do a full pass over the dataset to update these parameters, as that would imply a computational savings for mini-batch training of sparse variational Gaussian process regression.

The author is not sufficiently well-versed in optimization theory to speculate whether such a result is true, but believes that research in this direction would be useful to the Gaussian process community, potentially offering insight into the benefits of different parameterizations for $Q_{f_\mathbf{z}}$ and stochastic optimization methods as applied to sparse variational Gaussian process regression. Empirical work investigating parameterizations and optimization techniques for sparse variational Gaussian process regression is an active area of research, see for example Adam et al. (2021); Hensman et al. (2013); Panos et al. (2018); van der Wilk et al. (2020, 2022) and theory would be a useful guidepost for informing this research direction.

Non-conjugate Models Among the most interesting open questions related to the results presented in this chapter is whether any of the conclusions carry over to the case when a Gaussian process is used for the prior and a non-Gaussian likelihood is used. Opper and Archambeau (2009) proposed a Gaussian variational approximation for such models, while Hensman et al. (2015b) proposed a sparse version of this model. A natural question is: *how much of the error when performing sparse variational inference in non-conjugate Gaussian models is due to the sparsity, and how much is due to the Gaussian approximation*? Alternatively, if one targets a variational posterior with MCMC subject to factorization assumptions as in Hensman et al. (2015a), what is the Kullback-Leibler distribution of the target distribution to the posterior? A reasonable conjecture is that the error due to sparsity behaves similarly as in the case of sparse Gaussian process regression with a Gaussian likelihood, subject to some sort of mild conditions on the likelihood needed to prevent the true posterior from contracting too fast. This is a reasonable conjecture because the bounds presented in section 3.1.4 are closely related to the difficulty of approximating the prior with a finite-dimensional model, and so if a small change to the prior corresponds to a small change in the posterior, a similar result should hold in the non-conjugate case.

3.5.2 Interpretation and Two Words of Caution

The results in this chapter are designed to answer the question: *will it work?* for sparse variational Gaussian process regression. In some sense, they can be taken to have answered it in the affirmative, as we proved that under reasonable assumptions on the data-generating process and for certain kernels sparse approximations can be performed that are both fast and accurate. However, we add two words of caution, intended to address a trend of papers in the scalable Gaussian process research literature that the author believes use far too few inducing points to provide a realistic approximation to the posterior distribution.

As presented *these bounds are asymptotic with implicit constants that are heavily dependent on the model hyperparameters and underlying distribution of covariates*. It is very possible, and indeed likely that on the same dataset, more inducing points will be needed when using a squared exponential kernel with a short lengthscale than for a Matérn $5/2$ kernel with a long lengthscale, despite what the asymptotic bounds suggest. Additionally, if a very small likelihood noise is used, it will often be the case that sparse approximations will be inaccurate. In short, the bounds are not uniform in the model hyperparameters, and there are constants involved that cannot be determined without access to the ground truth distribution of covariates, which is rarely known. The dependence of the bounds on properties of the model and dataset should discourage Gaussian process researchers and practitioners from choosing a number of inducing points (e.g. $M = 100$ which is often reported in papers) and then running all experiments with this number of inducing points. Once a dataset has been observed, the a posteriori bounds on the Kullback-Leibler divergence and other diagnostics discussed in section 2.4, as well as in Davies (2015) and Huggins et al. (2020), are a better justification for the number of inducing points used on any given problem and can be used along with heuristics to guide more reasonable approximate inference.

Second and related, *because the bounds are not uniform in the model hyperparameters, it is difficult to determine how they interact with evidence lower bound maximization*. In general, under-fitting can occur when using the evidence lower bound in place of the log marginal likelihood for model selection. Additionally, this problem can be difficult to diagnose automatically, as the variational approximation is often very accurate at the set of hyperparameters found (see the discussion in section 2.5). While we do not have a satisfactory solution for this problem, common sense provides a useful starting point. If there is reason to believe there is signal in the data, and evidence lower bound maximization results in a large likelihood variance and an approximate posterior that collapses to the prior, then likely more inducing points are needed. While the upper bounds presented show that under reasonable assumptions sparse variational Gaussian process regression can work in a range of context with fewer inducing points than datapoints, both Gaussian process researchers and practitioners still have a responsibility to use common sense and available diagnostics to ensure approximate inference is successful on the particular problem considered.

Chapter 4

Gaussian Process Regression and Iterative Linear Algebra

In the previous two chapters, we focused on sparse variational approximations which rely on a low-rank approximation to the kernel matrix to provide an approximation to the posterior distribution and log marginal likelihood. While we showed in chapter 3 that in certain instances variational approximations that are simultaneously accurate and fast to compute can be found, we also gave an example where no method based on low-rank approximation can provide accurate approximations with a significant computational benefit (example 3.1).

Iterative linear algebra methods, such as the method of conjugate gradients (Hestenes and Stiefel, 1952), offer a complementary approach to computationally efficient approximate inference in Gaussian process regression that can be shown to converge in certain instances when low-rank approximations do not succeed: in example 3.1 the method of conjugate gradients returns an approximation to the predictive mean that is uniformly within ε of the true predictive mean in computational complexity $O\left(N^2 \log\left(\frac{\|\mathbf{y}\|_2}{\sigma\varepsilon}\right)\right)$, which is a large savings over practical direct methods (section 1.3) or accurate sparse variational approximations for this model and dataset.

Structure of the Chapter The focus of the chapter is the application of iterative methods from linear algebra to Gaussian process regression. We emphasize approaches to assessing whether a method has converged sufficiently to provide reliable results, as well as techniques for improving the ease of use of these approximations.

The first two sections of the chapter provide background information, addressing the questions:

- What are the method of conjugate gradients and Lanczos algorithm and what quantities do they allow us to compute or approximate (section 4.1)?
- How have iterative methods been used to approximate the posterior and perform model selection in Gaussian process regression (section 4.2)?

Our introduction will be far from exhaustive, and we refer the interested reader to Golub and Meurant (2009); Shewchuk (1994) and Golub and Van Loan (2013) for excellent introductions and discussion of the method of conjugate gradients and Lanczos algorithm. Davies (2015) and Pleiss et al. (2018) provide discussion on the application of iterative methods to Gaussian process regression.

The next three sections of this chapter explore methods for combining low-rank approaches and iterative approaches, focusing on model selection. The goal of this approach is to combine benefits of low-rank methods (deterministic objective function, stable optimization behavior of evidence lower bound) with the benefits of iterative methods (reduced bias on many problems). These sections answer the questions:

- How can we use iterative methods to derive new bounds on the log marginal likelihood (section 4.3)?
- How can we leverage bounds on the log marginal likelihood based on iterative methods to perform model selection (section 4.4)?
- Empirically, how does model selection based on a combination of low-rank methods and iterative methods compare to existing scalable methods for model selection such as other iterative approaches and evidence lower bound maximization (section 4.5)?

A theme of the approach taken in these sections is that assessing lower and upper bounds on quantities we are estimating with an iterative method gives a principled method for determining how long to run the iterative method. This is not a new idea, and builds on Gibbs and MacKay (1997) and Davies (2015). While practical, it appears to have fallen slightly out-of-fashion in modern applications of Gaussian process regression, and we emphasize that it can improve the ease of use and reliability of these methods. We further build on this approach in the sixth section of the chapter, which affirmatively answers the question:

- Can we design a method built on iterative approaches that automatically decides how much computation should be used to ensure that an estimate of the log marginal likelihood that has expected value within ε of the log marginal likelihood is returned (section 4.6)?

We conclude the chapter with several speculative directions for future research in the application of iterative methods to Gaussian process regression (section 4.7).

Themes of the Chapter Unlike the variational approach discussed in chapter 2 and chapter 3 which provides a canonical estimate for all quantities of interest through the lens of variational Bayesian inference, iterative approaches have a less unified perspective on model selection and approximate inference. As such, the approaches discussed may seem ad hoc. To structure the discussion, we return to the criterion laid out for any approximation in section 1.4, and highlight thematic aspects of these questions applied to iterative methods.

Will it work? Overall, we spend the least time on this question. A complete understanding of necessary and sufficient conditions for iterative methods remains elusive, although specific types of

assumptions under which these methods work can be identified using a priori bounds from numerical analysis.

As in the case of variational inference, the answer to this question must be adjusted to *how much computation does it require for an iterative method to work, and how does this depend on the model and dataset?* All the approaches discussed in this chapter rely on computing matrix-vector products, so in the absence of additional structure will require at least $\Omega(N^2)$ computation, but perhaps only $\Omega(N)$ memory depending on implementation details. On the other hand, under relatively weak assumptions a short calculation (eq. 4.9) shows that the predictive mean can be recovered to ε-accuracy with $O(\frac{N^{5/2}}{\sigma}\log(N/\sigma\varepsilon))$ computational cost which is a noticeable gain over the cubic cost typically associated with Gaussian process regression.

A simple heuristic for understanding when iterative methods will work is that they will provide accurate solutions quickly if the condition number of $\mathbf{K}$ is small. This will occur if 1. the likelihood variance is large or 2. all the datapoints are reasonably spread-out. The first criterion is also the case for variational methods. The second criterion suggests that iterative methods work for some problems on which the sparse variational approach will fail, including problems like example 3.1. While sufficient, the conditions fail to address all the situations where iterative methods work, especially when preconditioners are used. However, to first order assuming iterative methods will work if the likelihood variance is reasonably large and will run into various issues when the likelihood variance is very small is a good heuristic, and we will give several examples that show poor behavior of these methods when the likelihood variance is small (see for example figure 4.5).

Did it work? Due to the lack of a unified, probabilistic theory for iterative approximations in Gaussian process regression the problem of diagnosing behavior on a given dataset is crucial. Luckily, as these methods are borrowed from the numerical analysis community, approaches for diagnosing their behavior can also be borrowed. We highlight several a posteriori bounds on the accuracy of moment approximations returned by iterative methods (eqs. 4.4, 4.43, 4.47).

We will spend the majority of this chapter discussing the problem of approximate model selection. As in the case of evidence lower bound maximization, diagnosing approximate maximum marginal likelihood using iterative methods is challenging. Our heuristic approach to address this will be to obtain guarantees on the quality of an approximation to the log marginal likelihood, and hope that these translate into a reliable model selection procedure. This brings us to a theme of this chapter: *we should stop running an iterative method when running it for longer would not improve our estimates of quantities we are interested in for inference and model selection. In practice, this means designing a stopping criterion that directly relates to these quantities*. This is not a new idea as applied to Gaussian processes, and was discussed in Gibbs and MacKay (1997) and Davies (2015). However, we present simplified derivations of earlier results that allow us to refine and extend previous approaches.

Is it Easy to Use? The largest contribution of this chapter is made toward improving the ease-of-use of iterative approaches. We focus on improving the reliability of approximate maximum marginal likelihood with iterative approaches, and make several observations:

- Linking stopping criteria to the estimation of quantities of interest tends to improve the stability of gradient based model selection (for example figure 4.7).
- Using an optimizer with line search can simplify and accelerate model selection. It can be worth using a biased, but deterministic lower bound on the log marginal likelihood in place of a stochastic estimate with less bias (figure 4.4).
- In some instances, we are able to select good initial guesses for quantities we solve for with iterative methods. When performing model selection we often solve perturbed versions of the same problem many times in a row. By using the previous solution as an initial guess one can often avoid the need to run more than a handful of iterations (figure 4.2).

Bibliographic Notes The results presented in sections 4.3 to 4.5 originally appeared in Artemev et al. (2021) and the proposed method is included as part of the `GPflow` package (Matthews et al., 2017). The ideas in section 4.6 were developed in Burt et al. (2021). Both papers were written with Artem Artemev and Mark van der Wilk. The experimental results presented in this chapter, as well a final version of code used to run them, were produced by Artem Artemev. I wrote initial versions of the code for the methods and derived the inequalities used to motivate the methods.

4.1 Conjugate Gradients and Lanczos Quadrature

In section 1.3, we discussed an implementation of Gaussian process regression centered around computing a Cholesky decomposition of the (noisy) kernel matrix $\mathbf{K}$. Once this decomposition is computed, the log marginal likelihood and predictive posterior moments can be computed relatively efficiently. This is an example of a *direct method*: after a fixed amount of computation, an exact (up to numerical error) solution for a desired quantity is computed. An alternative approach is to use an *iterative method*, in which an initial guess to the solution is made, and then refined with subsequent calculation. The advantage of an iterative approach is that if the initial guess is good, or if the subsequent calculation converges rapidly, good approximate solutions can be found much faster than is possible with direct methods.

The central quantities we will be interested in computing are matrix-inverse vector products and log determinants, as these appear in the log marginal likelihood (eq. 1.38) and predictive posterior moments (eq. 1.6). We begin with background on the method of conjugate gradients, which is a memory efficient and (often) quickly converging iterative method for computing matrix-inverse vector products when the matrix is positive definite. We then turn to stochastic trace estimation, the Lanczos algorithm and Lanczos quadrature. These methods can be combined to estimate the log determinant of a positive definite matrix (Golub and Meurant, 2009, Section 11.6).

The information presented in this section is background material. Golub and Van Loan (2013) provides a thorough introduction to the methods discussed, except stochastic trace estimation. A detailed description of stochastic trace estimation can be found in Avron and Toledo (2011).

4.1.1 Krylov Subspaces

An important concept used in the iterative methods described in the remainder of this section is the *order-t Krylov subspace associated to a matrix and vector*,

$$\mathcal{K}_t(\mathbf{K},\mathbf{v}) = \text{span}\{\mathbf{K}^s\mathbf{v}\}_{s=0}^t \subset \mathbb{R}^N. \tag{4.1}$$

From the definition $\dim(\mathcal{K}_t(\mathbf{K},\mathbf{v})) \leq t+1$. Both the method of conjugate gradients and Lanczos algorithm can be formulated as solutions to search problems over the Krylov subspace, where the goal is to find the best approximation to a quantity of interest within $\mathcal{K}_t(\mathbf{K},\mathbf{v})$.

Krylov subspaces have several appealing properties in this context. First, the Krylov subspace generally reveals spectral information about that matrix $\mathbf{K}$: for a generic $\mathbf{v}$ and large t, $\mathbf{K}^t\mathbf{v} \approx c_\mathbf{v}\lambda_1^t(\mathbf{K})\mathbf{u}_1$, where $c_\mathbf{v} = \langle \mathbf{v},\mathbf{u}\rangle$ and $\mathbf{u}$ is the eigenvector associated to the largest eigenvector of $\mathbf{K}$. This can be seen from writing $\mathbf{v}$ in terms of the basis of eigenvectors of $\mathbf{K}$, and is the central motivation of the *power method* for computing eigenvalues of a symmetric matrix. Second, the basis for $\mathcal{K}_t(\mathbf{K},\mathbf{v})$ can be computed using just matrix-vector multiplications. These can be performed in a matrix-free fashion (e.g. Charlier et al., 2021) and are well-suited to exploit parallelization and any additional matrix-structure that accelerates matrix-vector multiplication. However, using Krylov subspace methods is not without obstacles. The vectors, $\{\mathbf{K}^s\mathbf{v}\}_{s=0}^t$ rapidly become nearly linearly dependent and for this reason do not form a good basis with which to perform computation. Both the method of conjugate gradients and the Lanczos algorithm instead construct an orthogonal (or conjugate) basis for $\mathcal{K}_t(\mathbf{K},\mathbf{v})$ that allows for more stable computation.

4.1.2 The Method of Conjugate Gradients

The method of Conjugate Gradients or simply *conjugate gradients* rephrases the problem of solving a linear system of equations as a quadratic optimization problem, and provides an iterative algorithm for solving this problem that is both memory efficient and rapidly convergent. Consider the $N \times N$ system of equations

$$\mathbf{K}\mathbf{v}^* = \mathbf{y}, \tag{4.2}$$

with $\mathbf{K}$ (strictly) positive definite. The solution to this linear system of equations can be written as a quadratic optimization problem,

$$\mathbf{v}^* = \min_{\mathbf{v}\in\mathbb{R}^N} \frac{1}{2}\mathbf{v}^\top\mathbf{K}\mathbf{v} - \mathbf{y}^\top\mathbf{v} =: \min_{\mathbf{v}\in\mathbb{R}^N} H(\mathbf{v}). \tag{4.3}$$

Equation (4.3) can be proved by setting the gradient of H to 0, and noting that the second derivative is $\mathbf{K}$, which is positive definite and so H is convex.

We have some initial guess at the solution $\mathbf{v}_0$, which is often taken to be $\mathbf{0}$. Define the residual, $\mathbf{r}_t = \mathbf{y} - \mathbf{K}\mathbf{v}_t$. At iteration t, the method of conjugate gradients computes an approximation to $\mathbf{v}^*$ by

maximizing $H(\mathbf{v})$ over the affine space $\mathbf{v}_0 + \mathcal{K}(\mathbf{K}, \mathbf{r}_0)$. That is,

$$\mathbf{v}_t = \arg \min_{\mathbf{v} \in \mathbf{v}_0 + \mathcal{K}(\mathbf{K}, \mathbf{r}_0)} H(\mathbf{v}). \tag{4.4}$$

A bit of rearranging shows,

$$\mathbf{v}_t = \arg \min_{\mathbf{v} \in \mathbf{v}_0 + \mathcal{K}(\mathbf{K}, \mathbf{r}_0)} \|\mathbf{v}^* - \mathbf{v}_t\|_{\mathbf{K}}. \tag{4.5}$$

That is, the method of conjugate gradients finds the minimum error approximation to the solution to the linear system in eq. (4.2) within the affine subspace searched over, as measured by the norm induced by the positive definite matrix $\mathbf{K}$.

The minimization in eq. (4.4) is achieved by computing the gradient of $H(\mathbf{v})$ at each iteration, which is $\mathbf{r}_t$, then ensuring that at each iteration the step taken is *conjugate*, meaning orthogonal with respect to the inner product

$$\langle \mathbf{v}, \mathbf{w} \rangle_{\mathbf{K}} = \mathbf{v}^\top \mathbf{K} \mathbf{w}, \tag{4.6}$$

to all previous steps. Ensuring conjugacy of the step avoids back-tracking, which can slow the convergence of typical gradient descent on quadratic optimization problems.

At step t let $\mathbf{\Pi}_t$ denote projection onto $\mathcal{K}(\mathbf{K}, \mathbf{r}_0)$ with respect to the $\mathbf{K}$-inner product. Then $\mathbf{p}_t = (\mathbf{I} - \mathbf{\Pi}_t)\mathbf{r}_t$ is the projection of the residual onto the orthogonal complement of the space searched so far. The method of conjugate gradients takes a step in this direction at each iteration. The optimal step size in this direction can be computed in closed-form as $\langle \mathbf{r}_t, \mathbf{p}_t \rangle / \|\mathbf{p}_t\|_{\mathbf{K}}$. While it appears that this algorithm requires storage of all the $\{\mathbf{p}_i\}_{i=1}^t$ in order to form the projection matrix $\mathbf{\Pi}_t$, this is not the case due to a recurrence relation that allows $(\mathbf{I} - \mathbf{\Pi}_t)\mathbf{r}_t$ to be computed using only $\mathbf{p}_{t-1}$ and several scalars. The algorithm is detailed in algorithm 3, where we set $\mathbf{Q} = \mathbf{I}$. The $\mathbf{p}_i$ form a conjugate basis for $\mathcal{K}(\mathbf{K}, \mathbf{r}_0)$, and the main iteration in conjugate gradients can be seen as generating a new direction via the matrix-vector multiplication to increase the order of the Krylov subspace searched over, then performing Gram-Schmidt with respect to the $\mathbf{K}$-inner product to ensure conjugacy of the direction in which the update will be performed.

Condition Number and Numerical Considerations

The practical performance of the method of conjugate gradients depends on the quality of the initial solution $\mathbf{v}_0$ and properties of the matrix $\mathbf{K}$. The number of iterations needed to achieve a small error depends on the entire distribution of eigenvalues of $\mathbf{K}$ (Hackbusch, 1994, Remark 10.15). However, the most commonly quoted upper bound on the norm of the error of conjugate gradient after t iterations depends only on the *condition number* of $\mathbf{K}$, $\kappa(\mathbf{K}) = \lambda_1(\mathbf{K}) / \lambda_N(\mathbf{K})$.

Proposition 4.1 (Golub and Van Loan, 2013, Theorem 10.2.6)**.** *Let* $\mathbf{v}^*$ *denote the solution to the system of linear equations. For any* $t \in \mathbb{N}$*,*

$$\|\mathbf{v}^* - \mathbf{v}_t\|_{\mathbf{K}} \leq 2\left(1 - \frac{2}{\sqrt{\kappa(\mathbf{K})}+1}\right)^t \|\mathbf{v}^* - \mathbf{v}_0\|_{\mathbf{K}}. \tag{4.7}$$

In the case of a kernel matrix, with a kernel satisfying $k(x,x) \leq v$ we have $\lambda_N(\mathbf{K}) \geq \sigma^2$ and

$$\lambda_1(\mathbf{K}) = \sigma^2 + \lambda_1(\mathbf{K}_{\mathbf{x},\mathbf{x}}) \leq \operatorname{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}}) + \sigma^2 \leq Nv + \sigma^2, \tag{4.8}$$

hence $\kappa(\mathbf{K}) \leq 1 + Nv/\sigma^2$. This can be quite large if the signal-to-noise ratio is high or if N is large, and it can be the case that the error remains reasonably large for many iterations. In particular, inverting proposition 4.1 shows a sufficient condition for $\|\mathbf{v}^* - \mathbf{v}_t\|_{\mathbf{K}} \leq \varepsilon$ is

$$t \geq \log\frac{2\|\mathbf{v}^* - \mathbf{v}_0\|_{\mathbf{K}}}{\varepsilon} \frac{1}{\log(1 + \frac{2}{\sqrt{\kappa(\mathbf{K})}+1})} = \log\frac{2\|\mathbf{v}^* - \mathbf{v}_0\|_{\mathbf{K}}}{\varepsilon}\left(\frac{\sqrt{\kappa(\mathbf{K})}}{2} + O(1)\right). \tag{4.9}$$

The asymptotic equality comes from computing the Laurent series of $1/\log(1+a)$ at $a = 0$. In the case that the condition number is $O(N)$ and $\mathbf{v}_0 = \mathbf{0}$, we see that the number of iterations required is not more than $O(\sqrt{N}\log\frac{\|\mathbf{v}^* - \mathbf{v}_0\|_{\mathbf{K}}}{\varepsilon})$. Further, this sort of control on the error suffices to control the predictive mean function, in Gaussian process regression, as shown in lemma 4.4. On the other hand, there are instances in which even for large N, the condition number is small: in example 3.1, the condition number is $O(1)$, so that the number of iterations that need to be run for the method of conjugate gradients to converge to a desired accuracy is small.

In cases where the condition number is large and conjugate gradient is slow to converge, numerical issues can arise, further slowing convergence. In particular, Gram-Schmidt is known to be unstable as errors compound and, after more than a handful of iterations, the collection $\{\mathbf{p}_i\}_{i=1}^t$ are no longer conjugate due to numerical error (Golub and Van Loan, 2013, Sections 5.2.7-5.2.9). This slows, but does not prevent, convergence. A detailed description of the method of conjugate gradients in finite precision is beyond the scope of this thesis, and we refer the interested reader to Meurant (2006).

Preconditioning

A practical way to alleviate issues with the condition number of $\mathbf{K}$ is to introduce a positive definite matrix referred to as a *preconditioner*, which we suggestively denote by $\mathbf{Q} \in S_{++}^N$ such that $\mathbf{K} \approx \mathbf{Q}$ and $\mathbf{Q}^{-1}\mathbf{v}$ can be computed efficiently. The preconditioned conjugate gradient algorithm is given in algorithm 3, and the original conjugate gradient algorithm is recovered by taking $\mathbf{Q} = \mathbf{I}$. Preconditioned conjugate gradients is equivalent to solving the system of equations,

$$\mathbf{Q}^{-1/2}\mathbf{K}\mathbf{Q}^{-1/2}\hat{\mathbf{v}}^* = \mathbf{Q}^{-1/2}\mathbf{y}, \tag{4.10}$$

with $\hat{\mathbf{v}}^* = \mathbf{Q}^{1/2}\mathbf{v}^*$ via conjugate gradients. It is unnecessary to compute a matrix square root of $\mathbf{Q}$ to use preconditioned conjugate gradients, as evidenced in algorithm 3.

The hope when using preconditioning is that $\kappa(\mathbf{Q}^{-1/2}\mathbf{K}\mathbf{Q}^{-1/2}) \ll \kappa(\mathbf{K})$, in which case proposition 4.1 suggests faster convergence may occur. In the extreme case $\mathbf{Q} = \mathbf{K}$, the condition number takes its minimum value of 1 and the method of conjugate gradients converges to the exact solution in a single iteration.

A variety of preconditioners have been investigated in the context of kernel methods, including block diagonal (Jacobi) preconditioners and Nyström preconditioners (Cutajar et al., 2016). More sophisticated preconditioners based on, for example, hierarchical approximations to the matrix can also be used (Geoga et al., 2020). Care is needed when designing a preconditioner, as seemingly sensible preconditioners can make the condition number of the kernel matrix significantly worse on real datasets and hinder convergence, see Cutajar et al. (2016, Figure 2) for such an example. Following earlier work in the numerical analysis literature (Golub and Van Loan, 2013, Section 10.3.2), Gardner et al. (2018) advocate using an incomplete Cholesky decomposition as a preconditioner. This is equivalent to using a Nyström approximation-based preconditioner with inducing points selected via algorithm 1.

Algorithm 3 The Method of Conjugate Gradients

Input: $\mathbf{K} \in S^N_{++}$, $\mathbf{y} \in \mathbb{R}^N$, $\mathbf{v}_0 \in \mathbb{R}^N$, preconditioner $\mathbf{Q} \in S^N_{++}$ stopping criterion STOP : $\mathbb{R}^N \to \{\text{True}, \text{False}\}$.
Returns: An (approximate) solution to the system of equation $\mathbf{K}\mathbf{v}^* = \mathbf{y}$.
$t = 0$, $\mathbf{r}_0 = \mathbf{y} - \mathbf{K}\mathbf{v}_0$, $\mathbf{z}_0 = \mathbf{Q}^{-1}\mathbf{r}_0$
while Not STOP($\mathbf{r}$) **do**
  $t = t + 1$
  if t=1 **then**
    $\mathbf{p}_1 = \mathbf{z}_0$
    $\gamma_1 = \langle \mathbf{r}_0, \mathbf{z}_0 \rangle$
  else
    $\gamma_t = \langle \mathbf{r}_{t-1}, \mathbf{z}_{t-1} \rangle$
    $\tau = \gamma_t / \gamma_{t-1}$
    $\mathbf{p}_t = \mathbf{z}_{t-1} + \tau \mathbf{p}_{t-1}$
  end if
  $\nu = \gamma_t / \|\mathbf{p}_t\|^2_{\mathbf{K}}$
  $\mathbf{v}_t = \mathbf{v}_{t-1} - \nu \mathbf{p}_t$.
  $\mathbf{r}_t = \mathbf{r}_{t-1} - \nu \mathbf{K}\mathbf{p}_t$
  $\mathbf{z}_t = \mathbf{Q}^{-1}\mathbf{r}_t$
end while

4.1.3 Stochastic Trace Estimation

In order to evaluate the log determinant that appears in the marginal likelihood, or its gradient with respect to kernel hyperparameters, one generally needs to evaluate the trace of a matrix. In the case of

the gradient, this arises naturally,

$$\frac{\partial \log\det \mathbf{K}}{\partial \theta} = \operatorname{tr}\left(\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{K}^{-1}\right), \tag{4.11}$$

where $\frac{\partial \mathbf{K}}{\partial \theta}$ is the matrix formed by taking derivatives of each entry of $\mathbf{K}$ with respect to θ. In the case of the log determinant itself, we arrive at an expression involving a trace by writing

$$\log\det \mathbf{K} = \operatorname{tr}\log \mathbf{K}, \tag{4.12}$$

where the matrix logarithm is defined as the matrix with the same eigenvectors as $\mathbf{K}$, but replacing each eigenvalue of $\mathbf{K}$ with its logarithm. While one can approximately solve the N $N \times N$ system of equations presented in eq. (4.11) via the method of conjugate gradients, the result would involve repeated matrix-matrix multiplications and be very expensive. Hutchinson (1989) observed that if $\mathbf{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{W} \in \mathbb{R}^{N\times L}$ for some $L \leq N$ satisfies $\mathbb{E}[\mathbf{W}\mathbf{W}^\top] = \mathbf{I}$, then

$$\operatorname{tr}\overbrace{\mathbf{A}}^{N\times N} = \operatorname{tr}(\mathbf{A}\mathbf{I}) = \operatorname{tr}(\mathbf{A}\mathbb{E}[\mathbf{W}\mathbf{W}^\top]) = \mathbb{E}[\operatorname{tr}(\overbrace{\mathbf{W}^\top\mathbf{A}\mathbf{W}}^{L\times L})] = \mathbb{E}\Big[\sum_{\ell=1}^{L}\mathbf{w}_\ell^\top\mathbf{A}\mathbf{w}_\ell\Big], \tag{4.13}$$

where we have used proposition A.10 and linearity of trace, and $\mathbf{w}_\ell$ denotes the ℓ^{th} column of $\mathbf{W}$. We can therefore construct an unbiased estimator $\operatorname{tr}\left(\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{K}^{-1}\right) \approx \operatorname{tr}\left(\mathbf{W}^\top\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{K}^{-1}\mathbf{W}\right)$ by solving only L $N \times N$ system of equations. Frequently, $\mathbf{W}$ is selected to be have independently and identically distributed columns each of which contains appropriately scaled Rademacher or Gaussian random variables. For $L > 1$, conceivably introducing negative correlations in the columns of $\mathbf{W}$ could yield modest amounts of variance reduction. Among $\mathbf{W}$ with independent, zero mean columns, Rademacher vectors minimize the variance of the estimator Hutchinson (1989, Proposition 1). On the other hand, concentration results have been established for this estimator with different random variables, and are somewhat tighter for Gaussian random variables than for Rademacher random variables (Avron and Toledo, 2011).

4.1.4 The Lanczos Method

We now describe the Lanczos method, which is closely related to the method of conjugate gradients and is the basis of a log determinant estimator proposed in Bai et al. (1996).

The goal of the Lanczos method is to construct a tridiagonalization of a positive definite matrix,

$$\mathbf{K} = \mathbf{U}\mathbf{T}\mathbf{U}^\top, \tag{4.14}$$

with the columns of $\mathbf{U}$ orthonormal and $\mathbf{T}$ tridiagonal. This is done by selecting a starting vector $\mathbf{u}_1$, $\|\mathbf{u}_1\|_2 = 1$, which is taken to be the first column of $\mathbf{U}$. In the t-th iteration, an orthogonal basis for the order-$(t-1)$ Krylov subspace associated to $\mathbf{K}$ and $\mathbf{u}_1$ is formed by extending the basis for the order-t

space by computing a matrix-vector product and performing Gram-Schmidt.[1] As in the case of the method of conjugate gradients a two-term recurrence can be applied, so that in order to compute $\mathbf{T}$ it is not necessary to store all the columns of $\mathbf{U}$. Generally the algorithm is stopped after $t \ll N$ iterations and returns an approximate low-rank factorization

$$\mathbf{K} \approx \mathbf{U}^{(t)} \mathbf{T}^{(t)} \mathbf{U}^{(t)\top}, \tag{4.15}$$

where $\mathbf{U}^{(t)} \in \mathbb{R}^{N \times t}$ consists of t orthogonal columns spanning the order-$(t-1)$ Krylov subspace associated to $\mathbf{u}_1$ and $\mathbf{K}$ and $\mathbf{T}^{(t)} \in \mathbb{R}^{t \times t}$ is tridiagonal. The extreme eigenvalues of $\mathbf{T}^{(t)}$ converge rapidly to the corresponding eigenvalues of $\mathbf{K}$. In particular, the eigenvalues of $\mathbf{T}^{(t)}$ are *Ritz values* with respect to $\mathbf{K}$ and the t-dimensional Krylov subspace $\mathcal{K}_t(\mathbf{K}, \mathbf{u}_1)$ meaning that they have the characterization

$$\lambda_m(\mathbf{T}^{(t)}) = \min_{\substack{U \subset \mathcal{K}_{t-1}(\mathbf{K}, \mathbf{u}_1) \\ \dim(U) = \dim(\mathcal{K}_t(\mathbf{K}, \mathbf{u}_1)) - m + 1}} \max_{\substack{\mathbf{v} \in U \\ \mathbf{v} \neq 0}} \frac{\mathbf{v}^\top \mathbf{K} \mathbf{v}}{\|\mathbf{v}\|^2}. \tag{4.16}$$

Recall that the eigenvalues of a symmetric matrix $\mathbf{K}$ have the characterization (Horn and Johnson, 2012, Theorem 4.2.6),

$$\lambda_m(\mathbf{K}) = \min_{\substack{U \subset \mathbb{R}^N \\ \dim(U) = N - m + 1}} \max_{\substack{\mathbf{v} \in U \\ \mathbf{v} \neq 0}} \frac{\mathbf{v}^\top \mathbf{K} \mathbf{v}}{\|\mathbf{v}\|^2}. \tag{4.17}$$

Comparing eq. (4.16) and eq. (4.17), we see that the approximations to the eigenvalues of $\mathbf{K}$ computed using the Lanczos method are the analogue of the eigenvalues of $\mathbf{K}$, upon restricting to the Krylov subspace. Because the Krylov subspace is formed by applying powers of the matrix, extreme eigenvalues of $\mathbf{K}$ are quickly identified by this approach, (Golub and Van Loan, 2013, Chapter 9).

A basic version of the Lanczos algorithm is described in algorithm 4, although we note that this version should not be implemented for numerical reasons. For more stable versions, see Golub and Van Loan (2013, Chapter 9.2). In practice, the columns of $\mathbf{U}$ often fail to be orthogonal due to numerical errors and re-orthogonalization may be needed (i.e. orthogonalizing against a subset of the columns in $\mathbf{U}$ instead of relying on a recurrence relation to ensure that $\mathbf{u}_{i+1}$ is orthogonal to $\text{span}\{\mathbf{u}_s\}_{s=0}^{i}$). This means that some or all of the columns in $\mathbf{U}$ need to be stored, dramatically increasing the memory overhead of the algorithm. As in the case of conjugate gradients, preconditioned variants of the Lanczos algorithm can also be derived and can help with the convergence properties of the algorithm.

[1] If the order-t Krylov space is not $t+1$ dimensional, then the Lanczos algorithm will terminate early. This can happen if $\mathbf{K}$ has a repeated eigenvalue, or if the initial vector is orthogonal to at least one eigenvector of $\mathbf{K}$.

Algorithm 4 The Lanczos algorithm. The $\mathbf{u}_i$ computed are the columns of $\mathbf{U}^{(t)}$, the α_i are the diagonal entries of $\mathbf{T}^{(t)}$, while the β_i are the off-diagonal entries of $\mathbf{T}^{(t)}$, which is symmetric tridiagonal. The description below does not lead to a numerically stable algorithm, and other references e.g. Golub and Van Loan (2013, Chapter 9) should be consulted for practical implementations.

Input: $\mathbf{K} \in S^N_{++}$, $\mathbf{u}_1 \in \mathbb{R}^N, \|\mathbf{u}_1\| = 1$, $\mathbf{v}_0 \in \mathbb{R}^N$, preconditioner $\mathbf{Q} \in S^N_{++}$, $t \in \mathbb{N}$.
Returns: An (approximate) tridiagonalization of $\mathbf{K}$, $\mathbf{K} \approx (\mathbf{U}^{(t)})^\top \mathbf{T}^{(t)} \mathbf{U}^{(t)}$.
$\mathbf{r}_0 = \mathbf{u}_1$, $\beta_0 = 1$, $\mathbf{u}_0 = \mathbf{0}$.
for $1 \leq i \leq t$ **do**
  $\alpha_i = \mathbf{u}_i^\top \mathbf{K} \mathbf{u}_i$
  $\mathbf{r}_i = (\mathbf{K} - \alpha_i \mathbf{I})\mathbf{u}_i - \beta_{i-1}\mathbf{u}_{i-1}$.
  $\beta_i = \|\mathbf{r}_i\|^2$.
  $\mathbf{u}_{i+1} = \mathbf{r}_i / \beta_i$.
end for

4.1.5 Lanczos Quadrature

When computing the log determinant, we are faced with the task of computing $\log\det\mathbf{K}$. As noted previously, this can be rewritten,

$$\log\det\mathbf{K} = \operatorname{tr}\log\mathbf{K} \approx \sum_{\ell=1}^{L} \mathbf{w}_\ell^\top \log\mathbf{K}\mathbf{w}_\ell, \tag{4.18}$$

where the $\mathbf{w}_\ell$ are columns of a random matrix $\mathbf{W}$ satisfying $\mathbb{E}[\mathbf{W}\mathbf{W}^\top] = \mathbf{I}$ and the approximation coming from stochastic trace estimation is unbiased.

Gauss and Gauss-Radau Quadrature

We now take a brief detour to describe Gauss quadrature, which will be the key to estimating eq. (4.18). Faced with an integral on the real line of the form,

$$\int_a^b f(\lambda) d\mu(\lambda) \tag{4.19}$$

where $a, b \in \mathbb{R} \cup \{-\infty, \infty\}$ the central idea of (interpolatory) quadrature is to approximate f by a degree H polynomial for some $H \in \mathbb{N}$, and compute the integral of this polynomial exactly. We assume we can compute the moments of the measure μ, in which case integrating a polynomial is easy.

A straightforward approach to constructing a polynomial surrogate for f is to select $H+1$ points in $[a,b]$ referred to as *nodes*, $\{s_h\}_{h=0}^H$, and define the polynomial π_H to be the (unique) polynomial of degree at most H interpolating $\{f(s_h)\}_{h=0}^H$. This leads to an approximation of the form,

$$\int_a^b f(\lambda) d\mu(\lambda) \approx \sum_{h=0}^{H} w_h f(s_h) =: G_H(f) \tag{4.20}$$

for some weights $w_h \in \mathbb{R}$. The weights are determined by the requirement that π_H is interpolatory, and depend on the location of the nodes and the moments of the measure μ. We refer to such a procedure as a 'quadrature rule'. By the uniqueness of the interpolating polynomial, π_H, this rule computes the integral of any polynomial of degree at most H exactly.

Perhaps surprisingly, it is possible to integrate higher degree polynomials exactly using only $H+1$ nodes. A *Gauss quadrature rule* with $H+1$ nodes is a quadrature rule that correctly integrates f if f is any polynomial of degree up to $2H+1$. The idea is to select the location of the nodes to be the zeroes of a degree $H+1$ polynomial that is orthogonal to all polynomials of degree H, in which case a standard argument (see for example Stoer and Bulirsch, 2002, Theorem 3.6.12), shows that the quadrature rule integrates all polynomials of degree up to $2H+1$ exactly.

At least if f is reasonably smooth, Gauss quadrature provides very accurate estimates of the integral eq. (4.19) with only a modest number of nodes. Additionally, if we have information about the derivative of f, we can understand the direction of the error in the estimate eq. (4.20).

Theorem 4.2 (Stoer and Bulirsch, 2002, Theorem 3.6.24)**.** *Suppose f is at least $2H+2$ times continuously differentiable. Let $G_H(f)$ denote the Gauss quadrature estimate of f using H nodes. Then there exists a $\lambda' \in [a,b]$ such that*

$$\int_a^b f(\lambda)d\mu(\lambda) - G_H(f) = \frac{f^{(2H+2)}(\lambda')}{(2H+2)!}\int_a^b \chi_H(\lambda)^2 d\mu, \tag{4.21}$$

where $\chi_H(\lambda) = \prod_{h=0}^{H}(\lambda - s_h)$.

Instead of choosing all the nodes to integrate the highest degree polynomial possible correctly, one can prescribe a single node, and then select the remaining H nodes to maximize the degree of the polynomials which the rule correctly integrates. This leads to the approximation,

$$\int_a^b f(\lambda)d\mu(\lambda) \approx v' f(s_0') + \sum_{h=1}^{H} w_h' f(s_h') := \mathrm{GR}_{s_0',H}(f), \tag{4.22}$$

where the s_0' is the prescribed node (often $s_0' = a$ or $s_0' = b$). The remaining s_h' are chosen to maximize the degree of the integration rule, following a similar logic as applied in Gauss quadrature.

For f that are $2H+1$ times continuous differentiable, theorem 4.2 has the modified version, that there exists a $\lambda' \in [a,b]$ such that

$$\int_a^b f(\lambda)d\mu(\lambda) - \mathrm{GR}_{s_0,H}(f) = \frac{f^{(2H+1)}(\lambda')}{(2H+1)!}\int_a^b (\lambda - s_0')\psi_H(\lambda)^2 d\mu, \tag{4.23}$$

where $\psi_H(\lambda) = \prod_{h=1}^{H}(\lambda - s_h')$ (Golub and Meurant, 2009, Equation 6.10).

Stochastic Lanczos Quadrature

We have now described the preliminaries to describe the *stochastic Lanczos quadrature method* (Bai et al., 1996) which can be used to estimate $\log\det \mathbf{K}$. It combines the *Lanczos quadrature method* introduced by Golub and Welsch (1969) with stochastic trace estimation. Let $\mathbf{K} = \tilde{\mathbf{U}}\mathbf{\Lambda}\tilde{\mathbf{U}}^\top$[2] denote the eigendecomposition of $\mathbf{K}$. The goal of the method is to estimate,

$$\operatorname{tr}(f(\mathbf{K})) = \mathbb{E}\big[\sum_{\ell=1}^{L} \mathbf{w}_\ell^\top f(\mathbf{K})\mathbf{w}_\ell\big] \tag{4.24}$$

$$= \sum_{\ell=1}^{L} \mathbb{E}\big[(\tilde{\mathbf{U}}^\top \mathbf{w}_\ell)^\top f(\mathbf{\Lambda})(\tilde{\mathbf{U}}^\top \mathbf{w}_\ell)\big] \tag{4.25}$$

$$= \sum_{\ell=1}^{L} \mathbb{E}\big[\int_a^b f(\lambda) d\mu_\ell(\lambda)\big] \tag{4.26}$$

where $[\lambda_1(\mathbf{K}), \lambda_N(\mathbf{K})] \subset [a,b]$ and μ_ℓ is the (discrete) measure supported on $\{\lambda_n(\mathbf{K})\}_{n=1}^N$ with

$$\mu_\ell(\{\lambda_n(\mathbf{K})\}) = (\tilde{\mathbf{U}}\mathbf{w}_\ell)_n^2 \qquad 1 \le n \le N. \tag{4.27}$$

Quadrature is applied to estimate each integral in eq. (4.26). The nodes of the Gauss quadrature rule with respect to the measure are precisely the eigenvalues of $\mathbf{T}_t$ produced by Lanczos quadrature run with initial vector $\mathbf{w}_\ell/\|\mathbf{w}_\ell\|$, and the weights are the square of the entries of the largest eigenvector in $\mathbf{T}_t$ (Golub and Welsch, 1969). The Gauss-Radau rule can also be computed with the result of Lanczos quadrature after several minor modifications to account for the additional node, see Golub (1973) or Golub and Meurant (2009, Section 6.5.2) for details.

4.2 Iterative Gaussian Process Regression

We now discuss prior work applying the iterative methods discussed in the previous section to Gaussian process regression. For convenience, we recall the log marginal likelihood, its gradient with respect to hyperparameters and the posterior mean and covariance:

$$\mathcal{L}(\theta) = c - \frac{1}{2}\underbrace{\mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y}}_{\text{quadratic term}} - \frac{1}{2}\underbrace{\log\det \mathbf{K}}_{\text{log det. term}}, \tag{4.28}$$

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = -\frac{1}{2}\overbrace{\mathbf{y}^\top \mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{K}^{-1}\mathbf{y}} - \frac{1}{2}\overbrace{\operatorname{tr}\left(\frac{\partial \mathbf{K}}{\partial \theta}\mathbf{K}^{-1}\right)}, \tag{4.29}$$

$$\hat{\mu}(x) = \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{y} \qquad \text{and} \tag{4.30}$$

$$\hat{k}(x,x') = k(x,x') - \mathbf{k}_{x\mathbf{x}}\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x}x'}. \tag{4.31}$$

[2] We use $\tilde{\mathbf{U}}$ in place of $\mathbf{U}$ as was used earlier to avoid confusion with the matrix produced by the Lanczos algorithm.

4.2.1 Model Selection and Mean Estimation with Conjugate Gradients

If the kernel is differentiable with respect to hyperparameters, one can perform gradient descent on the log marginal likelihood by directly estimating the gradient of the log marginal likelihood eq. (4.29), without necessarily estimating the log marginal likelihood (eq. 4.28). Gibbs and MacKay (1997) built on earlier work of Skilling (1993), and proposed selecting model parameters by approximating (4.29) using the method of conjugate gradients. Estimating the quadratic term is straightforward, and is done by computing $\mathbf{v}_t \approx \mathbf{K}^{-1}\mathbf{y}$. To estimate the gradient of the log determinant, Hutchinson's trace estimator was used (section 4.1.3). This leads to solving systems of equations of the form $\tilde{\mathbf{v}}^{(\ell)} \approx \mathbf{K}^{-1}\mathbf{w}_\ell$, with $\mathbf{w}_\ell$ Rademacher distributed, and the method of conjugate gradients can again be employed.

The authors relied on bounds on the quadratic form to decide how many iterations of conjugate gradient to run. In particular, the authors establish the following lemma,

Lemma 4.3 (Gibbs and MacKay 1997, Equation 50)**.** *Suppose* $\mathbf{K} \succ \sigma^2\mathbf{I}$. *For any* $\mathbf{v} \in \mathbb{R}^N$ *define* $\mathbf{r} = \mathbf{y} - \mathbf{K}\mathbf{v}$. *Then*

$$2\mathbf{y}^\top\mathbf{v} - \mathbf{v}^\top\mathbf{K}\mathbf{v} \leq \mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} = \mathbf{r}^\top\mathbf{K}^{-1}\mathbf{r} + 2\mathbf{y}^\top\mathbf{v} - \mathbf{v}^\top\mathbf{K}\mathbf{v} \leq \frac{\mathbf{r}^\top\mathbf{r}}{\sigma^2} + 2\mathbf{y}^\top\mathbf{v} - \mathbf{v}^\top\mathbf{K}\mathbf{v}. \tag{4.32}$$

Proof. The equality follows from expanding the quadratic form $\mathbf{y}^\top\mathbf{K}^{-1}\mathbf{y} = (\mathbf{r}+\mathbf{K}\mathbf{v})^\top\mathbf{K}^{-1}(\mathbf{r}+\mathbf{K}\mathbf{v})$. For the lower bound use that $\mathbf{K}^{-1}$ is positive definite (c.f. eq. 4.3). For the upper bound, use that $\mathbf{K}^{-1} \prec \frac{1}{\sigma^2}\mathbf{I}$ (proposition A.19). □

Subtracting the upper and lower bounds gives the stopping criterion

$$\text{STOP}(\mathbf{r}) = \left(\frac{1}{2\sigma^2}\mathbf{r}^\top\mathbf{r} \leq \varepsilon\right). \tag{4.33}$$

Using this criterion ensures that only a small amount of error can be introduced into the quadratic term $\mathbf{y}\mathbf{K}^{-1}\mathbf{y}$ that appears in the log marginal likelihood. This criterion differs from the conventional stopping criterion for conjugate gradients based on $\|\mathbf{r}\|_2$ by a factor of σ. The authors commented that "Simply fixing the number of conjugate gradient iterations...without checking to see if the approximation was accurate enough led to severe numerical instability" Gibbs and MacKay (1997, page 11). The posterior mean at arbitrary test points can be readily approximated as

$$\tilde{\mu}(x) \approx \mathbf{k}_{x\mathbf{x}}\mathbf{v}_t = \sum_{n=1}^{N} (\mathbf{v}_t)_n k(x_n, x), \tag{4.34}$$

where $\mathbf{v}_t$ is the approximation to $\mathbf{K}^{-1}\mathbf{y}$ obtained via conjugate gradient.

Davies (2015) provided an extended discussion of this approach, as well as comparisons to related methods. Perhaps of most interest to our discussion, Davies (2015) suggests relating the stopping criterion from Gibbs and MacKay (1997) (eq. 4.33) to a bound on the predictive mean.

Lemma 4.4 (Davies 2015, p. 33)**.** *Let $\hat{\mu}(\mathbf{x})$ denote the posterior mean of the Gaussian process. Let $\tilde{\mu}$ be defined by $\tilde{\mu}(x) = \sum_{n=1}^{N} \mathbf{v}_n k(x_n, x)$. Define $\mathbf{r} = \mathbf{y} - \mathbf{K}\mathbf{v}$. Then*

$$|\hat{\mu}(x) - \tilde{\mu}(x)| \leq \sqrt{k(x,x)} \frac{\|\mathbf{r}\|_2}{\sigma}, \tag{4.35}$$

Proof. Write $\hat{\mu}(x) = \langle f_x, \sum_{n=1}^{N} \mathbf{v}_n^* f_{x_n} \rangle_{\mathcal{H}}$ with $\mathbf{v}^* = \mathbf{K}^{-1}\mathbf{y}$ and $\tilde{\mu}(x) = \langle f_x, \sum_{n=1}^{N} \mathbf{v}_n f_{x_n} \rangle_{\mathcal{H}}$. Applying Cauchy-Schwarz to the difference,

$$|\hat{\mu}(x) - \tilde{\mu}(x)| \leq \|f_x\|_{\mathcal{H}} \Big\| \sum_{n=1}^{N} (\mathbf{v}^* - \mathbf{v})_n f_{x_n} \Big\|_{\mathcal{H}} \tag{4.36}$$

$$= \sqrt{k(x,x)} \Big\| \sum_{n=1}^{N} (\mathbf{K}^{-1}\mathbf{r})_n f_{x_n} \Big\|_{\mathcal{H}} \tag{4.37}$$

$$= \sqrt{k(x,x)} \sqrt{\mathbf{r}^\top \mathbf{K}^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}} \mathbf{K}^{-1} \mathbf{r}}. \tag{4.38}$$

Also,

$$\mathbf{K}^{-1/2} \mathbf{K}_{\mathbf{x},\mathbf{x}} \mathbf{K}^{-1/2} \prec \mathbf{K}^{-1/2} \mathbf{K}_{\mathbf{x},\mathbf{x}} \mathbf{K}^{-1/2} + \mathbf{K}^{-1/2} \sigma^2 \mathbf{I} \mathbf{K}^{-1/2} = \mathbf{I}. \tag{4.39}$$

Hence,

$$\mathbf{r}^\top \mathbf{K}^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}} \mathbf{K}^{-1} \mathbf{r} \leq \mathbf{r}^\top \mathbf{K}^{-1} \mathbf{r} \leq \tfrac{1}{\sigma^2} \|\mathbf{r}\|_2^2 \tag{4.40}$$

and the claim follows. □

From lemma 4.4, we see that stopping criterion suggested in MacKay (2003) allows one to bound any additional error incurred in the predictive mean by the procedure.

4.2.2 Estimating the Predictive Variance

Estimating the variance using iterative approaches is not as straightforward as estimating the predictive mean. Computing an estimator of $\tilde{\mu}$ can be done by estimating $\mathbf{v}^* = \mathbf{K}^{-1}\mathbf{y}$ via the method of conjugate gradients. Once an estimate has been computed, the approximate predictive mean at a new test point can be computed with a single inner product (eq. 4.66). A similar approach applied to the variance would involve solving $\mathbf{K}^{-1}\mathbf{k}_{\mathbf{x}x}$ for *each* test point x, and was suggested in Gibbs and MacKay (1997). In many instances, this approach is prohibitively slow. To avoid this cost, Davies (2015) abandoned the use of conjugate gradients and Lanczos iteration entirely when estimating the variance, and instead computed the posterior variances based on fitting a Gaussian process on a subset of the data.

An approach for estimating the posterior covariance directly using the computation done when estimating the log marginal likelihood was proposed in Pleiss et al. (2018). They suggested using the

matrices generated from the Lanczos method and defined,

$$\tilde{k}(x,x') = k(x,x') - \mathbf{k}_{x\mathbf{x}}\mathbf{U}^{(t)}{\mathbf{T}^{(t)}}^{-1}{\mathbf{U}^{(t)}}^{\top}\mathbf{k}_{\mathbf{x}x'}. \tag{4.41}$$

The natural choice of initial vector for Lanczos algorithm is the vector that is being solved against, $\mathbf{k}_{\mathbf{x}x'}$. However, using this vector would require repeatedly running Lanczos method at test time, and so Pleiss et al. (2018) suggest using an average of covariance vectors as the probe, or simply using a cached result from Lanczos method run with any other probe vector (Pleiss, 2020). Pleiss (2020) suggests checking the norm of the residual,

$$\|\mathbf{k}_{\mathbf{x}x'} - \mathbf{K}(\mathbf{U}^{(t)}{\mathbf{T}^{(t)}}^{-1}{\mathbf{U}^{(t)}}^{\top}\mathbf{k}_{\mathbf{x}x'})\|_2 \tag{4.42}$$

as a criterion to ensure accurate solves have been performed. By the same argument as in lemma 4.4

$$|\hat{k}(x,x') - \tilde{k}(x,x')| \leq \sqrt{k(x,x)}\frac{\|\mathbf{k}_{\mathbf{x}x'} - \mathbf{K}(\mathbf{U}^{(t)}{\mathbf{T}^{(t)}}^{-1}{\mathbf{U}^{(t)}}^{\top}\mathbf{k}_{\mathbf{x}x'})\|_2}{\sigma}, \tag{4.43}$$

and so the residual should be scaled accordingly if additive error guarantees on the posterior covariance are desirable.

4.2.3 Estimating the Log Determinant and the Log Marginal Likelihood

Just as the predictive variance is more challenging to efficiently estimate than the predictive mean, the log determinant in eq. (4.28) generally presents more difficulty than the quadratic term. One approach consists of performing some form of series expansion of the logarithm. In particular, finding functions g_i and coefficients c_i such that

$$\operatorname{tr}\log\mathbf{K} \approx \sum_{i=1}^{t} c_i \operatorname{tr}\left(g_i(\mathbf{K})\right). \tag{4.44}$$

Often, each g_i is a polynomial, in which case stochastic trace estimation can be applied to this estimator. Zhang and Leithead (2007) considered a Taylor series expansion of logarithm, while Han et al. (2015) considered a Chebyshev series expansion. Aune et al. (2014) used an estimate developed in Hale et al. (2008) based on applying a trapezoidal quadrature rule to a contour integral representation of the matrix logarithm and applying a change of variable for numerical efficiency.

Ubaru et al. (2017) considered the application of stochastic Lanczos quadrature, as described in section 4.1.5, for estimating $\log\det\mathbf{K}$ in Gaussian process regression. This approach was later adopted and scaled in the `GPyTorch` package (Gardner et al., 2018). In the remainder of this chapter, comparisons will focus on the stochastic Lanczos quadrature estimator, which is known to inherit good convergence properties due to its connection to Gaussian quadrature at least if the smallest eigenvalue of $\mathbf{K}$ is not too close to 0 so that the logarithm and its derivatives are well-behaved on an interval containing the

eigenvalues of $\mathbf{K}$. Practically, this method is widely used in the machine learning community due to its use in `GPyTorch`, making comparisons relevant to current practice.

4.3 Tighter Lower Bounds on the Log Marginal Likelihood

In this section, we describe the lower bounds on the log marginal likelihood that were a primary contribution of Artemev et al. (2021), as well as a slightly sharper version of an earlier upper bound on the log marginal likelihood from Kim and Teh (2018). The central idea of both bounds is to combine ideas from Nyström approximation (see chapter 2) with iterative methods. Unlike the upper bound from Kim and Teh (2018), the lower bound from Artemev et al. (2021) is a suitable surrogate for the log marginal likelihood when using approximate maximum marginal likelihood to select hyperparameters.

There are several motivations for deriving bounds of this form. First, the evidence lower bound can be slow to converge for certain models and datasets (example 3.1). In these instances approximate maximum marginal likelihood model selection with the evidence lower bound leads to a large bias in the selected hyperparameters, generally resulting in modeling the data largely as noise. Optimizing a tighter bound on the log marginal likelihood may alleviate this issue. Second, unlike estimates of the log marginal likelihood or its gradient relying on stochastic trace estimation, the lower bound derived is deterministic. This facilitates faster optimization of the surrogate when selecting hyperparameters, leading to a computational savings. Finally, the use of a lower bound on the log marginal likelihood as a surrogate objective for selecting hyperparameters, as opposed to an estimate that can have bias in either direction, leads to better optimization behavior: if an estimate can be larger than the log marginal likelihood, optimization may select hyperparameters where the estimate is very far from the log marginal likelihood instead of parameters where the log marginal likelihood is reasonably high. See the discussion in Kim and Teh (2018, Appendix L) regarding maximization of upper bounds on the log marginal likelihood.

In the following arguments we bound the log determinant and the quadratic terms from eq. (4.28) separately.

4.3.1 Quadratic Term

The starting point for the bounds derived is the following simple generalization of lemma 4.3, the proof of which is a minor modification of the proof given in the previous section.

Lemma 4.5. *Suppose* $\mathbf{K}, \mathbf{Q} \in S^N_{++}$ *with* $\mathbf{K} \succ \mathbf{Q}$. *For any* $\mathbf{v} \in \mathbb{R}^N$, *and* $\mathbf{r} = \mathbf{y} - \mathbf{K}\mathbf{v}$,

$$2\mathbf{y}^\top \mathbf{v} - \mathbf{v}^\top \mathbf{K}\mathbf{v} \leq \mathbf{y}^\top \mathbf{K}^{-1}\mathbf{y} \leq \mathbf{v}\mathbf{r}^\top \mathbf{Q}^{-1}\mathbf{r} + 2\mathbf{y}^\top \mathbf{v} - \mathbf{v}^\top \mathbf{K}\mathbf{v}. \tag{4.45}$$

Subtracting the upper and lower bounds in lemma 4.5 shows that if $\mathbf{r}^\top \mathbf{Q}^{-1}\mathbf{r}$ is small, we have obtained an accurate estimate of the quadratic term. This suggests the stopping criterion for conjugate

gradient,

$$\text{STOP}(\mathbf{r}) = \left(\frac{1}{2}\mathbf{r}^\top \mathbf{Q}^{-1}\mathbf{r} \leq \varepsilon\right). \tag{4.46}$$

Inspecting algorithm 3, one can see that γ_t is the same as the quantity on the right-hand side of eq. (4.46) if $\mathbf{Q}$ is used as a preconditioner, implying that this criterion can be checked quickly. Also, analogous to lemma 4.4

$$|\hat{\mu}(x) - \tilde{\mu}(x)| = \sqrt{k(x,x)}\sqrt{\mathbf{r}^\top \mathbf{Q}^{-1}\mathbf{r}}, \tag{4.47}$$

where $\tilde{\mu}(x) = \mathbf{k}_{x\mathbf{X}}\mathbf{v}$, so the above stopping criterion is closely related to the quality of predictions, at least relative to the prior variance at a point. The proof of eq. (4.47) is a minor modification of lemma 4.4 presented in the previous section. In the case $M = 0$, we have $\mathbf{Q} = \sigma^2\mathbf{I}$ and in general we have $\mathbf{Q} \succ \sigma^2\mathbf{I}$, so lemma 4.5 and eq. (4.47) generalize lemma 4.3 and lemma 4.4 respectively, and improve upon them for $M > 0$. This means that to obtain the same guarantees on quantities of interest, the bounds in lemma 4.5 and eq. (4.47) will generally require fewer iterations of conjugate gradients than lemma 4.3 and lemma 4.4.

4.3.2 Log Determinant Term

We now turn to $\log\det\mathbf{K}$. The determinant and trace are related through the following inequalities,

Proposition 4.6 (Arithmetic-Geometric inequality for log determinant; Vakili et al., 2021, Lemma 1)**.** *For* $\mathbf{A} \in S^N_{++}$

$$\log\det(\mathbf{A}) \leq N\log\left(\text{tr}(\mathbf{A})/N\right). \tag{4.48}$$

Proof. Write the log determinant as a product of eigenvalues (proposition A.13), and apply the arithmetic-geometric mean inequality. □

Proposition 4.7. *For* $\mathbf{A} \in S^N_+$, $\log|\mathbf{I} + \mathbf{A}| \leq \text{tr}(\mathbf{A})$.

Proof. Apply proposition 4.6 to $\mathbf{I} + \mathbf{A}$ and use the inequality $\log(1+a) \leq a$ on the result. □

We now turn to a derivation of the inequality

$$\log\det\mathbf{K} \leq \log\det\mathbf{Q} + \frac{1}{\sigma^2}\text{tr}\left(\mathbf{K} - \mathbf{Q}\right), \tag{4.49}$$

which is used in the variational evidence lower bound (eq. 2.52). Along the way we re-derive a tighter lower bound on $\log\det\mathbf{K}$ given in Shi et al. (2020, Appendix A).

$$\begin{aligned}
\log\det\mathbf{K} &= \log\det\mathbf{Q} + \log\det(\mathbf{Q}^{-1/2}\mathbf{K}\mathbf{Q}^{-1/2}) && (4.50)\\
&= \log\det\mathbf{Q} + \log\det(\mathbf{I} + \mathbf{Q}^{-1/2}(\mathbf{K} - \mathbf{Q})\mathbf{Q}^{-1/2}) && (4.51)\\
&\leq \log\det\mathbf{Q} + \text{tr}\left(\mathbf{Q}^{-1}(\mathbf{K} - \mathbf{Q})\right). && (4.52)
\end{aligned}$$

The first equality uses that the determinant is multiplicative. The inequality in eq. (4.52) combines proposition 4.7 with the cyclic property of trace (proposition A.10). Because $\log(1+a) \leq a$ is only tight when $a \approx 0$, this is only tight when $\mathbf{Q}^{-1/2}(\mathbf{K}-\mathbf{Q})\mathbf{Q}^{-1/2}$ has exclusively small eigenvalues. Equation (4.52) is one of the terms in the lower bound given in Shi et al. (2020, Appendix A) and can be computed in $O(N^2M)$. Using lemma 3.7,

$$\begin{aligned}\operatorname{tr}\left(\mathbf{Q}^{-1}(\mathbf{K}-\mathbf{Q})\right) &\leq \lambda_1(\mathbf{Q}^{-1})\operatorname{tr}(\mathbf{K}-\mathbf{Q}) \\ &\leq \frac{1}{\sigma^2}\operatorname{tr}(\mathbf{K}-\mathbf{Q})\,. \end{aligned} \tag{4.53}$$

Equation (4.53) is the term from the evidence lower bound stated in eq. (4.49) and can be computed in $O(NM^2)$.

Lemma 4.8. *For* $\mathbf{K},\mathbf{Q} \in S^N_{++}$ *with* $\mathbf{K} \succ \mathbf{Q} \succ \sigma^2\mathbf{I}$

$$\log\det\mathbf{K} \leq \log\det\mathbf{Q} + N\log\left(\frac{\operatorname{tr}\left(\mathbf{Q}^{-1}\mathbf{K}\right)}{N}\right) \tag{4.54}$$

$$\leq \log\det\mathbf{Q} + N\log\left(1+\frac{\operatorname{tr}(\mathbf{K}-\mathbf{Q})}{N\sigma^2}\right). \tag{4.55}$$

Remark 4.9. *Equation (4.54) does not assume* $\mathbf{Q} \succ \sigma^2\mathbf{I}$ *but eq. (4.55) does. In our application* $\mathbf{Q}-\sigma^2\mathbf{I} = \mathbf{Q}_{\mathbf{x},\mathbf{x}} \in S^N_+$.

Remark 4.10. *Equation (4.54) improves upon eq. (4.52) with the same computational cost* $O(N^2M)$*, while eq. (4.55) improves upon eq. (4.53) with the same computational cost* $O(NM^2)$*. This can be seen by applying* $\log(1+a) \leq a$ *in both cases.*

Proof of lemma 4.8. The proof is identical to the proof of eq. (4.49) presented above after replacing proposition 4.7 with proposition 4.6 in eq. (4.52). □

4.3.3 A Lower Bound on the Log Marginal Likelihood

Combining lemmas 4.5 and 4.8 gives a lower bound on the log marginal likelihood.

Lemma 4.11. *Let* $\mathbf{K}$ *as in eq. (1.38),* $\mathbf{Q}$ *as in eq. (2.52),* $C = -\frac{N}{2}\log 2\pi$*, then for any* $\mathbf{v} \in \mathbb{R}^N$

$$\mathcal{L}(\theta) \geq C - \frac{1}{2}\left(\mathbf{r}^\top\mathbf{Q}^{-1}\mathbf{r} + 2\mathbf{y}^\top\mathbf{v} - \mathbf{v}^\top\mathbf{K}\mathbf{v}\right) - \frac{1}{2}\left(\log\det\mathbf{Q} + N\log\left(\frac{\operatorname{tr}\left(\mathbf{Q}^{-1}\mathbf{K}\right)}{N}\right)\right), \tag{4.56}$$

$$\geq C - \frac{1}{2}\left(\mathbf{r}^\top\mathbf{Q}^{-1}\mathbf{r} + 2\mathbf{y}^\top\mathbf{v} - \mathbf{v}^\top\mathbf{K}\mathbf{v}\right) - \frac{1}{2}\left(\log\det\mathbf{Q} + N\log\left(1+\frac{\operatorname{tr}(\mathbf{K}-\mathbf{Q})}{N\sigma^2}\right)\right), \tag{4.57}$$

where $\mathbf{r} = \mathbf{y} - \mathbf{K}\mathbf{v}$.

Remark 4.12. *When* $\mathbf{v} = \mathbf{0}$ *or* $\mathbf{v} = \mathbf{K}^{-1}\mathbf{y}$*, the bounds in lemma 4.11 are at least as tight as eq. (2.10). For general* $\mathbf{v}$*, they may be smaller than the lower bound* $\mathcal{L}(\mathbf{z},\theta)$ *in eq. (2.10). In practice, we will*

select $\mathbf{v}$ *with the method of conjugate gradients in which case they are almost always larger than the corresponding* $\mathcal{L}(\mathbf{z}, \theta)$.

Remark 4.13. *For a fixed* $\mathbf{v}$*, the computational cost of eq.* (4.56) *is* $O(N^2M)$ *while the computational cost of eq.* (4.57) *is* $O(N^2 + NM^2)$.

Comparison to the Lower Bound in Davies (2015) A similar idea was developed in Davies (2015, Section 6) inspired by variational approaches. The bounds derived in Davies (2015) are generally quite loose due to the handling of the log determinant term in the log marginal likelihood. In particular, the bounds in lemma 4.11 are generally tighter than the evidence lower bound discussed in Titsias (2009), whereas the bound in Davies (2015) can be substantially looser. Crucially, we recover the log marginal likelihood when the number of inducing points is sufficiently large (e.g. when $\mathbf{x} \subset \mathbf{z}$), unlike the approach suggested in Davies (2015).

Lemma 4.14 (Davies, 2015, Equations 218-219)**.** *Let* $\mathbf{K}$ *as in eq.* (1.38)*,* $C = -\frac{N}{2}\log 2\pi$*, and* $\mathbf{z} \subset \mathbf{x}$*. For any* $\mathbf{v} \in \mathbb{R}^N$

$$\begin{aligned}\mathcal{L}(\theta) \geq C - \frac{1}{2}\left(\tfrac{\mathbf{r}^\top\mathbf{r}}{\sigma^2} + 2\mathbf{y}^\top\mathbf{v} - \mathbf{v}^\top\mathbf{K}\mathbf{v}\right) \\ - \frac{1}{2}\left(\log\det(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I}) + (N-M)\log\sigma^2 + \tfrac{1}{\sigma^2}\operatorname{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{K}_{\mathbf{x},\mathbf{z}}(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^\top\right)\right). \quad (4.58)\end{aligned}$$

As there are fixable errors in the proof given in Davies (2015), a simplified proof of lemma 4.14 is included.

Proof of lemma 4.14. Let $\bar{\mathbf{z}} = \mathbf{x} \setminus \mathbf{z}$. Since $\mathbf{z} \subset \mathbf{x}$, the matrix $\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I}$ is a principal sub-matrix of $\mathbf{K} = \mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma^2\mathbf{I}$. By the block-matrix determinant lemma (proposition A.14)

$$\log\det\mathbf{K} = \log\det(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I}) + \log\det(\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}} + \sigma^2\mathbf{I} - \mathbf{K}_{\bar{\mathbf{z}},\mathbf{z}}^\top(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}}). \quad (4.59)$$

For compactness, define $\widetilde{\mathbf{K}}_{\bar{\mathbf{z}},\bar{\mathbf{z}}} = \mathbf{K}_{\bar{\mathbf{z}},\mathbf{z}}^\top(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}}$. Factoring out $\sigma^2\mathbf{I}$,

$$\begin{aligned}\log\det(\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}} + \sigma^2\mathbf{I} - \widetilde{\mathbf{K}}_{\bar{\mathbf{z}},\bar{\mathbf{z}}}) &= (N-M)\log\sigma^2 + \log\det(\mathbf{I} + \tfrac{1}{\sigma^2}(\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}} - \widetilde{\mathbf{K}}_{\bar{\mathbf{z}},\bar{\mathbf{z}}})) && (4.60)\\ &\leq (N-M)\log\sigma^2 + \tfrac{1}{\sigma^2}\operatorname{tr}\left(\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}} - \widetilde{\mathbf{K}}_{\bar{\mathbf{z}},\bar{\mathbf{z}}}\right) && (4.61)\\ &\leq (N-M)\log\sigma^2 + \tfrac{1}{\sigma^2}\operatorname{tr}\left(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{K}_{\mathbf{x},\mathbf{z}}(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^\top\right). && (4.62)\end{aligned}$$

The first inequality uses that for $\mathbf{A}$ positive semi-definite, $\log\det(\mathbf{I} + \mathbf{A}) \leq \operatorname{tr}(\mathbf{A})$ (proposition 4.7). The second uses that each diagonal entry of $\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{K}_{\mathbf{x},\mathbf{z}}^\top(\mathbf{K}_{\mathbf{z},\mathbf{z}} + \sigma^2\mathbf{I})^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}$ is non-negative: the diagonal entries are the posterior variance of a Gaussian process after observing inputs at $\mathbf{z}$, evaluated at $\mathbf{x}_i$ and that the diagonal entries of $\mathbf{K}_{\bar{\mathbf{z}},\bar{\mathbf{z}}} - \widetilde{\mathbf{K}}_{\bar{\mathbf{z}},\bar{\mathbf{z}}}$ are a subset of these variances. □

When $\mathbf{z} = \mathbf{x}$ and $\mathbf{v} = \mathbf{K}^{-1}\mathbf{y}$ the bound in lemma 4.14 differs from the log marginal likelihood by the sum of the posterior variance at each data point divided by the noise ratio, and in particular does not

recover the log marginal likelihood. We could instead use eq. (4.61), which does coincide with the log marginal likelihood when $\mathbf{z} = \mathbf{x}$, but will converge slowly as it depends on the posterior variance of a Gaussian process fit on the $\mathbf{z}$ contracting. This is reminiscent of a subset-of-data approach, instead of an inducing point approach.

4.3.4 A Refinement of the Upper Bound on the Log Marginal Likelihood of Kim and Teh (2018)

Combining proposition 2.16 and lemma 4.5 yields an upper bound on the log marginal likelihood, that is a minor refinement of a bound proposed in Kim and Teh (2018):

Lemma 4.15. *Let* $\mathbf{K}$ *as in eq.* (1.38)*,* $\mathbf{Q}$ *as in eq.* (2.52)*,* $C = -\frac{N}{2}\log 2\pi$*, then for any* $\mathbf{v} \in \mathbb{R}^N$

$$\mathcal{L}(\theta) \leq C - \frac{1}{2}\left(2\mathbf{y}^\top \mathbf{v} - \mathbf{v}^\top \mathbf{K}\mathbf{v}\right) - \frac{1}{2}\left(\log\det \mathbf{Q} + \log(1 + \tfrac{\mathrm{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2})\right). \tag{4.63}$$

The difference between this bound and Kim and Teh (2018, Equations 3 and 4) is the term

$$-\frac{1}{2}\log(1 + \tfrac{\mathrm{tr}(\mathbf{K}_{\mathbf{x},\mathbf{x}} - \mathbf{Q}_{\mathbf{x},\mathbf{x}})}{\lambda_1(\mathbf{Q}_{\mathbf{x},\mathbf{x}}) + \sigma^2}) \leq 0, \tag{4.64}$$

which is often small. This upper bound could be used for upper bounding the Kullback-Leibler divergence as in proposition 2.18 or for other diagnostic purposes.

4.4 Model Selection with Tighter Lower Bounds on the Log Marginal Likelihood

In this section, we discuss the method for approximate maximum marginal likelihood proposed in Artemev et al. (2021). The core idea is to maximize the lower bound given in lemma 4.11, and we refer to the overall procedure as *conjugate gradient lower bound maximization* (CGLB). Relative to evidence lower bound maximization, the advantage of conjugate gradient lower bound maximization is that the lower bound is generally tighter, and therefore may introduce less bias into selection of hyperparameters. Particularly, the tightness of the lower bound will make the procedure slightly more robust to using too few inducing points. Relative to iterative approaches, the advantage of conjugate gradient lower bound maximization is that the objective function is a deterministic lower bound on the log marginal likelihood, which can improve the reliability of the optimization procedure.

The key considerations are:

- How to select the auxiliary vector $\mathbf{v}$?
- How to select the inducing inputs $\mathbf{z}$?
- How to make predictions once a set of hyperparameters is selected?

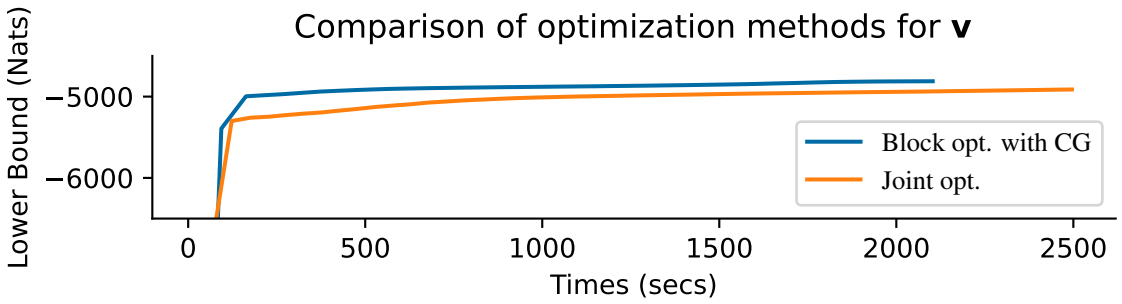


Fig. 4.1 Comparison of a block optimization procedure using the method of conjugate gradients for $\mathbf{v}$ and L-BFGS for θ (blue) to joint optimization of the auxiliary vector $\mathbf{v}$ and hyperparameters θ with L-BFGS (joint, orange) on the `elevators` dataset with $M = 750$. The block optimization procedure is beneficial in terms of wall-clock time.

We begin with the question of how to select $\mathbf{v}$, which is essential to the tightness of the bound. A naive approach is to optimize $\mathbf{v}$ jointly with hyperparameters using a gradient-based optimizer. While this is feasible, it adds N dimensions to the optimization problem. Perhaps unsurprisingly given the topic of this chapter, the alternative approach we take is to use the method of conjugate gradients to obtain a $\mathbf{v} \approx \mathbf{K}^{-1}\mathbf{y}$, as $\mathbf{v} = \mathbf{K}^{-1}\mathbf{y}$ maximizes the lower bound in lemma 4.11. Figure 4.1 shows an illustrative example of the difference in optimization speed using the method of conjugate gradients for $\mathbf{v}$ and L-BFGS for θ as opposed to joint optimization with L-BFGS for $\{\mathbf{v}, \theta\}$. The former procedure gives a clear benefit in terms of the wall-clock performance of optimization of the objective function, despite the higher cost per update of the hyperparameters θ.

4.4.1 Training Procedure

The training approach consists of block optimization, where the lower bound in eq. (4.57) is first maximized with respect to the auxiliary vector $\mathbf{v}$ via the method of conjugate gradients, then θ is optimized. In order to fully specify the conjugate gradient procedure used, we need to specify a stopping criterion, a preconditioner and an initial vector.

Stopping Criterion As a stopping criterion we use eq. (4.46). Subtracting the upper and lower bounds on the quadratic form in lemma 4.5, shows that this criterion ensures that more iterations of conjugate gradient could not improve the lower bound by more than ε. As noted earlier, this coincides with the stopping criterion proposed in Gibbs and MacKay (1997) in the case $M = 0$, and is less strict for $M > 0$, from which we expect a computational savings. If $\mathbf{Q}$ is the preconditioner used, we have $\gamma_t = \mathbf{r}_t^\top \mathbf{Q}^{-1}\mathbf{r}_t$ for the γ_t in algorithm 3. Hence, very little additional computation needs to be used to check this criterion.

Most importantly, unlike commonly used stopping criterion such as $\|\mathbf{r}\|_2^2 \leq \varepsilon'$, the criterion eq. (4.46) ensures that we approximate this term in the log marginal likelihood well *uniformly over hyperparame-*

ters, which we expect will reduce bias in hyperparameter selection. In particular, the bias in the lower bound in lemma 4.5 is $\mathbf{r}^\top \mathbf{K}^{-1}\mathbf{r}$, which in the worst case can be as large as $\|\mathbf{r}\|_2^2/(2\sigma^2)$. Unless the magnitude of σ^2 is accounted for in the stopping criterion, this means a bias towards small values of σ^2 can be introduced when optimizing the lower bound. Similarly, in the worst case the upper bound in lemma 4.3 can have a bias of a similar magnitude in the opposite direction, which would bias optimization to larger likelihood variances.

By instead ensuring $\frac{1}{2}\mathbf{r}^\top \mathbf{Q}^{-1}\mathbf{r} \leq \varepsilon$, we ensure the bias we introduce into estimation of the log marginal likelihood from stopping conjugate gradients early in the lower bound is uniformly bounded by ε over all hyperparameter settings.

Preconditioner and Selecting Inducing Inputs We use a preconditioner based on a Nyström approximation, with M inducing points denoted by $\mathbf{z}$. This approach has been investigated in, for example, Cutajar et al. (2016). In the first iteration, we select the inducing points via algorithm 1. As noted in chapter 3 this is equivalent to performing an incomplete Cholesky decomposition with pivoting, which has been used as a preconditioner in Gaussian process regression (Gardner et al., 2018).

We use the same matrix for estimating the log determinant lemma 4.8, preconditioning, and computing the bound on the quadratic form lemma 4.5. Gradients with respect to the inducing points $\mathbf{z}$ can be computed with automatic differentiation, and we can optimize the lower bound in eq. (4.57) with respect to $\mathbf{z}$. We optimize $\mathbf{z}$ jointly with θ in subsequent iterations. While this approach lacks a variational interpretation we found it works well in practice.

Initial Vector In the first iteration it is unclear how to select an initial vector, so we take $\mathbf{v}_0 = \mathbf{0}$. We assume the optimizer used is local, in the sense that it is likely to select values of the hyperparameters similar to the previous setting. Assuming the kernel is continuous in the hyperparameters, at iteration $\eta + 1$ of hyperparameter selection and making dependence on kernel hyperparameter explicit, we want to compute

$$\mathbf{K}_{\theta_{\eta+1}}^{-1}\mathbf{y} = (\mathbf{K}_{\theta_\eta} + \mathbf{E}_{\eta+1})^{-1}\mathbf{y}, \tag{4.65}$$

where $\mathbf{E}_{\eta+1} = \mathbf{K}_{\theta_{\eta+1}} - \mathbf{K}_{\theta_\eta}$. $\mathbf{E}_{\eta+1}$ should be small by the assumption that the optimizer searches locally for new hyperparameters. Since inversion is Lipschitz continuous for matrices with eigenvalues in $[\sigma^2, \infty)$ with the constant depending on σ^2, the solution to eq. (4.65) is a perturbed version of the solution found in the previous iteration of hyperparameter optimization. As a result, particularly as optimization begins to converge and hyperparameters change slowly, very few iterations of conjugate gradients need to be run in each step of hyperparameter optimization. This behavior is illustrated in figure 4.2.

Reusing past solutions to similar problems leads to only a small computational savings in approaches that rely on stochastic trace estimation (Gardner et al., 2018; Gibbs and MacKay, 1997). This is because at each iteration the vectors solved against when estimating the log determinant are re-sampled.

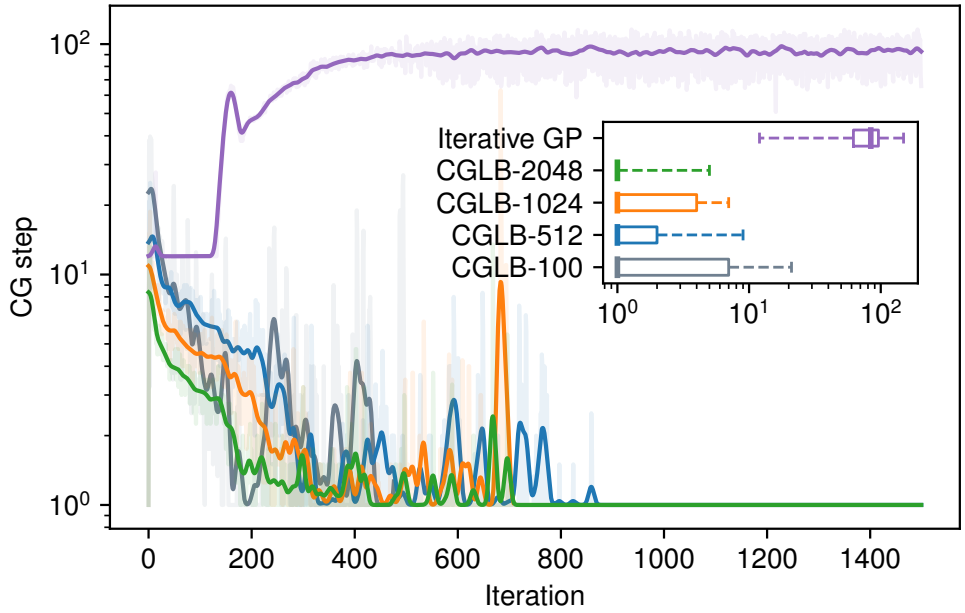


Fig. 4.2 Number of steps of the method of conjugate gradients run during conjugate gradient lower bound maximization plotted against training iteration on the protein dataset. Inset is the distribution of the number of iterations of conjugate gradient run throughout training. Iterative GP indicates the approach taken in Gardner et al. (2018) based on stochastic trace estimation, stochastic Lanczos quadrature and conjugate gradients. This method re-samples vectors for stochastic trace estimation and Lanczos quadrature, so it does not reuse computation. Conjugate gradient lower bound maximization, shown with different rank preconditioners ranging from $M = 100$ to $M = 2048$, generally needs to only run a few iterations of the method of conjugate gradients per hyperparameter update after the first couple hundred iterations of L-BFGS, as at that point hyperparameters change reasonably slowly.

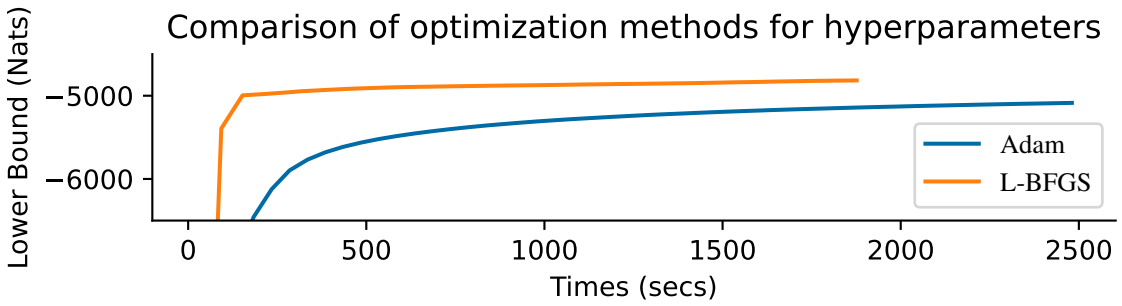


Fig. 4.3 A comparison of the objective function (eq. 4.57) plotted against time with Adam (blue) and L-BFGS (orange) used for hyperparameters selection with conjugate gradient lower bound maximization. A learning rate of 0.1 is used for Adam with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ (the default momentum parameters in `tensorflow`). More careful tuning of the learning rate and momentum parameters may improve the speed of convergence. However, this generally involves the user choosing more parameters, and closing the performance gap is difficult even with careful tuning. This example is on the `elevators` dataset using $M = 750$ inducing points to form $\mathbf{Q}$.

Optimizing Hyperparameters We use L-BFGS for optimizing the hyperparameters, θ. Because our approximation to the gradient is the gradient of our approximation to the objective function[3] and there is no stochasticity in either the objective or its gradients, standard quasi-Newton optimizers with line search (e.g. in `python` the `scipy`, Virtanen et al., 2020, implementation of L-BFGS) work without modification. Such methods are often the default for maximum marginal likelihood estimation in Gaussian process regression and in evidence lower bound maximization, see for example, the `GPflow` documentation (Matthews et al., 2017), whereas first order methods without line search have been commonly used in iterative linear algebra in Gaussian processes. This may be in part due to the stochasticity and the objective function due to stochastic trace estimation, and the gradient of the approximate log marginal likelihood not equaling the approximation of the gradient of the log marginal likelihood, which occurs due to estimating the log marginal likelihood and its gradient with different approaches.

In practice, we observe a large benefit from using L-BFGS as opposed to Adam (Kingma and Ba, 2015) for hyperparameter selection (figure 4.3). Additionally, there is no need to specify an initial learning rate as line search handles this automatically, and the `scipy` implementation has sensible default convergence criteria that often mean a high number of maximum iterations can be specified, and the algorithm will stop early when hyperparameters converge. This improves the ease-of-use of the model selection process.

While both probabilistic line search methods (Mahsereci and Hennig, 2017) and stochastic quasi-Newton methods (Schraudolph et al., 2007) have been developed, they do not appear to have been widely adopted for model selection in Gaussian process regression. Gibbs and MacKay (1997) report using (non-linear) conjugate gradients to select hyperparameters. Gibbs and MacKay (1997) do not

[3] Although we ignore the dependence of $\mathbf{v}$ on the hyperparameters, if $\mathbf{v}$ is sufficiently close to its optimum in each iteration this has minimal effect.

report issues with instability or irregular termination, although this is surprising as non-linear conjugate gradients is not by default robust to stochasticity in the estimated gradient. It is unclear whether and how line search was used (as is typically done in conjugate gradients) given the authors do not propose an approximation for the log marginal likelihood, but only for its gradient. Gardner et al. (2018) rely on Adam (Kingma and Ba, 2015), which generally requires more iterations for hyperparameters to converge (figure 4.3). Wang et al. (2019) used a handful of steps of L-BFGS, fixing the matrix $\mathbf{W}$ used for stochastic trace estimation during each line search to avoid unsuccessful line searches, then switched to using Adam, presumably to avoid bias introduced by estimating the log determinant and its gradient without re-sampling $\mathbf{W}$. Subsequent to our work, Wenger et al. (2021) suggested using the preconditioner as a control variate in the approach taken by Gardner et al. (2018) and, perhaps surprisingly, reported good results directly applying L-BFGS with line search despite the remaining stochasticity in the objective function and its gradients.

4.4.2 Making Predictions

As the lower bound from eq. (4.57) is derived via a direct calculation, as opposed to based on variational principles it is unclear how to define an approximation to the posterior to make predictions with. In the case of a moderately large dataset, it may be the case that even though computing a Cholesky decomposition many times in order to select hyperparameters is an unacceptable computational cost, computing it a single time to make predictions is reasonable. As a more scalable approach, we advocate using the predictive mean function

$$\tilde{\mu}(x) = \mathbf{k}_{x\mathbf{x}}\mathbf{v} + \mathbf{q}_{x\mathbf{x}}\mathbf{Q}^{-1}(\mathbf{y} - \mathbf{K}\mathbf{v}), \tag{4.66}$$

where we recall $\mathbf{q}_{x\mathbf{x}} = \mathbf{k}_{x\mathbf{z}}\mathbf{K}_{\mathbf{z},\mathbf{z}}^{-1}\mathbf{K}_{\mathbf{x},\mathbf{z}}^{\top}$. The first term in eq. (4.66) is the mean often used when applying the method of conjugate gradients (eq. 4.34). The second term is the variational Gaussian process mean (eq. 2.48), replacing the data vector $\mathbf{y}$ with the residual vector $\mathbf{r}$, that is not accounted for in the first term. Equation (4.66) recovers the posterior mean in the limiting cases that 1. $\mathbf{v} = \mathbf{K}^{-1}\mathbf{y}$, so that conjugate gradient has succeeded or 2. $\mathbf{Q} = \mathbf{K}$, so that sparse approximation has succeeded. These are precisely the cases where our bound on the quadratic form lemma 4.5 is tight. Any a posteriori analysis of the mean of a Nyström based (e.g. proposition 2.15) approach can be used to upper and lower bound the error between eq. (4.66) and the posterior mean with minor modifications. For the predictive variance and covariance, we fall back onto the sparse variational Gaussian process regression estimate (eq. 2.48), although iterative approaches such as eq. (4.41) could be used instead.

4.5 Empirical Behavior of Conjugate Gradient Lower Bound Maximization

In the absence of an exhaustive theoretical understanding of when conjugate gradient lower bound maximization works, we address the question of *will it work?* through running the method on several datasets. We compare hyperparameter selection via conjugate gradient lower bound maximization (CGLB), evidence lower bound maximization (SGPR) and the iterative method considered in Gardner et al. (2018); Wang et al. (2019) which uses conjugate gradients and stochastic trace estimation to estimate the gradient of the log marginal likelihood and conjugate gradients together with stochastic Lanczos quadrature to estimate the log marginal likelihood (Iterative GP). For conjugate gradient lower bound maximization, we focus on eq. (4.57) due to its lower computational cost than eq. (4.56).

4.5.1 Experimental Details

We consider several regression datasets all of which are from the UCI repository (Dua and Graff, 2017) except for `kin40k` (Ghahramani, 1996). The methods are compared on the basis of root-mean-square error (RMSE) and negative log predictive density (NLPD) on held-out data to assess predictive performance.

Root-mean-square error and negative log predictive density conflate the quality of the approximation with modeling assumptions (see our discussion of what an approximation 'working' means in section 1.4). To directly assess the extent to which an optimization procedure approximates maximum marginal likelihood model selection, we compare the log marginal likelihood of the model with the hyperparameters selected by each method, computed directly with Cholesky decomposition. This is only feasible on the smaller datasets considered. All experiments were run on a single Tesla V100-32GB GPU. The code for experiments can be found at https://github.com/awav/CGLB, while the implementation of conjugate gradient lower bound maximization is included in the `GPflow` package (Matthews et al., 2017).

Data Preparation

We randomly split each dataset into a training set consisting of 67% of examples, and a test set consisting of the remaining 33%. We run 3 seeds in all experiments, with each seed corresponding to a different random split of the dataset. Each input dimension is standardized to have mean 0 and variance 1 within the training set. Similarly, the training outputs are standardized to have 0 mean and variance 1. We apply the same standardization to test data when making predictions, using the statistics computed on the training data. All metrics reported are on the standardized data, so that we would expect a model predicting the zero function to have root-mean-square error close to 1.0. Datasets are downloaded using `Bayesian Benchmarks` (Salimbeni, 2019).

The datasets range in size from containing just over 10000 training examples to just over 40000 training examples. The number of training examples used in each dataset, as well as the dimensionality of the covariates can be found in the left-most column of table 4.1.

Model Class and Initialization of Parameters

We run experiments with a Matérn 3/2 kernel, with independently learned lengthscales along each input dimension (i.e. automatic relevance detection). While in the derivations we assumed the prior mean is 0, in experiments we take the prior mean to be a constant function that is learned as a hyperparameter, in which case one must substitute $\mathbf{y} \to \mathbf{y} - \mu(\mathbf{x})$ in the estimates of the log marginal likelihood, and make similar considerations during predictions. This is the same experimental setup considered in Wang et al. (2019).

All kernel lengthscales, the kernel variance and the likelihood variance are initialized at 1.0. The prior mean is initialized at 0. We use soft-plus constraints on the lengthscales and likelihood, lower bounding them at 1×10^{-6} for evidence lower bound maximization and conjugate gradient lower bound maximization and 1×10^{-4} for iterative GP, as we found this improved numerical stability.[4] Experiments are run using double precision.

Conjugate Gradient Lower Bound Maximization For conjugate gradient lower bound maximization we vary the number of inducing points between 512 and 4096. As described in the previous section, we use the L-BFGS optimizer (Liu and Nocedal, 1989) for either 2000 steps or until the optimizer stops due to a small projected gradient norm or being unable to find a point that improves upon the current value during line-search.[5] We use the stopping criterion eq. (4.46) with $\varepsilon = 1.0$ during training and $\varepsilon = 1 \times 10^{-3}$ before making predictions. Jitter of 1×10^{-6} is used (section 2.6) to avoid the Cholesky decomposition failing when estimating the log determinant.

For small datasets ($n < 20000$), we use the `GPflow` implementation of conjugate gradient lower bound maximization as we found this to be faster than the `GPyTorch` implementation. However, for larger datasets we use the `GPyTorch` kernel methods with `KeOps` (Charlier et al., 2021) to perform matrix-vector operations without forming the entire kernel matrix at any time to reduce memory requirements.

Evidence Lower Bound Maximization For experiments with evidence lower bound maximization we use `GPflow` (Matthews et al., 2017). Otherwise, the experimental setup mirrors the procedure used in conjugate gradient lower bound maximization; we vary the number of inducing points between 512 and 4096; we use L-BFGS to optimize $\{\mathbf{z}, \theta\}$; jitter of 1×10^{-6} is used.

Iterative Gaussian Process Regression We follow the procedure described in Wang et al. (2019) to train Iterative GP. We first pre-train the model on a subset of 10000 observations with 10 iterations of

[4] A lower bound of 1×10^{-4} is the default in `GPyTorch`.

[5] These are the default criterion in `scipy`

L-BFGS. Then we run 10 iterations of Adam optimizer (Kingma and Ba, 2015) on the same subset, followed by 2000 iterations of Adam with 0.1 learning rate on the full dataset. Ten independent Gaussian vectors $\{\mathbf{w}_\ell\}_{\ell=1}^{10}$ are sampled for stochastic trace estimation. In order to run L-BFGS, the vectors $\{\mathbf{w}_\ell\}_{\ell=1}^{10}$ are fixed along each line search, as suggested in the documentation for `GPyTorch` (Gardner et al., 2018). We use the `GPyTorch` implementation (Gardner et al., 2018), and use the default stopping criterion in the package: $\|\mathbf{r}\|_2/N \leq 1.0$ during hyperparameter selection and $\|\mathbf{r}\|_2/N \leq 0.01$ for making predictions. A rank-100 preconditioner is used for both the method of conjugate gradients and the Lanczos algorithm. Following Wang et al. (2019), we use the Lanczos variance estimator eq. (4.41) introduced in Pleiss et al. (2018) for computing the predictive negative log probability density for the method using stochastic trace estimation (Iterative GP).

4.5.2 Experimental Results

In figure 4.4, we compare the predictive performance obtained with hyperparameters found by conjugate gradient lower bound maximization to the performance when hyperparameters are selected via evidence lower bound maximization or with Iterative GP. Comparisons are made on the `protein` (top) and `keggundirected` (middle) UCI datasets (Dua and Graff, 2017), as well as the `kin40k` dataset (bottom) (Ghahramani, 1996).

Comparison of Conjugate Gradient Lower Bound Maximization to Evidence Lower Bound Maximization Conjugate gradient lower bound maximization leads to better performance for a given amount of computation, both in terms of root-mean-square error and negative log probability density on `protein` and `kin40k` as compared to evidence lower bound maximization (figure 4.4). On both of these datasets many (more than 4096) inducing points are needed for evidence lower bound maximization to select hyperparameters similar to those favored by the log marginal likelihood, and so maximization of the evidence lower bound leads to under-fitting. Indeed, van der Wilk et al. (2020) suggested as many as 10000 inducing points might be needed for `kin40k` to obtain a reasonable approximation to the posterior. It seems by reducing the bias, as well as using a predictive mean quite close to the full predictive mean, conjugate gradient lower bound maximization results in a large benefit on these datasets.

On the other hand, on `keggundirected` no real benefit is observed in terms of test root-mean-square error or negative log probability density for conjugate gradient lower bound maximization as opposed to evidence lower bound maximization. In fact, the evidence lower bound maximization models with few inducing variables seem to obtain better metrics than either conjugate gradient lower bound maximization or evidence lower bound maximization with more inducing points. This may indicate that the maximum marginal likelihood hyperparameters are suboptimal on this dataset for the class of models considered, and the biases introduced by less accurate approximations to the log marginal likelihood lead to better performance.

In terms of finding hyperparameters with high log marginal likelihood, conjugate gradient lower bound maximization generally outperformed evidence lower bound maximization on the smaller datasets

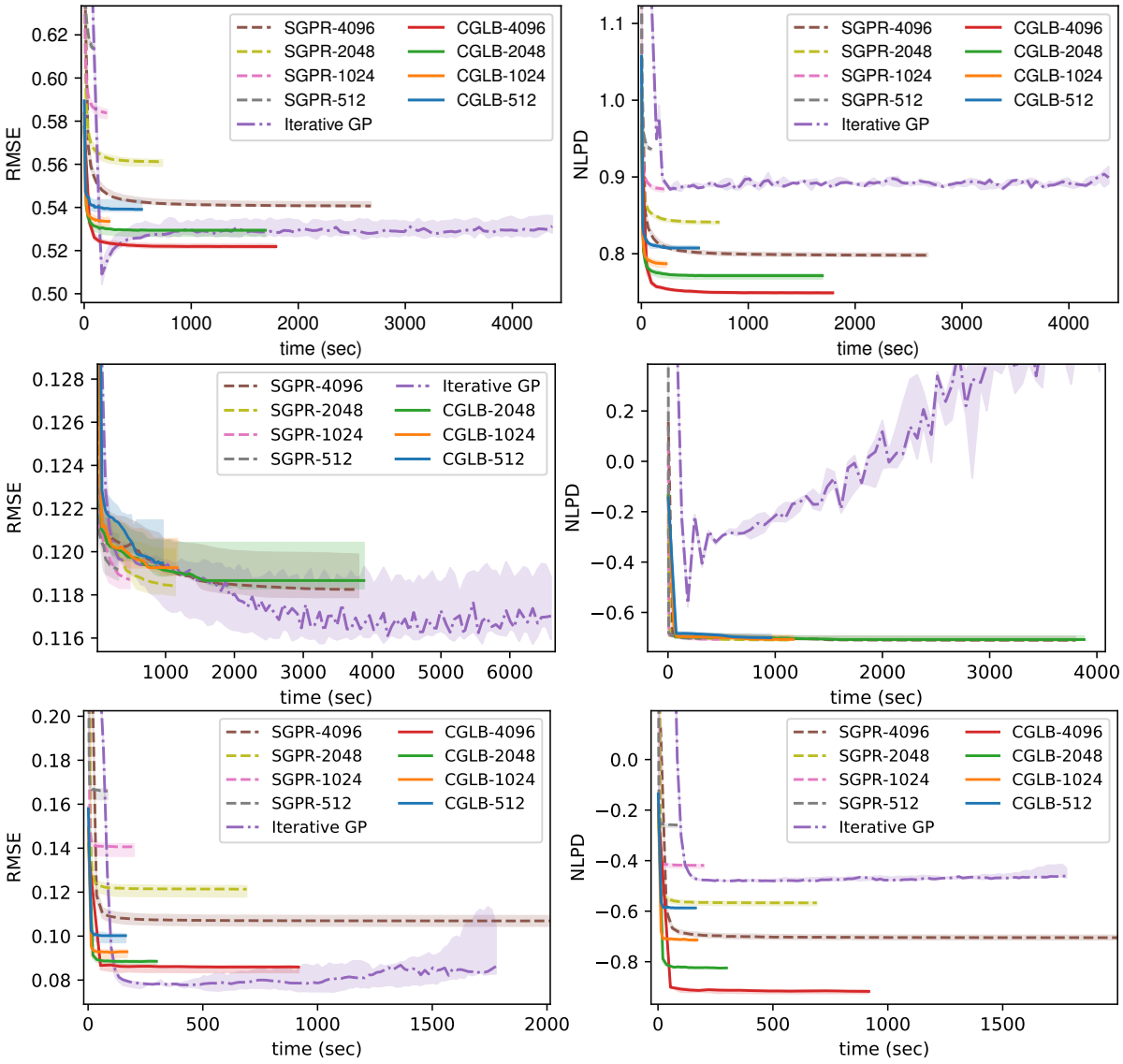


Fig. 4.4 A comparison of evidence lower bound maximization (SGPR, dashed lines), conjugate gradient lower bound maximization (CGLB, solid lines) and the iterative approach taken in Gardner et al. (2018); Wang et al. (2019) (Iterative GP, dot-dashed line) on the `protein`, `keggundirected` and `kin40k` datasets (from top to bottom) in terms of root-mean-square error (RMSE, left) and negative log predictive density (NLPD, right). The trailing number in the legend for evidence lower bound maximization and conjugate gradient lower bound maximization denotes the number of inducing points used. Time only takes into account the time for optimization, though prediction time for all methods is small compared to the time to select hyperparameters. Conjugate gradient lower bound maximization was not run on `keggundirected` with 4096 inducing points due to memory constraints. Conjugate gradient lower bound maximization generally performs comparable or better in terms of these metrics using a similar amount of compute when compared to evidence lower bound maximization and the iterative approach considered as a baseline.

where we checked this metric (table 4.1). In particular, on both `bike` and `poletele` the log marginal likelihood (LML) of the hyperparameters found using conjugate gradient lower bound maximization is noticeably better than the hyperparameters found using evidence lower bound maximization, while on `elevators` there is very little difference between the methods. This provides the most direct evidence that conjugate gradient lower bound maximization better approximates maximum marginal likelihood model selection.

In summary, on many medium-sized datasets (e.g. $10,000 - 40,000$ training examples) conjugate gradient lower bound maximization offers better performance than evidence lower bound maximization on a comparable computational budget. It is likely this is because the tighter lower bound results in an optimization procedure that more closely resembles maximum marginal likelihood. In applications to larger datasets, memory issues emerge. These are in some sense an artifact of implementation since all the highly memory-inefficient operations in conjugate gradient lower bound maximization can be computed in memory efficient ways. However, it is a non-trivial engineering task to realize these memory savings without significantly slowing down the method using current python libraries.

Comparison of Conjugate Gradient Lower Bound Maximization and Methods Based on Stochastic Trace Estimation It is ambiguous, and depends on the dataset, whether conjugate gradient lower bound maximization or the iterative Gaussian process method using stochastic trace estimation performs better in terms of root-mean-square error (see figure 4.4, table 4.1 for details). On all the smaller datasets where we computed the log marginal likelihood of parameters directly conjugate gradient lower bound maximization found parameters with a higher log marginal likelihood, suggesting the model selection procedure more closely resembles maximum marginal likelihood. Additionally, conjugate gradient lower bound maximization generally achieves better negative log probability density. In certain instances, the approach using stochastic Lanczos quadrature led to very small values for σ^2, perhaps due to the biases in the objective function discussed in the previous section.

Under-estimation of the likelihood variance may be due to not running enough iterations of the method of conjugate gradient: the gradient used in Gardner et al. (2018) and Wang et al. (2019) for the quadratic term is the gradient of the lower bound on the quadratic term from lemma 4.5, which differs from the exact quantity by $\mathbf{r}^\top \mathbf{K}^{-1}\mathbf{r}$. It does not penalize very small values of the likelihood variance as much as the exact quadratic term, and the stopping criterion used by the method in Wang et al. (2019) based on the residual norm does not ensure the bias introduced is small. One of the primary advantages of the stopping criterion we consider is that it avoids this pathology.

We additionally see instability in training for the training procedure described in Wang et al. (2019) (figure 4.4). We expect this is due to the same issue of bias in the approximation to the objective function. Since the bias is not a lower bound, the optimizer can select settings of hyperparameters that lead to large amounts of positive bias, which cannot occur when optimizing a lower bound (see also the discussion of upper bound maximization in Kim and Teh, 2018, Appendix L). Similar observations regarding optimization instability of this procedure were made concurrently to our work in Potapczynski et al. (2021).

We investigated whether smaller learning rates or enforcing higher constraints on the likelihood variance improved stability of model selection on datasets in figure 4.5. Changing optimization parameters can lead to significant improvements in stability, usually at the cost of slower optimization. In contrast, there was no need to tune optimization parameters for conjugate gradient lower bound maximization, as line search can be used to determine the step length in each iteration.

Conjugate gradient lower bound maximization improves the reliability of the model selection process relative to previous iterative approaches. Additionally, conjugate gradient lower bound maximization removes the need to tune parameters controlling the optimization process, as well as the number of vectors sampled to estimate the log determinant and its gradient. The only challenging parameter to select in conjugate gradient lower bound maximization is the number of inducing points to use M. The lower bound eq. (4.57) is monotonic in M, and the challenges with selecting it are similar as to with evidence lower bound maximization. In practice, selecting M to be as large as possible within a computational budget, or by assessing an upper bound on the log marginal likelihood (lemma 4.15) are both reasonable approaches. Overall, conjugate gradient lower bound maximization improves the reliability of model selection and ease-of-use compared to existing iterative approaches while often more closely resembling maximum marginal likelihood model selection.

4.5.3 Method Summary and Future Directions

In this section as well as sections 4.3 and 4.4 we discussed *conjugate gradient lower bound maximization*, the method proposed in Artemev et al. (2021) for hyperparameter selection. This approach combines aspects of the typical workflow in variational Gaussian process regression with the method of conjugate gradients. The central idea is to consider a lower bound on the log marginal likelihood that is generally tighter than the evidence lower bound and to use this as a surrogate to the log marginal likelihood for model selection. This comes at the cost of performing matrix-vector multiplication with the full kernel matrix, which is $O(N^2)$, but can be implemented in a matrix-free fashion (that is, without storing the full matrix in memory), and therefore does not change the scaling of the memory overhead relative to variational Gaussian process regression. In general, the resulting procedure still introduces bias into hyperparameter selection, in the sense that the maximizer of the bound in eq. (4.57) does not coincide with the maximum marginal likelihood setting of hyperparameters; however, we find empirically conjugate gradient lower bound maximization leads to better model selection than evidence lower bound maximization for many datasets and models.

As compared to existing iterative approaches that utilize stochastic trace estimation, conjugate gradient lower bound maximization leverages a deterministic objective function to facilitate faster convergence of hyperparameter optimization. Additionally, it is better able to reuse previous computation as it avoids the need to solve entirely new systems of equations that occurs due to re-sampling vectors for stochastic trace estimation. Finally, maximizing a lower bound on the log marginal likelihood and relating the stopping criterion to the accuracy of the approximation to the log marginal likelihood leads

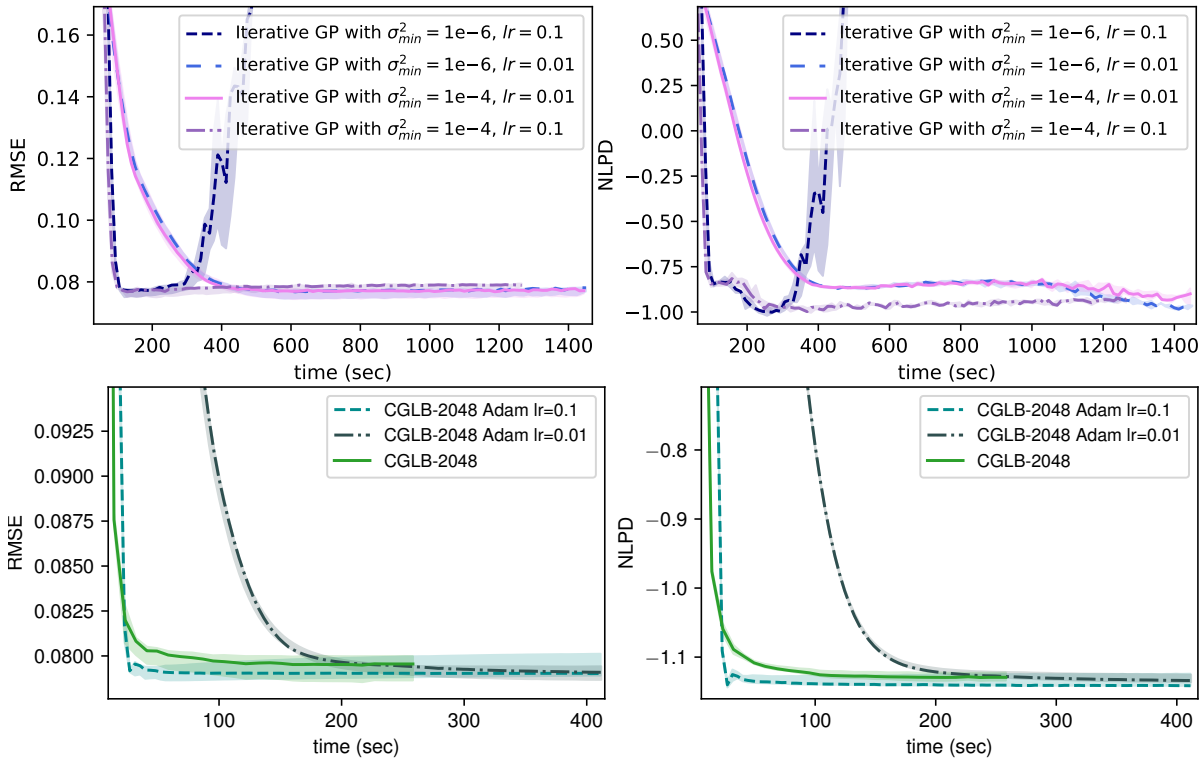


Fig. 4.5 Test performance over time of the method used in Gardner et al. (2018) (Iterative GP) (top) and of conjugate gradient lower bound maximization (bottom) on the `poletele` dataset. If a reasonably high lower threshold is not set on the likelihood noise with Iterative GP, and a learning rate of 0.1 was used, the method diverges (blue dashed curve). Lowering the learning rate resolves this issue, at the cost of slower convergence (pink and blue curves). Alternatively, setting a higher minimum noise level improves stability of the method (purple curve), though this risks modeling some signal in the data as likelihood noise. In contrast, conjugate gradient lower bound maximization was robust to choice of learning rate when trained with Adam, and can be trained with L-BFGS (green curve) which utilizes line search. The behavior of model selection with both approaches depends on the specific dataset, but we never observed instability with conjugate gradient lower bound maximization.

Table 4.1 Median log marginal likelihood, predictive negative log predictive density and predictive root-mean-square error over three datasets splits for the iterative approach taken in Gardner et al. (2018); Wang et al. (2019) (Iterative GP), evidence lower bound maximization (SGPR) and conjugate gradient lower bound maximization (CGLB). Cholesky subcolumns represent the same metrics evaluated by using a Cholesky-based Gaussian process regression implementation (section 1.3) with hyperparameters found by each method. On the `poletele` dataset, Iterative GP overestimates the log marginal likelihood by a large amount, which cannot occur with the other two methods. Conjugate gradient lower bound maximization with $M = 4096$ is missing for `keggundirected` due to the high memory requirement.

		LML		NLPD		RMSE	
		Approx	Cholesky	Approx	Cholesky	Approx	Cholesky
`bike` N=11643, D=17	Iterative GP	30992.8	31319.1	-2.016	-3.257	0.020	0.014
	SGPR-4096	30502.5	32814.2	-3.280	-3.336	0.010	0.010
	CGLB-4096	37732.7	**42023.0**	**-4.216**	-4.329	0.004	0.004
	CGLB-2048	34102.8	38936.7	-3.811	-3.972	**0.003**	0.003
	CGLB-1024	30493.9	35351.8	-3.403	-3.615	0.005	0.005
`elevators` N=11121, D=18	Iterative GP	-4709.0	-4705.1	0.407	0.384.	**0.353**	0.353
	SGPR-4096	-4675.3	**-4653.3**	**0.386**	0.386	0.354	0.354
	CGLB-4096	-4669.8	-4659.1	**0.386**	0.386	0.354	0.354
	CGLB-2048	-4677.9	-4656.4	0.387	0.387	0.355	0.355
	CGLB-1024	-4712.0	-4670.0	0.392	0.391	0.356	0.356
`poletele` N=10050, D=26	Iterative GP	13552.5	-7641.5	-0.935	1.217	0.079	0.078
	SGPR-4096	9057.7	9624.0	-1.172	-1.180	0.078	0.078
	CGLB-4096	9377.1	**9862.2**	**-1.201**	-1.203	**0.077**	0.077
	CGLB-2048	8248.6	9248.0	-1.126	-1.145	0.080	0.080
	CGLB-1024	7250.7	8694.2	-1.057	-1.098	0.083	0.083
`kin40k` N=26800, D=8	Iterative GP	23859.0	—	-0.454	—	0.087	—
	SGPR-4096	7486.0	—	-0.705	—	0.107	—
	CGLB-4096	12244.2	—	**-0.919**	—	**0.086**	—
	CGLB-2048	10028.6	—	-0.826	—	0.088	—
	CGLB-1024	7260.0	—	-0.714	—	0.093	—
`protein` N=29267, D=9	Iterative GP	-25703.9	—	0.897	—	0.531	—
	SGPR-4096	-27714.6	—	0.798	—	0.541	—
	CGLB-4096	-26570.7	—	**0.749**	—	**0.522**	—
	CGLB-2048	-27442.0	—	0.771	—	0.529	—
	CGLB-1024	-28243.7	—	0.790	—	0.535	—
`keggundirected` N=42617, D=27	Iterative GP	-21659.7	—	1.310	—	**0.117**	—
	SGPR-4096	-29837.5	—	**-0.710**	—	0.118	—
	CGLB-4096	—	—	—	—	—	—
	CGLB-2048	-29751.7	—	-0.708	—	0.119	—
	CGLB-1024	-29728.9	—	-0.707	—	0.120	—

to more stable model selection than previous approaches. These modifications make the method more reliable and easier to use.

In its current form conjugate gradient lower bound maximization has benefits compared to other approaches for scalable model selection, particularly in instances where maximum marginal likelihood is believed to be a suitable approach. However, the derivation is largely ad hoc. Ideally, the objective function and predictive distribution would be more closely linked, as the evidence lower bound and variational posterior are in sparse inference. To this end, we pose the following open problem:

Open Problem 4.16. *Can the objective functions eq.* (4.56) *and eq.* (4.57) *be formulated as evidence lower bounds within a variational Bayesian framework? That is, does there exist a (computationally favorable) approximate posterior such that the difference between each of these bounds and the log marginal likelihood is a Kullback-Leibler divergence?*

A sensible starting point for investigating this question is the 'decoupled' variational family considered in Cheng and Boots (2017); Salimbeni et al. (2018), which considers a variational family containing processes with a broader class of mean functions than considered in the method of Titsias (2009).

4.6 Adaptive and Barely Biased Gaussian Process Regression

We now investigate whether we can improve the reliability of training of iterative methods applied to Gaussian process regression using stochastic Lanczos quadrature. We saw in the previous section that hyperparameter optimization with existing implementations of this method can introduce biases in to the optimization procedure and exhibit instability during model selection with gradient-based hyperparameter optimization. In order to resolve this issue, we previously relied on a method based on Nyström approximation to approximate the log determinant (sections 4.3 to 4.5). However, it would be advantageous to have a method to estimate the log determinant that relies on iterative approaches to estimate this term, while providing reliable estimates that can be used for model selection. Directly using iterative methods avoids the need to select a number of inducing points, M, and can inherit convergence properties from Lanczos quadrature, which are preferable on problems with spread-out data to convergence properties from methods based on Nyström approximation (example 3.1). In this section, we make progress towards this goal by extending the idea of directly linking stopping criterion to the quality of the estimate of the log marginal likelihood to the log determinant term using Lanczos quadrature.

We describe a minor modification of the method in Ubaru et al. (2017), that ensures estimates of the log marginal likelihood have bias less than ε. Instead of using Gauss quadrature alone to estimate the log determinant, we keep track of the Gauss-Radau quadrature estimate as well, prescribing a single node that is not larger than the smallest eigenvalue of $\mathbf{K}$. This allows us to assess an upper bound on the bias each time an estimate of the log marginal likelihood is evaluated, and terminate the iterative method when this bias is small. As such, the method is automatically adaptive: if the iterative method converges quickly then very few iterations will be run, while if the method converges slowly, more compute will be used to account for this. The calculations involved are outlined in section 4.6.1 and summarized in algorithm 5.

A similar approach to quantifying error was analyzed and applied in the kernel-based machine learning context in Li et al. (2016b). They focus on the problem of matrix inversion, though the idea readily generalizes to log determinant estimation up to the use of a stochastic estimator. A primary difference in our application as compared to Li et al. (2016b) is the need to estimate gradients of the quantity estimate with stochastic Lanczos quadrature.

4.6.1 Description of Estimator and Gradients

We use stochastic Lanczos quadrature (section 4.1) with a Gaussian stochastic trace estimator for estimating $\log\det \mathbf{K}$ as in Ubaru et al. (2017). In order to perform gradient based optimization, we additionally need to estimate $\frac{\partial}{\partial\theta}\log\det\mathbf{K}$. The entire procedure used is outlined in algorithm 5.

Estimator of log determinant Golub and Welsch (1969) showed that the Gauss quadrature estimate of $\mathbf{w}_\ell^\top \log\mathbf{K}\mathbf{w}_\ell$ using t nodes is given by,

$$d^{G,t}(\mathbf{w}_\ell) = \|\mathbf{w}_\ell\|^2 \mathbf{e}_1^\top \log\mathbf{T}^{(t,\ell)}\mathbf{e}_1, \tag{4.67}$$

where $\mathbf{e}_1 = [1,0,\ldots,0]^\top \in \mathbb{R}^t$, and $\mathbf{T}^{(t,\ell)} \in \mathbb{R}^{t\times t}$ is the tridiagonal matrix produced by Lanczos quadrature with initial vector $\mathbf{w}_\ell/\|\mathbf{w}_\ell\|$. The estimate for the log marginal likelihood we use is

$$\tilde{\mathcal{L}}^t(\theta,\mathbf{W}) = C - \frac{1}{2L}\sum_{\ell=1}^{L} d^{G,t}(\mathbf{w}_\ell) - \frac{1}{2}\left(\frac{\|\mathbf{r}_t\|_2^2}{\sigma^2} + 2\mathbf{y}^\top\mathbf{v}_t - \mathbf{v}_t^\top\mathbf{K}_\theta\mathbf{v}_t\right). \tag{4.68}$$

By theorem 4.2 and since even derivatives of logarithm are negative,

$$\mathbf{w}_\ell^\top \log\mathbf{K}\mathbf{w}_\ell \leq d_\ell^{G,t}(\mathbf{w}_\ell). \tag{4.69}$$

Taking expectations on both sides of eq. (4.68) and using lemma 4.3,

$$\mathbb{E}[\tilde{\mathcal{L}}^t(\theta,\mathbf{W})] \leq \mathcal{L}(\theta). \tag{4.70}$$

Hence, the expected value of the estimate used for the log marginal likelihood is a lower bound.

On the other hand, since odd derivatives of the logarithm are positive, forming a Gauss-Radau estimator with prescribed node $\sigma^2 \leq \lambda_N(\mathbf{K})$ results in an estimator

$$d^{GR,t}(\mathbf{w}_\ell) \leq \mathbf{w}_\ell^\top \log\mathbf{K}\mathbf{w}_\ell \tag{4.71}$$

by eq. (4.23). Consider,

$$\tilde{\mathcal{U}}^t(\theta,\mathbf{w}_{1:L}) = C - \frac{1}{2L}\sum_{\ell=1}^{L} d^{GR,t}(\mathbf{w}_\ell) - \frac{1}{2}\left(2\mathbf{y}^\top\mathbf{v}_t - \mathbf{v}_t^\top\mathbf{K}_\theta\mathbf{v}_t\right). \tag{4.72}$$

From eq. (4.71) and lemma 4.5,

$$\mathcal{L}(\theta) \leq \mathbb{E}[\tilde{\mathcal{U}}^t(\theta,\mathbf{w}_{1:L})]. \tag{4.73}$$

We run the method of conjugate gradients (algorithm 3) and Lanczos algorithm (algorithm 4) until we find the smallest $t = \tau$ such that

$$\tilde{\mathcal{U}}^{\tau}(\theta, \mathbf{w}_{1:L}) - \tilde{\mathcal{L}}^{\tau}(\theta, \mathbf{w}_{1:L}) \leq \varepsilon, \tag{4.74}$$

where ε is a user-specified parameter. Combining eqs. (4.70), (4.72) and (4.74)

$$\mathbb{E}[\tilde{\mathcal{L}}^{\tau}(\theta, \mathbf{w}_{1:L})] \leq \mathcal{L}(\theta) \leq \mathbb{E}[\tilde{\mathcal{L}}^{\tau}(\theta, \mathbf{w}_{1:L})] + \varepsilon. \tag{4.75}$$

We refer to performing model selection with this procedure as *barely biased log marginal likelihood estimation*, because the bias of the approximation to the log marginal likelihood used can be controlled explicitly by a user-specified parameter. From a practical perspective, this has the appealing property that instead of specifying a stopping criterion for iterative methods in terms of a residual, the user specifies a parameter that directly controls the accuracy of the approximation to the log marginal likelihood. Additionally, the bias in the estimate for the log marginal likelihood is *uniform* over hyperparameters, which will prevent optimization favoring models with large bias but low log marginal likelihood.

In practice, we use a preconditioned variant of both conjugate gradients and Lanczos quadrature with a Nyström approximation as a preconditioner as in conjugate gradient lower bound maximization. We therefore replace the use of the upper bound from lemma 4.3 in eq. (4.68) with the bound from lemma 4.5, and use

$$\log\det \mathbf{K} = \log\det \mathbf{Q} + \log\det(\mathbf{Q}^{-1/2}\mathbf{K}\mathbf{Q}^{-1/2}). \tag{4.76}$$

We can compute $\log\det \mathbf{Q}$ in closed form and estimate $\log\det(\mathbf{Q}^{-1/2}\mathbf{K}\mathbf{Q}^{-1/2})$ via stochastic Lanczos quadrature as described above.

Gradient of Objective Function We now consider computing the gradient $\nabla_{\theta}\underline{\mathcal{L}}^{\tau}(\theta, \mathbf{w}_{1:L})$. The simplest method from an implementation perspective for computing $\nabla_{\theta}\underline{\mathcal{L}}^{\tau}(\theta, \mathbf{w}_{1:L})$ is to differentiate through the Lanczos algorithm and conjugate gradients using automatic differentiation. While feasible, this has computational disadvantages as the computational graph grows larger with each iteration of conjugate gradients and Lanczos algorithm, leading to more computational overhead.

A second approach is to use conjugate gradients to directly estimate the gradient of the log marginal likelihood (eq. 4.29), as in Gibbs and MacKay (1997). While practical, the approximation of the gradient and of the objective function would then be decoupled, and it would be unclear if the proposed stopping criterion (eq. 4.74) has any interpretation in relation to the gradient estimate used.

We take a third approach inspired by viewing the method as a form of block optimization as in section 4.4. The central idea will be to consider $\mathbf{U}^{(\tau,\ell)}$, the matrix with orthogonal columns produced by τ iterations of the Lanczos algorithm with initial vector $\mathbf{w}_{\ell}$ as an auxiliary parameter that affects the tightness of the bound. We therefore rewrite the Lanczos Gauss quadrature estimate fully in terms of $\mathbf{U}^{(\tau,\ell)}$.

Since $(\mathbf{U}^{(\tau,\ell)})^\top \mathbf{U}^{(\tau,\ell)} = \mathbf{I}_t$ and the first column of $\mathbf{U}^{(\tau,\ell)}$ is $\mathbf{w}_\ell/\|\mathbf{w}_\ell\|$, we have

$$\|\mathbf{w}_\ell\|\mathbf{e}_1 = (\mathbf{U}^{(\tau,\ell)})^\top \mathbf{w}_\ell. \tag{4.77}$$

Also, by well-known properties of the Lanczos algorithm (see Golub and Van Loan, 2013, Chapter 9),

$$\mathbf{T}^{(\tau,\ell)} = (\mathbf{U}^{(\tau,\ell)})^\top \mathbf{K}_\theta \mathbf{U}^{(\tau,\ell)}. \tag{4.78}$$

Hence,

$$d^{G,\tau}(\mathbf{w}_\ell) = \mathbf{w}_\ell^\top \mathbf{U}^{(\tau,\ell)} \log((\mathbf{U}^{(\tau,\ell)})^\top \mathbf{K}_\theta \mathbf{U}^{(\tau,\ell)})(\mathbf{U}^{(\tau,\ell)})^\top \mathbf{w}_\ell. \tag{4.79}$$

This can be computed in $O(N^2\tau)$, which is the same computational cost as running Lanczos algorithm. We then estimate,

$$\begin{aligned}\nabla_\theta \underline{\mathcal{L}}^\tau(\theta, \mathbf{w}_{1:L}) \approx \frac{1}{2L}\sum_{\ell=1}^{L} \nabla_\theta \mathbf{w}_\ell^\top \mathbf{U}^{(\tau,\ell)} \log((\mathbf{U}^{(\tau,\ell)})^\top \mathbf{K}_\theta \mathbf{U}^{(\tau,\ell)})(\mathbf{U}^{(\tau,\ell)})^\top \mathbf{w}_\ell \\ - \frac{1}{2}\nabla_\theta \left(\frac{\|\mathbf{y} - \mathbf{K}_\theta \mathbf{v}_\tau\|_2^2}{\sigma^2} + 2\mathbf{y}^\top \mathbf{v}_\tau - \mathbf{v}_\tau^\top \mathbf{K}_\theta \mathbf{v}_\tau \right).\end{aligned} \tag{4.80}$$

Both $\mathbf{U}^{(\tau,\ell)}$ and $\mathbf{v}_\tau$ depend on θ through the iterative procedure used to find them, but these gradients are ignored in eq. (4.79). It is unclear that eq. (4.80) is consistent as a result. Morally, it seems possible this can be rephrased as defining an objective function $\underline{\mathcal{L}}(\theta, \mathbf{w}_{1:L}, \{\mathbf{U}^{(\cdot,\ell)}\}_{\ell=1}^L, \mathbf{v})$, and performing block-coordinate ascent, where first the random variable $\mathbf{W}$ is sampled, then the Lanczos algorithm and conjugate gradients are used to update $\{\{\mathbf{U}^{(\cdot,\ell)}\}_{\ell=1}^L, \mathbf{v}\}$ for fixed $\mathbf{W}$ and θ and finally one step of gradient ascent is performed to maximize with respect to θ with all other arguments fixed. However, as the matrices $\{\mathbf{U}^{(\cdot,\ell)}\}_{\ell=1}^L$ change shape during the auxiliary update and are also subject to orthogonality constraints, this interpretation seems rather delicate, and we do not attempt to formalize it here. Burt et al. (2021) gave a class of matrices such that eq. (4.79) is a lower bound for all matrices in this class; however the argument is not sufficiently convincing to justify why the estimator eq. (4.80) is consistent. We instead rely on the empirical finding that the gradients are consistent and reasonably accurate for $t \ll N$ so long as the convergence criterion for Lanczos quadrature eq. (4.74) is satisfied for a reasonably small ε (e.g. $\varepsilon = 1$).

When we use preconditioning, we need to perform matrix-multiplication with $\mathbf{Q}^{-1/2}$ with τ vectors in $\mathbb{R}^N$ in order to recover $\mathbf{U}^{(\tau,\ell)}$, which is not required for computing the approximation to the log marginal likelihood, but only for the gradient estimate in eq. (4.80). This can be done with total cost $O(NM^2 + NM\tau)$ with careful linear-algebraic manipulation, making the overall cost of computing the estimate of the log marginal likelihood and its gradient $O(NM^2 + LN^2\tau + LNM\tau)$.

Algorithm 5 Barely Biased Log Marginal Likelihood Estimation

Input: $\mathbf{K}_\theta \in S^N_{++}$, $\varepsilon > 0$, $L \in \mathbb{N}$.
Returns: A stochastic estimate, $\tilde{\mathcal{L}}^\tau(\theta, \mathbf{w}_{1:L})$ of the log marginal likelihood, $\mathbb{E}[\tilde{\mathcal{L}}^\tau(\theta, \mathbf{w}_{1:L})] \leq \mathcal{L}(\theta) \leq \mathbb{E}[\tilde{\mathcal{L}}^\tau(\theta, \mathbf{w}_{1:L})] + \varepsilon$.
Sample $\{\mathbf{w}_\ell\}_{\ell=1}^L$ s.t. $\mathbb{E}[\mathbf{w}_\ell \mathbf{w}_\ell^\top] = \mathbf{I}$.
$t = 0$
while $\Delta \geq \varepsilon$ **do**
  $t = t + 1$
  Run an iteration the Lanczos algorithm (algorithm 4) to update the matrices $\mathbf{T}^{(t,\ell)}, \mathbf{U}^{(t,\ell)}$ and an iteration of conjugate gradients (algorithm 3) to update $\mathbf{v}_t$.
  Compute $\tilde{\mathcal{L}}^t(\theta, \mathbf{w}_{1:L})$ (eq. 4.68)
  Compute $\tilde{\mathcal{U}}^t(\theta, \mathbf{w}_{1:L})$ (eq. 4.72). See Golub (1973) for an implementation of the Gauss-Radau estimate.
  $\Delta = \tilde{\mathcal{U}}^t(\theta, \mathbf{w}_{1:L}) - \tilde{\mathcal{L}}^t(\theta, \mathbf{w}_{1:L})$.
end while
$\tau = t$
Compute the approximation in eq. (4.80), $G^\tau(\theta, \mathbf{w}_{1:L}) \approx \nabla_\theta \tilde{\mathcal{L}}^\tau(\theta, \mathbf{w}_{1:L})$.
Return: $\tilde{\mathcal{L}}^\tau(\theta, \mathbf{w}_{1:L}), G^\tau(\theta, \mathbf{w}_{1:L})$.

4.6.2 Experimental Results

We present a small collection of empirical results for hyperparameter selection to investigate properties of model selection with the method described in algorithm 5. As the estimator for the log marginal likelihood and its gradient are stochastic, we perform optimization with Adam (Kingma and Ba, 2015) using initial learning rate 0.1. Barely biased log marginal likelihood estimation has two parameters to tune: the bias tolerance $\varepsilon > 0$ and the number of probe vectors L in eq. (4.79). We investigated the effect of changing the bias tolerance and the number of probe vectors in figure 4.6. We found that using a single probe vector does not increase the variance dramatically, and yields similar performance to $L = 10$. With bias tolerances of $\varepsilon = \{10, 100\}$ we observed that the number of steps required to reach the desired level of bias decreases as does the stability of the log marginal likelihood estimate during training. The range of ε considered is reasonably small: 100 nats often corresponds to a small change in hyperparameters particularly if the noise variance is small.

The data preparation and training procedure for the baseline methods is identical to the one followed in section 4.5. Barely biased log marginal likelihood estimation (BBGP) shows promising results in terms of stability of training and performance as measured by root-mean-square error (RMSE) on both datasets considered in figure 4.7. Barely biased log marginal likelihood estimation is run with $\varepsilon = 100$ and preconditioner rank 500. The plot on the left shows the training objective plotted against the number of iterations. For all the methods except the iterative approach used by Gardner et al. (2018); Wang et al. (2019) the estimate is an upper bound on the negative log marginal likelihood. We see that the bound provided by barely biased log marginal likelihood estimation becomes significantly lower than for evidence lower bound maximization or conjugate gradient lower bound maximization, providing some

evidence that the parameters selected are favored by the log marginal likelihood (although this could also be due to using a tighter bound). Comparing to table 4.1, we see that the upper bound found by barely biased log marginal likelihood estimation is lower than the actual negative log marginal likelihood of the hyperparameters found by conjugate gradient lower bound maximization on these datasets, providing more definitive evidence it finds hyperparameters favored by the log marginal likelihood. We also see that the root-mean-square error on held-out data over time of barely biased log marginal likelihood estimation decreases and does not appear to run into any of the reliability issues exhibited by existing iterative approaches.

While small-scale empirical results are somewhat promising in terms of reliability of model selection with barely biased log marginal likelihood estimation, the method is not currently practical. First, in terms of wall-clock time, the method requires several times more computation than competing methods, while not achieving significant performance gains. Additionally, we found it was necessary to perform re-orthogonalization in algorithm 4; otherwise numerical errors led to poor performance in algorithm 5. This, as well as the gradient estimator eq. (4.80) mean that we need to store the matrices $\{\mathbf{U}^{(\tau,\ell)}\}_{\ell=1}^{L}$ produced during Lanczos algorithm. This greatly increases the memory overhead as compared to other implementations of Gaussian process regression built on similar machinery. Due to this memory limitation, in practice, it may be necessary to set maximum number of iterations to run. This means the stopping criterion eq. (4.74) may not be satisfied, and the estimator could be quite biased. In its current form barely biased log marginal likelihood estimation is not a feasible replacement for existing iterative approaches. However, it provides an illustration that it is possible to directly relate stopping criterion for the objective function to the quality of the estimate of the log marginal likelihood, as well as preliminary evidence that this improves reliability of model selection. We believe these ideas could be useful in more practical methods in the future.

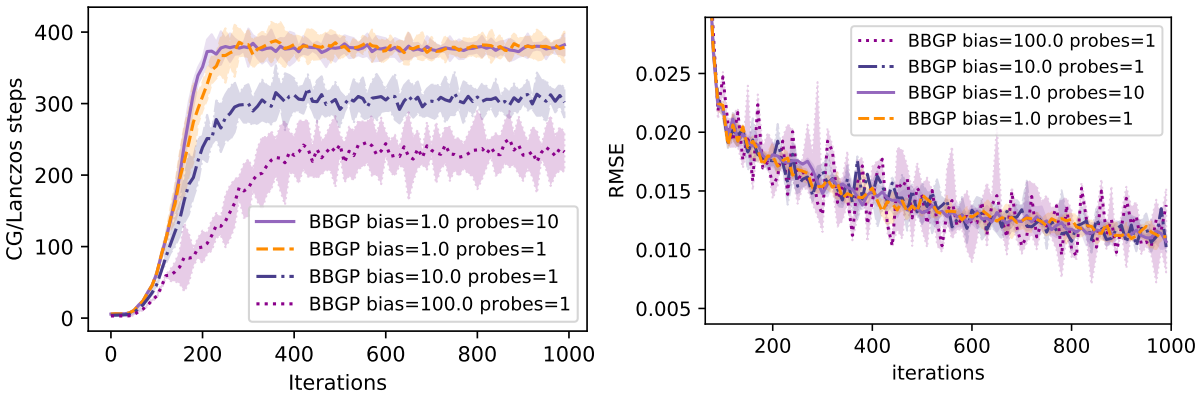


Fig. 4.6 The plot on the right shows the number of steps spent per optimization iteration for $\varepsilon = \{1, 10, 100\}$ (mean and standard deviation over 5 splits). Larger values of ε correspond to more biased estimates of the log marginal likelihood at lower computational cost. The right plot compares root-mean-square error on a held-out set for several choices of ε and number of probe vectors L. There seems to be negligible difference in training stability from changing L from 1 to 10, while decreasing ε slightly improve stability of training.

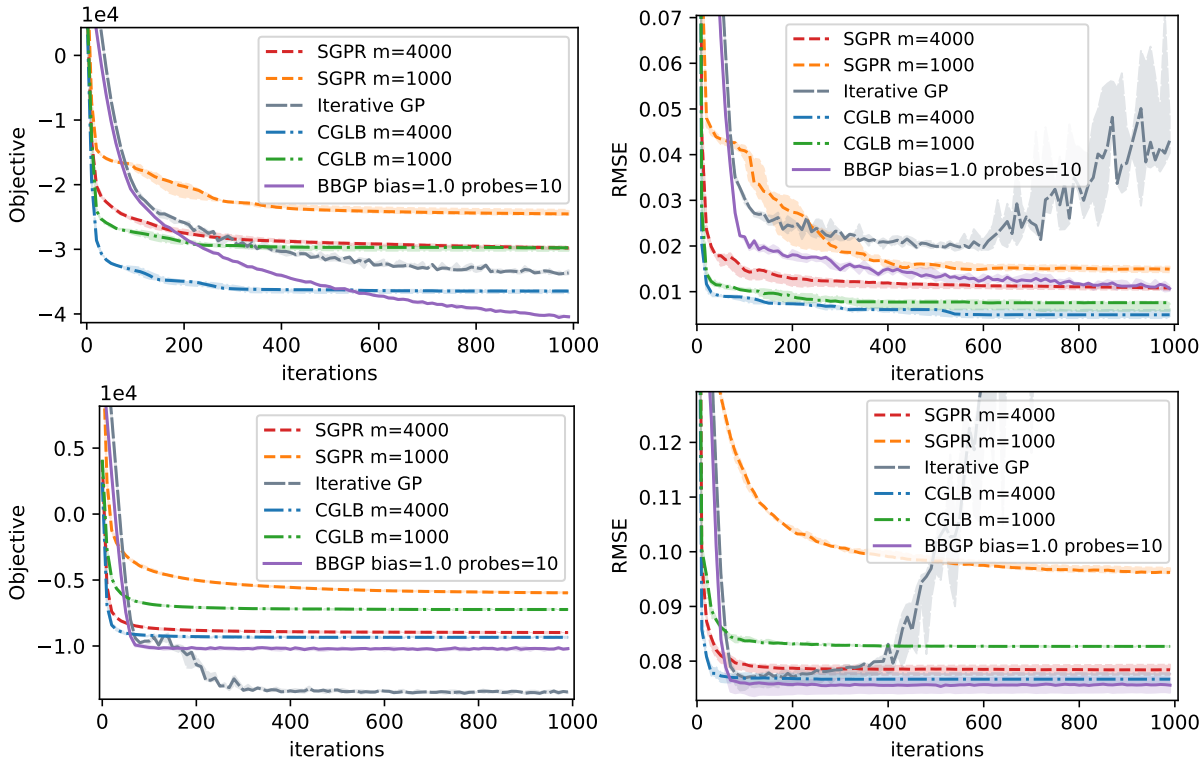


Fig. 4.7 Model performance on testing data and objective (an estimate of negative log marginal likelihood) traces plotted against steps of optimization of θ for the `bike` dataset (top) and the `poletele` dataset (bottom). Barely biased Gaussian process regression generally is stable to train and achieves reasonable performance on a per iteration basis.

4.7 Summary and Future Directions

In this chapter we discussed the use of iterative methods, especially the method of conjugate gradients and the Lanczos algorithm, for approximate inference and maximum marginal likelihood in Gaussian process regression. We focused on methods for improving the reliability and ease-of-use of model selection with these methods, particularly through linking the number of iterations of the method run to the quality of estimation of the log marginal likelihood. Conjugate gradient lower bound maximization (sections 4.3 to 4.5) is a practical method for model selection combining low-rank approximation and conjugate gradients. It combines several favorable aspects of evidence lower bound maximization (deterministic objective function, control on direction of bias of estimate for log marginal likelihood) that tend to lead to stable optimization with advantages of iterative methods, particularly reduced bias relative to maximum marginal likelihood model selection. Barely biased log marginal likelihood estimation (section 4.6) is conceptually a minor modification of existing iterative approaches to scalable model selection in Gaussian process regression designed to improve the reliability of these methods by ensuring the bias when estimating the log marginal likelihood is small. In practice, it is not currently a practical method due to increased memory and computational requirements.

4.7.1 Concurrent and Subsequent Work

We have highlighted relevant prior work in the background sections as well as when introducing methods. We now briefly discuss work that was concurrent or subsequent to Artemev et al. (2021) and Burt et al. (2021).

Concurrent to Artemev et al. (2021) and prior to Burt et al. (2021) Potapczynski et al. (2021) observed that the approach taken in Gardner et al. (2018); Wang et al. (2019), involving stochastic trace estimation and conjugate gradients for estimating the gradient of the log marginal likelihood and stochastic Lanczos quadrature together with conjugate gradient for estimating the log marginal likelihood, can be unreliable for model selection. They proposed removing the bias of this estimator via randomizing the number of iterations run and weighting terms appropriately to form a "Russian roulette" estimator (Kahn, 1955, pages 4-5). This approach trades-off bias for variance. In contrast, conjugate gradient lower bound maximization emphasizes controlling the direction of the bias as a method for increasing training stability, and barely biased log marginal likelihood estimation trades-off bias and computation. In general, with iterative approaches there appears to be a bias-variance-computation trade-off, and exploring the effect of this trade-off on model selection is an active research area.

Several other works have considered the application of L-BFGS directly with the approach of Gardner et al. (2018), building on our observation that it greatly improved convergence with conjugate gradient lower bound maximization. Wenger et al. (2021) observed that the preconditioner can act as a control variate for the objective function and its gradient, so that good preconditioning can reduce the variance of stochastic trace estimation. If the condition number of the preconditioned matrix is 1 and a Rademacher vector is used, the trace estimator is deterministic. As mentioned in section 4.4, Wenger et al. (2021) applied L-BFGS with line search, despite the remaining stochasticity in the objective and its gradients and reported good performance. Maddox et al. (2021) suggested that a low tolerance for the convergence of conjugate gradient suffices for the application of L-BFGS with the method of Gardner et al. (2018), and advocated changing the default criterion for this parameter accordingly. While taking a stricter definition of convergence may suffice to resolve many of the issues of instability we observed, we still advocate thinking critically about how a selected criterion relates to quantities of interest, particularly the log marginal likelihood. Using a stricter convergence criterion may help in many settings, but extreme parameter values (e.g. $\sigma^2 \approx 0$) will still likely lead to instability in training unless a stopping criterion is chosen that prevents any bias introduced from leading to significant over-estimation of the log marginal likelihood.

4.7.2 Future Directions

We now speculate about useful directions for future research on the application of iterative methods to Gaussian process regression, proposing several research questions.

Open Problem 4.17. *Can a method be designed that effectively leverages information from previous estimates of the log marginal likelihood when evaluating the log determinant? A potential formalism of*

this is: Given any θ and $\tilde{\mathcal{L}}(\theta)$ satisfying $|\tilde{\mathcal{L}}(\theta) - \mathcal{L}(\theta)| \leq \varepsilon$ and some intermediate cached calculations, find a method that will evaluate $\tilde{\mathcal{L}}(\theta+\delta)$ (and its gradients) satisfying $|\tilde{\mathcal{L}}(\theta+\delta) - \mathcal{L}(\theta+\delta)| \leq \varepsilon$ in $O(N^2 + h_\theta(\varepsilon, \|\delta\|)N^3)$ where $\lim_{d\to 0} h_\theta(\varepsilon, d) = 0$ for any $\varepsilon > 0$ and any θ.

The estimator we use for the quadratic term in conjugate gradient lower bound maximization and barely biased log marginal likelihood estimation satisfies this property, so long as the kernel is continuous in the hyperparameters. A method of this type would have significant practical ramifications for approximate maximum marginal likelihood using iterative methods as it would imply that when the hyperparameters do not change significantly, evaluation of the log marginal likelihood is fast.

A second natural question is whether stochastic estimation of the log determinant is necessary and if so why.

Open Problem 4.18. *Design a deterministic estimator for* $\log\det \mathbf{K}$ *with similar convergence properties to stochastic Lanczos quadrature.*

To answer open Problem 4.18 it is tempting to use the Ritz value characterization of eigenvalues of $\mathbf{T}_t$, the tridiagonal matrix formed by the Lanczos algorithm (c.f. eq. 4.16). This allows us to construct upper and lower bounds on the log determinant of $\mathbf{K}$ through upper and lower bounds on eigenvalues. Unfortunately, we do not see a practical way to design bounds in this way that have similar convergence properties to Lanczos quadrature. Additionally, we suspect numerical issues may make such a method impractical.

Finally, a more conceptual goal is to unify variational approaches, discussed in chapters 2 and 3, with iterative methods.

Open Problem 4.19. *Give a variational interpretation of iterative methods that does not suffer from the limitations of Nyström approximation in cases the prior covariance matrix is well-conditioned.*

One can somewhat naturally use Krylov subspace methods to define inducing points via the map

$$\psi : \mathbb{R}^N \to \mathcal{H}, \psi(\mathbf{v}) = \sum_{n=1}^{N} v_n f(x_n) \tag{4.81}$$

then define

$$\mathcal{H}_{\mathbf{z}} = \psi(\mathcal{K}_{M-1}(\mathbf{K}, \mathbf{v})) \tag{4.82}$$

and perform variational inference using this subspace as described in chapter 2. Bartels and Hennig (2020) proposed combining the features implicit in eq. (4.81) and eq. (4.82) with conventional inducing point methods, taking the viewpoint of probabilistic numerics as opposed to variational inference.

A disadvantage of performing variational Gaussian process regression with the subspace outlined above is that the log determinant estimate given this approximation does not inherit the convergence properties of stochastic Lanczos quadrature on datasets like example 3.1, where the condition number is near 1 and the matrix is not close to low-rank. A low-rank is still relied upon, and it faces the

same fundamental limitations as other inducing point methods discussed in section 3.3. A conceptual framework that endows iterative approaches with a variational interpretation would be useful for comparing the pros and cons of these approaches to Nyström methods, as both approaches could then be discussed and compared in the same language. Such a framework would offer probabilistic insights into the assumptions about the dataset and model are needed for iterative methods to work, as well as be the basis for developing reliable training procedures for Gaussian process regression built on iterative approximations.

Chapter 5

Discussion

Approximate Gaussian process regression is one of the best understood instances of approximate Bayesian inference: simple diagnostic tools and theory can be used to give confidence an approximation will succeed or has succeeded. Before taking a step back to summarize the contributions of this thesis in this area and to briefly reflect on this problem, we recall the questions laid out in chapter 1 for understanding an approximate Bayesian method: *Will it work?*, *Did it work?* and *Is it easy to use?* We will structure the remainder of our discussion around these questions as applied to approximate inference and model selection in Gaussian process regression.

5.1 Contributions of this Thesis

Will it work? A primary contribution of this thesis is an a priori analysis of variational Gaussian process regression, contained in chapter 3. Building off of Burt (2018), we formalized the intuition that accurate variational approximations to the posterior that require little computation can be found within the sparse framework if the model and dataset are sufficiently simple, for example if the prior is smooth and the data is concentrated. We discussed practical algorithms with a priori guarantees that, on average or with fixed probability over datasets and initializations of inducing points, result in an approximation to the posterior that has a small Kullback-Leibler divergence to the posterior.

While the interpretation of these bounds is the clearest if hyperparameters are fixed, if global optimization of the evidence lower bound is performed, then a small Kullback-Leibler divergence to the posterior for the maximum marginal likelihood hyperparameters ensures that the parameters selected do not have a much smaller log marginal likelihood than the maximum marginal likelihood parameters (eq. 2.99). Therefore, our a priori analysis also sheds a small amount of insight onto the problem of model selection with evidence lower bound maximization. However, this question is still largely unresolved from a practical point of view.

We additionally considered lower bounds that show that the number of inducing points cannot scale too slowly with dataset size, at least if one wants the Kullback-Leibler divergence between the approximate posterior and prior to remain small (section 3.3).

The analysis in chapter 3 helps to clarify the contexts in which sparse Gaussian process regression can be reasonably expected to work, and properties of the model and data that indicate a different scalable approximation is needed.

In chapter 4 we provided some empirical support for the claim that the proposed method for model selection, *conjugate gradient lower bound maximization* (Artemev et al., 2021), will work on a range of tasks. As the method borrows from sparse variational approaches and results in a generally more accurate approximation to the log marginal likelihood, one can extrapolate that it will work at least in instances in which sparse variational inference will work. On the other hand, empirically we saw that there were instances in which conjugate gradient lower bound maximization out-performs evidence lower bound maximization in terms of selecting hyperparameters favored by the log marginal likelihood. While the empirical analysis provided is not exhaustive and it would be beneficial to have theory to guide the set of assumptions needed for conjugate gradient lower bound maximization to succeed as a form of model selection, the experiments in section 4.5 provide some preliminary evidence that the method works in a range of settings.

Did it work? In chapter 2 we discussed diagnostic tools for inference in Gaussian process regression that, to the best of our knowledge, have not been previously used, to check the extent to which approximation has impacted statistical conclusions drawn from the model (especially, proposition 2.8, proposition 2.15, proposition 2.13). The few examples of diagnostics we gave are far from exhaustive, with other approaches outlined in Davies (2015) and, in a much more general setting, in Huggins et al. (2020). We believe the development of improved diagnostics for approximate Gaussian process regression, as well as automating these diagnostics in software packages, is critical to guiding practitioners to use approximate inference correctly and trust that the results obtained through this procedure.

In chapter 4 we derived a small refinement of a bound in Davies (2015) (eq. 4.34) that can be used as a tool for assessing the reliability of estimates of the posterior mean produced by conjugate gradients. Additionally, we showed a diagnostic suggested by Pleiss (2020) can be modified to obtain additive error bounds on the approximation quality of the posterior variance (eq. 4.43).

With both conjugate gradient lower bound maximization and barely biased log marginal likelihood estimation in chapter 4, we focused on determining termination criteria for iterative methods that provide a posteriori guarantees on the quality of the estimate of the log marginal likelihood, or a term in the log marginal likelihood, obtained. While this sort of guarantee does not appear to be strong enough to directly say anything about model selection in the workflow considered, it results in provably accurate estimation of quantities that may be of interest. This is particularly evident in barely biased log marginal likelihood estimation, where we obtain an estimate of the log marginal likelihood that is within a user-specified parameter, $\varepsilon > 0$, of an unbiased estimate.

Is it Easy to Use? The ease of use of approximate inference and model selection remains an obstacle to the adoption of approximate Gaussian process regression. In section 2.6 we discussed numerical issues in implementations of sparse variational Gaussian process regression. While these issues are

known within the Gaussian process research community, they are not often discussed in detail in the literature. We believe this discussion is a useful contribution in highlighting some numerical problems that currently make variational methods more difficult to use in practice. Additionally, the diagnostics used in section 2.4 could be used to automate the selection of the number of inducing points in sparse variational methods while ensuring accurate statistical inference is made. This approach could help to automate the procedure of performing sparse variational Gaussian process regression.

Chapter 3 discusses several methods for initializing inducing inputs for sparse variational Gaussian process regression with a priori performance guarantees. While on a practical basis we would not necessarily advocate for the use of either M-determinantal point processes or ridge leverage score sampling as a method for selecting inducing inputs over existing heuristics, rigorous theory for other initialization methods may lead to default initialization for inducing points that mean practitioners do not need to consider placement of inducing inputs when employing sparse variational inference in Gaussian process regression.

In chapter 4 we suggested methods for improving the reliability of model selection when using iterative methods to estimate the log marginal likelihood. In both conjugate gradient lower bound maximization and barely biased log marginal likelihood estimation a practitioner must specify a bias parameter, which trades-off between the accuracy of approximation to the log marginal likelihood and computation used. Unlike conventional stopping criteria used to determine how long to run an iterative method, this parameter has an interpretation as an upper bound on the number of additional nats of accuracy that could be gained by continuing to run the iterative method. We believe the relative interpretability of this parameter makes it easier to specify, improving the ease of use of iterative methods.

5.2 Reflections and Future Research Directions

Approximate Gaussian process regression is a reasonably mature form of approximate Bayesian inference, in that there are easy to use approximations with both a priori and a posteriori guarantees on the quality of inference. We believe there is significant progress to be made in terms of software for the use of approximate Gaussian process regression. In particular, while software packages such as `GPy` (GPy, 2012), `GPflow` (Matthews et al., 2017) and `GPyTorch` (Gardner et al., 2018) include approximate Gaussian process regression methods, they currently do not include many useful tools for practitioners to tell if they should trust the results obtained with these methods. The addition of diagnostics to commonly used Gaussian process toolboxes would help to ensure that these methods can be trusted. Moreover, increased use of the diagnostic tools currently available would undoubtedly highlight practical limitations in the application of these tools, motivating new theoretical questions.

On the side of theory, a comprehensive understanding of the interaction between model selection and scalable Gaussian process regression approximation remains elusive. Results in this direction, as well as progress on the related problem of approximate inference quality in hierarchical models involving

Gaussian process priors would be of great interest for extending the practical and reliable use of scalable approximate inference.

We hope that results and ideas from this thesis are useful to practitioners who want to reliably deploy methods involving Gaussian process regression at scale, and to researchers interested in improving scalable model selection and approximate inference in Gaussian process models.

References

Vincent Adam, Paul Chang, Mohammad Emtiyaz Khan, and Arno Solin. Dual parameterization of sparse variational Gaussian processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11474–11486, 2021. (Cited on page 90.)

Robert A. Adams and John J.F. Fournier. *Sobolev spaces*. Elsevier, 2003. (Cited on page 13.)

Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Advances in Neural Information Processing Systems (NIPS)*, pages 775–783, 2015. (Cited on pages 64, 76, and 87.)

Mauricio Álvarez and Neil D. Lawrence. Sparse convolved Gaussian processes for multi-output regression. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 57–64, 2008. (Cited on page 39.)

Sivaram Ambikasaran, Daniel Foreman-Mackey, Leslie Greengard, David W. Hogg, and Michael O'Neil. Fast direct methods for Gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):252–265, 2015. (Cited on page 25.)

Nima Anari, Shayan Oveis Gharan, and Alireza Rezaei. Monte Carlo Markov chain algorithms for sampling strongly Rayleigh distributions and determinantal point processes. In *Conference on Learning Theory (COLT)*, pages 103–115, 2016. (Cited on pages 68 and 69.)

Artem Artemev, David R. Burt, and Mark van der Wilk. Tighter bounds on the log marginal likelihood of Gaussian process regression using conjugate gradients. In *International Conference on Machine Learning (ICML)*, pages 362–372, 2021. (Cited on pages ix, 26, 45, 96, 109, 113, 124, 134, and 138.)

Hagai Attias. A variational Bayesian framework for graphical models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 209–215, 1999. (Cited on page 51.)

Erlend Aune, Daniel P. Simpson, and Jo Eidsvik. Parameter estimation in high dimensional Gaussian distributions. *Statistics and Computing*, 24:247–263, 2014. (Cited on page 108.)

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011. (Cited on pages 96 and 101.)

Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory (COLT)*, pages 185–209, 2013. (Cited on pages 64 and 87.)

Francis Bach. Playing with positive definite matrices – i: matrix monotony and convexity. Blog post, 2022. URL https://francisbach.com/matrix-monotony-and-convexity/. (Cited on page 61.)

François Bachoc. Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66:55–69, 2013. (Cited on page 17.)

François Bachoc. Asymptotic analysis of maximum likelihood estimation of covariance parameters for Gaussian processes: an introduction with proofs. In *Advances in Contemporary Statistics and Econometrics*, pages 283–303. Springer, 2021. (Cited on page 16.)

Zhaojun Bai, Gark Fahey, and Gene H. Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1):71–89, 1996. (Cited on pages 101 and 105.)

Simon Bartels and Philipp Hennig. Conjugate gradients for kernel machines. *Journal of Machine Learning Research (JMLR)*, 21:55–1, 2020. (Cited on page 135.)

Matthias Bauer, Mark van der Wilk, and Carl E. Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1533–1541, 2016. (Cited on pages 51 and 52.)

Mohamed-Ali Belabbas and Patrick J. Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences (PNAS)*, 106(2):369–374, 2009. (Cited on pages 66 and 67.)

Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011. (Cited on page 35.)

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc P. Deisenroth. Matérn Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12426–12437, 2020. (Cited on page 5.)

Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc P. Deisenroth, and Nicolas Durrande. Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2593–2601, 2021. (Cited on page 5.)

Mikio L. Braun. Accurate error bounds for the eigenvalues of the kernel matrix. *Journal of Machine Learning Research (JMLR)*, 7:2303–2328, 2006. (Cited on page 83.)

Thang D. Bui, Josiah Yan, and Richard E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research*, 18:3649–3720, 2017. (Cited on page 29.)

David R. Burt. Spectral methods in Gaussian process approximations. Master's thesis, University of Cambridge, 2018. (Cited on pages v, ix, 26, 59, 60, 61, 66, 71, and 137.)

David R. Burt, Carl E. Rasmussen, and Mark Van Der Wilk. Rates of Convergence for Sparse Variational Gaussian Process Regression. In *International Conference on Machine Learning (ICML)*, pages 862–871, 2019. (Cited on pages ix, 26, 59, 87, and 88.)

David R. Burt, Carl E. Rasmussen, and Mark van der Wilk. Variational orthogonal features. ArXiv, 2020a. URL https://arxiv.org/abs/2006.13170. (Cited on page 39.)

David R. Burt, Carl E. Rasmussen, and Mark van der Wilk. Convergence of Sparse Variational Inference in Gaussian Processes Regression. *Journal of Machine Learning Research*, 21:1–63, 2020b. (Cited on pages ix, 26, 59, 87, and 89.)

David R. Burt, Artem Artemev, and Mark van der Wilk. Barely biased learning for Gaussian process regression. In *I Can't Believe it's not Better, NeurIPS Workshop*, 2021. (Cited on pages ix, 26, 96, 130, and 134.)

Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic Complexity Theory*. Springer-Verlag, 1997. (Cited on page 21.)

Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Distributed adaptive sampling for kernel matrix approximation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1421–1429, 2017. (Cited on pages 64 and 88.)

Daniele Calandriello, Luigi Carratino, Alessandro Lazaric, Michal Valko, and Lorenzo Rosasco. Gaussian process optimization with adaptive sketching: Scalable and no regret. In *Conference on Learning Theory (COLT)*, pages 1–25, 2019. (Cited on pages 64, 70, 88, and 89.)

Daniele Calandriello, Michał Dereziński, and Michal Valko. Sampling from a k-DPP without looking at all items. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6889–6899, 2020. (Cited on page 88.)

Krzysztof Chalupka, Christopher K.I. Williams, and Iain Murray. A framework for evaluating approximation methods for Gaussian process regression. *Journal of Machine Learning Research (JMLR)*, 14: 333–350, 2013. (Cited on page 24.)

Benjamin Charlier, Jean Feydy, Joan Alexis Glaunès, François-David Collin, and Ghislain Durif. Kernel operations on the GPU, with autodiff, without memory overflows. *Journal of Machine Learning Research (JMLR)*, 22:1–6, 2021. (Cited on pages 97 and 120.)

Ching-An Cheng and Byron Boots. Variational inference for Gaussian process models with linear complexity. *Advances in Neural Information Processing Systems (NIPS)*, 30, 2017. (Cited on page 127.)

Badr-Eddine Chérief-Abdellatif. Consistency of ELBO maximization for model selection. In *Symposium on Advances in Approximate Bayesian Inference (AABI)*, pages 11–31, 2019. (Cited on page 51.)

Ali Çivril and Malik Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009. (Cited on page 69.)

Lehel Csató and Manfred Opper. Sparse On-Line Gaussian Processes. *Neural Computation*, 14(3): 641–668, 2002. (Cited on pages 24, 27, and 34.)

Kurt Cutajar, Michael Osborne, John Cunningham, and Maurizio Filippone. Preconditioning kernel matrices. In *International Conference on Machine Learning (ICML)*, 2016. (Cited on pages 100 and 115.)

Alexander Davies. *Effective Implementation of Gaussian Process Regression for Machine Learning*. PhD Thesis, University of Cambridge, 2015. (Cited on pages 28, 44, 89, 91, 94, 95, 106, 107, 112, and 138.)

Michał Dereziński, Daniele Calandriello, and Michal Valko. Exact sampling of determinantal point processes with sublinear time preprocessing. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 11542–11554, 2019. (Cited on pages 67 and 88.)

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL http://archive.ics.uci.edu/ml. (Cited on pages 119 and 121.)

Vincent Dutordoir, Nicolas Durrande, and James Hensman. Sparse Gaussian processes with spherical harmonic features. In *International Conference on Machine Learning (ICML)*, pages 2793–2802, 2020. (Cited on page 39.)

David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014. (Cited on page 4.)

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. (Cited on page 64.)

Giancarlo Ferrari-Trecate, Christopher K.I. Williams, and Manfred Opper. Finite-dimensional approximation of Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 218–224, 1999. (Cited on pages 65 and 88.)

Shai Fine and Katya Scheinberg. Efficient SVM training using low-rank kernel representations. *Journal of Machine Learning Research (JMLR)*, 2:243–264, 2001. (Cited on pages 48, 69, and 89.)

Leslie Foster, Alex Waagen, Nabeela Aijaz, Michael Hurley, Apolonio Luis, Joel Rinsky, Chandrika Satyavolu, Michael J. Way, Paul Gazis, and Ashok Srivastava. Stable and efficient Gaussian process calculations. *Journal of Machine Learning Research (JMLR)*, 10:857–882, 2009. (Cited on pages 69 and 89.)

Théo Galy-Fajou and Manfred Opper. Adaptive inducing points selection for Gaussian processes. In *Workshop on Continual Learning, ICML 2020*, 2020. (Cited on page 89.)

Jacob R. Gardner, Geoff Pleiss, Kilian Q. Weinberger, David Bindel, and Andrew G. Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. (Cited on pages xvii, xviii, xx, 69, 100, 108, 115, 116, 118, 119, 121, 122, 123, 125, 126, 131, 134, and 139.)

Guillaume Gautier, Guillermo Polito, Rémi Bardenet, and Michal Valko. DPPy: DPP sampling with Python. *Journal of Machine Learning Research (JMLR)*, 20:1–7, 2019. (Cited on page 67.)

Christopher J. Geoga, Mihai Anitescu, and Michael L. Stein. Scalable Gaussian process computations using hierarchical matrices. *Journal of Computational and Graphical Statistics*, 29(2):227–237, 2020. (Cited on pages 25 and 100.)

Zoubin Ghahramani. The `kin` family. http://www.cs.toronto.edu/ delve/data/kin/desc.html, 1996. (Cited on pages 119 and 121.)

Mark Gibbs and David J. C. MacKay. Efficient Implementation of Gaussian Processes. Technical report, University of Cambridge, 1997. (Cited on pages v, 25, 94, 95, 106, 107, 114, 115, 117, and 129.)

Alex Gittens and Michael W. Mahoney. Revisiting the Nyström method for improved large-scale machine learning. *Journal of Machine Learning Research (JMLR)*, 17(117):1–65, 2016. (Cited on pages 64 and 87.)

Gene H. Golub. Some modified matrix eigenvalue problems. *SIAM Review*, 15(2):318–334, 1973. Publisher: Society for Industrial and Applied Mathematics. (Cited on pages 105 and 131.)

Gene H. Golub and Gérard Meurant. *Matrices, moments and quadrature with applications*. Princeton University Press, 2009. (Cited on pages 94, 96, 104, and 105.)

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. John Hopkins University Press, 2013. (Cited on pages 94, 96, 99, 100, 102, 103, and 130.)

Gene H. Golub and John H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969. (Cited on pages 105 and 128.)

GPy. GPy: A gaussian process framework in python. http://github.com/SheffieldML/GPy, 2012. (Cited on page 139.)

Izrail S. Gradshteyn and Iosif M. Ryzhik. *Table of Integrals, Series, and Products*. Academic press, 2014. (Cited on page 78.)

Wolfgang Hackbusch. *Iterative solution of large sparse systems of equations*. Springer, 1994. (Cited on page 98.)

Nicholas Hale, Nicholas J. Higham, and Lloyd N. Trefethen. Computing a^α, $\log(a)$, and related matrix functions by contour integrals. *SIAM Journal of Numerical Analysis*, 46:2505–2523, 2008. (Cited on page 108.)

Insu Han, Dmitry Malioutov, and Jinwoo Shin. Large-scale log-determinant computation through stochastic Chebyshev expansions. In *International Conference on Machine Learning (ICML)*, page 908–917, 2015. (Cited on page 108.)

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290, 2013. (Cited on pages 31, 33, 89, and 90.)

James Hensman, Alexander G. de G. Matthews, Maurizio Filippone, and Zoubin Ghahramani. MCMC for variationally sparse Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*, volume 28, 2015a. (Cited on page 90.)

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. Scalable Variational Gaussian Process Classification. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 351–360, 2015b. (Cited on pages 31, 89, and 90.)

James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research (JMLR)*, 18:1–52, 2018. (Cited on pages 27 and 39.)

Jonathan Hermon and Justin Salez. Modified log-Sobolev inequalities for strong-Rayleigh measures. Arxiv, 2019. URL https://arxiv.org/abs/1902.02775. (Cited on page 68.)

Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49(6):409–436, 1952. (Cited on pages 25 and 93.)

Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, 2012. (Cited on pages 56, 57, 102, 153, 156, 157, and 158.)

Jonathan H. Huggins, Trevor Campbell, Mikołaj Kasprzak, and Tamara Broderick. Scalable Gaussian process inference with finite-data mean and variance guarantees. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 796–805, 2019. (Cited on pages 28, 29, 39, 40, 42, and 44.)

Jonathan H. Huggins, Mikołaj Kasprzak, Trevor Campbell, and Tamara Broderick. Validated variational inference via practical posterior error bounds. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1792–1802, 2020. (Cited on pages 28, 39, 40, 41, 91, and 138.)

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989. Publisher: Taylor & Francis. (Cited on page 101.)

Herman Kahn. *Use of Different Monte Carlo Sampling Techniques*. RAND Corporation, 1955. (Cited on page 134.)

Motonobu Kanagawa, Philipp Hennig, Dino Sejdinovic, and Bharath K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. Arxiv, 2018. URL https://arxiv.org/abs/1807.02582. (Cited on pages 6 and 35.)

Hyunjik Kim and Yee Whye Teh. Scaling up the Automatic Statistician: Scalable structure discovery using Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 575–584, 2018. (Cited on pages 28, 109, 113, and 123.)

George S. Kimeldorf and Grace Wahba. A correspondence between Bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, 41(2):495–502, 1970. (Cited on page 75.)

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015. (Cited on pages 117, 118, 121, and 131.)

Vladimir Koltchinskii and Evarist Giné. Random matrix approximation of spectra of integral operators. *Bernoulli*, 6(1):113–167, 2000. (Cited on page 71.)

Hermann König. *Eigenvalue distribution of compact operators*. Operator Theory: Advances and Applications. Birkhäuser, 2013. (Cited on page 11.)

Alex Kulesza and Ben Taskar. k-DPPs: Fixed-size determinantal point processes. In *International Conference on Machine Learning (ICML)*, pages 1193–1200, 2011. (Cited on pages 66 and 67.)

Miguel Lázaro-Gredilla, Joaquin Quiñonero-Candela, Carl E. Rasmussen, and Aníbal R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *The Journal of Machine Learning Research (JMLR)*, 11:1865–1881, 2010. (Cited on page 25.)

Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Fast DPP sampling for Nyström with application to kernel methods. In *International Conference on Machine Learning (ICML)*, pages 2061–2070, 2016a. (Cited on pages 64 and 88.)

Chengtao Li, Suvrit Sra, and Stefanie Jegelka. Gaussian quadrature for matrix inverse forms with applications. In *International Conference on Machine Learning (ICML)*, pages 1766–1775, 2016b. (Cited on page 127.)

Chi-Kwong Li and Gilbert Strang. An elementary proof of Mirsky's low rank approximation theorem. *Electronic Journal of Linear Algebra*, 36:694–697, 2020. (Cited on page 64.)

Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. (Cited on pages 19 and 120.)

Haitao Liu, Yew-Soon Ong, Xiaobo Shen, and Jianfei Cai. When Gaussian process meets big data: A review of scalable GPs. *IEEE transactions on neural networks and learning systems*, 31(11): 4405–4423, 2020. (Cited on page 24.)

Miguel Lázaro-Gredilla and Aníbal R. Figueiras-Vidal. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1087–1095, 2009. (Cited on page 39.)

David J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003. (Cited on pages 15 and 107.)

Wesley J. Maddox, Sanyam Kapoor, and Andrew Gordon Wilson. When are iterative Gaussian processes reliably accurate? In *ICML OPTML Workshop*, 2021. (Cited on page 134.)

Maren Mahsereci and Philipp Hennig. Probabilistic line searches for stochastic optimization. *Journal of Machine Learning Research (JMLR)*, 18:1–59, 2017. (Cited on page 117.)

Alexander G. de G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 231–239, 2016. (Cited on pages 31 and 32.)

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research (JMLR)*, 18(40):1–6, 2017. (Cited on pages 96, 117, 119, 120, and 139.)

Gérard Meurant. *The Lanczos and conjugate gradient algorithms: from theory to finite precision computations*. SIAM, 2006. (Cited on page 99.)

Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research (JMLR)*, 7(12), 2006. (Cited on page 6.)

Cameron Musco and Christopher Musco. Recursive sampling for the Nyström method. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3833–3845, 2017. (Cited on pages xvii, 63, 69, 70, 73, 74, 87, and 88.)

Dennis Nieman, Botond Szabó, and Harry van Zanten. Contraction rates for sparse variational approximations in Gaussian process regression. Arxiv, 2021. URL https://arxiv.org/abs/2109.10755. (Cited on pages 76, 77, 83, and 89.)

Jorge Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35 (151):773–782, 1980. (Cited on page 19.)

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009. (Cited on page 90.)

Aristeidis Panos, Petros Dellaportas, and Michalis K. Titsias. Fully scalable Gaussian processes using subspace inducing inputs. Arxiv, 2018. URL https://arxiv.org/abs/1807.02537. (Cited on page 90.)

Geoff Pleiss. *A Scalable and Flexible Framework for Gaussian Processes via Matrix-Vector Multiplication*. PhD thesis, Cornell, 2020. (Cited on pages 108 and 138.)

Geoff Pleiss, Jacob R. Gardner, Kilian Q. Weinberger, and Andrew Gordon Wilson. Constant-time predictive distributions for Gaussian processes. In *International Conference on Machine Learning (ICML)*, 2018. (Cited on pages 94, 107, 108, and 121.)

Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. *Lecture Notes for ECE563 (UIUC)*, 2014. (Cited on pages 30, 40, and 41.)

Andres Potapczynski, Luhuan Wu, Dan Biderman, Geoff Pleiss, and John P. Cunningham. Bias-free scalable Gaussian processes via randomized truncations. In *International Conference on Machine Learning (ICML)*, pages 8609–8619, 2021. (Cited on pages 123 and 134.)

Joaquin Quiñonero-Candela and Carl E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research (JMLR)*, 6:1939–1959, 2005. (Cited on page 24.)

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1177–1184, 2007. (Cited on page 25.)

Carl E. Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems (NIPS)*, pages 294–300, 2001. (Cited on page 15.)

Carl E. Rasmussen and Chris K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. (Cited on pages 4, 6, 13, 15, 17, 35, and 47.)

Christian P. Robert and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999. (Cited on page 69.)

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1657–1665, 2015. (Cited on pages 64 and 76.)

Hugh Salimbeni. Bayesian benchmarks. https://github.com/hughsalimbeni/bayesian_benchmarks, 2019. (Cited on page 119.)

Hugh Salimbeni, Ching-An Cheng, Byron Boots, and Marc Deisenroth. Orthogonally decoupled variational Gaussian processes. In *Advances in neural information processing systems (NeurIPS)*, volume 31, 2018. (Cited on page 127.)

Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 436–443, 2007. (Cited on page 117.)

Matthias W. Seeger, Christopher K.I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 254–261, 2003. (Cited on pages 24, 27, and 89.)

Matthias W. Seeger, Sham M. Kakade, and Dean P. Foster. Information consistency of nonparametric Gaussian process methods. *IEEE Transactions on Information Theory*, 54(5):2376–2382, 2008. (Cited on pages 9, 11, 77, 79, and 86.)

John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004. (Cited on page 39.)

John Shawe-Taylor, Christopher K.I. Williams, Nello Cristianini, and Jaz Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, 51(7):2510–2522, 2005. (Cited on pages 60, 71, and 74.)

Johnathan R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Carnegie Melon University, 1994. (Cited on page 94.)

Jiaxin Shi, Michalis K. Titsias, and Andriy Mnih. Sparse orthogonal variational inference for Gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. (Cited on pages 110 and 111.)

John Skilling. *Bayesian Numerical Analysis*, page 207–222. Cambridge University Press, 1993. (Cited on pages 25 and 106.)

Alexander J. Smola and Peter L. Bartlett. Sparse greedy Gaussian process regression. In *Advances in Neural Information Processing Systems (NIPS)*, pages 619–625, 2001. (Cited on page 24.)

Alexander J. Smola and Bernard Schölkopf. Sparse greedy matrix approximation for machine learning. In *International Conference on Machine Learning (ICML)*, pages 911–918, 2000. (Cited on page 88.)

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Advances in neural information processing systems (NIPS)*, pages 1257–1264, 2006. (Cited on page 24.)

Peter Sollich. Learning curves for Gaussian processes. *Advances in neural information processing systems (NIPS)*, pages 344–350, 1999. (Cited on page 11.)

Michael L. Stein. A comparison of generalized cross validation and modified maximum likelihood for estimating the parameters of a stochastic process. *The Annals of Statistics*, 18(3):1139 – 1157, 1990. (Cited on page 17.)

Michael L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 2012. (Cited on pages 6, 16, and 17.)

Ingo Steinwart and Clint Scovel. Mercer's theorem on general domains: On the interaction between measures, kernels, and RKHSs. *Constructive Approximation*, 35:363–417, 2012. (Cited on pages 10 and 11.)

William T. Stephenson, Soumya Ghosh, Tin D. Nguyen, Mikhail Yurochkin, Sameer K. Deshpande, and Tamara Broderick. Measuring the sensitivity of Gaussian processes to kernel choice. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022. (Cited on page 19.)

Josef Stoer and Roland Bulirsch. *Introduction to numerical analysis*, volume 3. Springer, 2002. (Cited on page 104.)

Amos J. Storkey. *Efficient Covariance Matrix Methods for Bayesian Gaussian Processes and Hopfield Neural Networks*. PhD thesis, Imperial College London, 1999. (Cited on page 25.)

Pieter Tans and Ralph Keeling. Trends in atmospheric carbon dioxide. https://gml.noaa.gov/ccgg/trends/data.html, 2022. Accessed: 2022-1-13. (Cited on page 47.)

Terence Tao. *Topics in Random Matrix Theory*. American Mathematical Society, 2012. (Cited on page 61.)

Michalis K. Titsias. Variational model selection for sparse Gaussian process regression. Technical report, University of Manchester, 2008. (Cited on page 56.)

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pages 567–574, 2009. (Cited on pages v, 24, 26, 27, 28, 29, 31, 34, 52, 90, 112, and 127.)

Michalis K. Titsias. Variational Inference for Gaussian and Determinantal Point Processes. In *Workshop on Advances in Variational Inference (NIPS)*, 2014. (Cited on pages 28, 45, and 46.)

Richard E. Turner and Maneesh Sahani. Two problems with variational expectation maximisation for time-series models. In *Bayesian Time series models*, pages 109–130. Cambridge University Press, 2011. (Cited on page 51.)

Shashanka Ubaru, Jie Chen, and Yousef Saad. Fast estimation of $f(a)$ via stochastic Lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, 38(4):1075–1099, 2017. Publisher: SIAM. (Cited on pages 108, 127, and 128.)

Sattar Vakili, Kia Khezeli, and Victor Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 82–90, 2021. (Cited on page 110.)

Sattar Vakili, Jonathan Scarlett, Da-shan Shiu, and Alberto Bernacchia. Improved convergence rates for sparse approximation methods in kernel-based learning. Arxiv, 2022. URL https://arxiv.org/abs/2202.04005. (Cited on page 89.)

Mark van der Wilk. *Sparse Gaussian process approximations and applications*. PhD thesis, University of Cambridge, 2019. (Cited on page 27.)

Mark van der Wilk, S. T. John, Artem Artemev, and James Hensman. Variational Gaussian process models without matrix inverses. In *Symposium on Advances in Approximate Bayesian Inference (AABI)*, 2020. (Cited on pages 90 and 121.)

Mark van der Wilk, Artem Artemev, and James Hensman. Improved inverse-free variational bounds for sparse Gaussian processes. In *Symposium on Advances in Approximate Bayesian Inference (AABI)*, 2022. (Cited on page 90.)

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. (Cited on page 117.)

Grace Wahba. *Spline models for observational data*. SIAM, 1990. (Cited on page 34.)

Ke Wang, Geoff Pleiss, Jacob R. Gardner, Stephen Tyree, Kilian Q. Weinberger, and Andrew Gordon Wilson. Exact Gaussian processes on a million data points. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. (Cited on pages xviii, xx, 118, 119, 120, 121, 122, 123, 126, 131, and 134.)

Limin Wang. *Karhunen-Lòeve expansions and their applications*. PhD thesis, London School of Economics and Political Science, 2008. (Cited on page 8.)

Holger Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004. (Cited on page 12.)

Jonathan Wenger, Geoff Pleiss, Philipp Hennig, John P. Cunningham, and Jacob R. Gardner. Reducing the variance of Gaussian process hyperparameter optimization with preconditioning. Arxiv, 2021. URL https://arxiv.org/abs/2107.00243. (Cited on pages 118 and 134.)

Harold Widom. Asymptotic behavior of the eigenvalues of certain integral equations. *Transactions of the American Mathematical Society*, 109(2):278–295, 1963. (Cited on pages 13, 79, and 86.)

Harold Widom. Asymptotic behavior of the eigenvalues of certain integral equations. II. *Archive for Rational Mechanics and Analysis*, 17(3):215–229, 1964. (Cited on page 79.)

Veit Wild and George Wynne. Variational Gaussian processes: A functional analysis view. Arxiv, 2021. URL https://arxiv.org/abs/2110.12798. (Cited on pages 27, 35, and 62.)

Veit Wild, Motonobu Kanagawa, and Dino Sejdinovic. Connections and equivalences between the Nyström method and sparse variational Gaussian processes. Arxiv, 2021. URL https://arxiv.org/abs/2106.01121. (Cited on page 34.)

Christopher K.I. Williams and Matthias W. Seeger. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems*, 2001. (Cited on page 24.)

Virginia V. Williams. Multiplying matrices faster than Coppersmith-Winograd. In *Symposium on Theory of Computing (STOC)*, pages 887–898, 2012. (Cited on page 21.)

Andrew Wilson and Hannes Nickisch. Kernel interpolation for scalable structured Gaussian processes (KISS-GP). In *International Conference on Machine Learning (ICML)*, pages 1775–1784, 2015. (Cited on page 25.)

James T. Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc P. Deisenroth. Pathwise conditioning of Gaussian processes. *Journal of Machine Learning Research (JMLR)*, 2020. (Cited on page 27.)

Yuling Yao, Aki Vehtari, Daniel P. Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In *International Conference on Machine Learning (ICML)*, 2018. (Cited on pages 23, 28, and 39.)

Dante C. Youla. The solution of a homogeneous Wiener-Hopf integral equation occurring in the expansion of second-order stationary random functions. *IRE Transactions on Information Theory*, 3 (3):187–193, 1957. (Cited on page 9.)

Yunong Zhang and William E. Leithead. Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *Journal of Statistical Computation and Simulation*, 77 (4):329–348, 2007. (Cited on page 108.)

Huaiyu Zhu, Christopher K.I. Williams, Richard Rohwer, and Michał Morciniec. Gaussian regression and optimal finite dimensional linear models. Technical report, Aston University, 1997. (Cited on pages 8, 12, 66, and 88.)

Appendix A

Matrix Properties

In this section, we recall matrix properties needed in the derivations in several chapters of this thesis. For a more complete treatment of matrix properties, we refer the reader to Horn and Johnson (2012).

A.1 Basic Definitions and Notation for Matrices

For $\mathbf{A} \in \mathbb{R}^{M\times N}$ we denote the transpose of $\mathbf{A}$ by $\mathbf{A}^\top \in \mathbb{R}^{N\times M}$. The transpose is defined as the matrix with entries $\mathbf{A}^\top_{nm} = \mathbf{A}_{mn}$ for $1 \le n \le N, 1 \le m \le M$. We say $\mathbf{A}$ is *symmetric* if $\mathbf{A}^\top = \mathbf{A}$, which can only be the case if $M = N$. We use $S^N \subset \mathbb{R}^{N\times N}$ to denote the space of symmetric matrices.

A.1.1 Positive Definite Matrices

Positive definite matrices are of particular interest to us, as the covariances between collections of random variables form positive definite matrices.

Definition A.1 (Positive (Semi-)Definite Matrix)**.** *A symmetric matrix $\mathbf{A} \in S^N$ is positive semi-definite, if for all $\mathbf{v} \in \mathbb{R}^N$,*

$$\mathbf{v}^\top \mathbf{A}\mathbf{v} \ge 0. \tag{A.1}$$

$\mathbf{A}$ is positive definite if additionally $\mathbf{v}^\top \mathbf{A}\mathbf{v} = 0 \Rightarrow \mathbf{v} = \mathbf{0}$.

We denote the space of positive semi-definite matrices by $S^N_+ \subset S^N$ and the space of positive definite matrices by $S^N_{++} \subset S^N_+$.

There is a natural identification between inner products (i.e. symmetric, bilinear, positive-definite forms) on $\mathbb{R}^N$ and matrices in S^N_{++}. In particular, for a positive definite matrix $\mathbf{A}$, we can define the bilinear function,

$$g_\mathbf{A} : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}, \quad g_\mathbf{A}(\mathbf{u},\mathbf{v}) = \mathbf{u}^\top \mathbf{A}\mathbf{v}. \tag{A.2}$$

It can be verified that $g_\mathbf{A}(\mathbf{u},\mathbf{v})$ is an inner product. This gives us the following definition,

Definition A.2. *For* $\mathbf{A} \in S^N_{++}$*, and for arbitrary* $\mathbf{u}, \mathbf{v} \in \mathbb{R}^N$ *define*

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{A}} := \mathbf{u}^\top \mathbf{A} \mathbf{v} \quad \textit{and} \quad \|\mathbf{u}\|_{\mathbf{A}} := \sqrt{\mathbf{u}^\top \mathbf{A} \mathbf{u}}. \tag{A.3}$$

The converse is also true: every inner product on $\mathbb{R}^N$ arises from some positive definite matrix, although we will not make use of this fact. For positive semi-definite matrices, a non-negative bilinear form and semi-norm can be defined analogously to definition A.2.

A.2 Eigenvalues, Singular Values and Matrix Norms

A.2.1 Eigenvalues

Eigenvalues are a central object of interest in much of linear algebra. While they can be defined for any square matrix, but we focus on the case of symmetric matrices, as it will suffice for our purposes and simplifies statements when considering matrices over the real numbers.

Definition A.3 (Eigenvalue and eigenvector)**.** *For* $\mathbf{A} \in S^N$*,* $\lambda \in \mathbb{R}$*,* $\mathbf{v} \in \mathbb{R}^N$*,* $\mathbf{v} \neq \mathbf{0}$*, if*

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}, \tag{A.4}$$

we say λ *is an eigenvalue of* $\mathbf{A}$ *and* $\mathbf{v}$ *is an eigenvector of* $\mathbf{A}$*.*

Definition A.4. *For* $\mathbf{A} \in S^N$ *and* $\lambda \in \mathbb{R}$*, an eigenvalue of* $\mathbf{A}$*, the* multiplicity *of* λ *is* $\dim(\{\mathbf{v} \in \mathbb{R}^N : \mathbf{A}\mathbf{v} = \lambda \mathbf{v}\})$.

Note that the set $\{\mathbf{v} \in \mathbb{R}^N : \mathbf{A}\mathbf{v} = \lambda \mathbf{v}\}$ is a subspace of $\mathbb{R}^N$, so definition A.4 makes sense. Eigenvectors associated to distinct eigenvalues of a matrix can be shown to be orthogonal. In addition, for eigenvalues with multiplicity greater than 1, we can choose the eigenvectors arbitrarily in the space. In particular, we lose no generality in assuming that each eigenvalue λ_n (counted with multiplicity) is associated to an eigenvector $\mathbf{v}_n$ such that $\{\mathbf{v}_n\}_{n=1}^N$ forms an orthonormal basis for $\mathbb{R}^N$. We will always take this convention moving forward. For $\mathbf{A} \in S^N$, we will use $\lambda_n(A)$ to denote the the n^{th} largest eigenvalue of $\mathbf{A}$, counted with multiplicity.

Theorem A.5 (Spectral Theorem for Real, Symmetric Matrices)**.** *Let* $\mathbf{A} \in S^N$*. Then there exists a* $\mathbf{U} \in \mathbb{R}^{N \times N}$ *with* $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$*, and* $\mathbf{\Lambda} \in \mathbb{R}^{N \times N}$ *a diagonal matrix such that* $\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$*. Moreover, the columns of* $\mathbf{U}$ *are eigenvectors of* $\mathbf{A}$ *and the entries of* $\mathbf{\Lambda}$ *are the eigenvalues of* $\mathbf{A}$*.*

For $\mathbf{A}$ a positive semi-definite matrix, we have $\lambda_n(\mathbf{A}) \geq 0$ for $1 \leq n \leq N$, and if $\mathbf{A}$ is positive definite, this inequality is strict.

Singular values

In some cases, we want analogues of the eigenvalues of positive semi-definite matrices in cases where a matrix may not even be square. The singular values are one such generalization of the eigenvalues.

For a matrix $\mathbf{A} \in \mathbb{R}^{N\times M}$, the singular values of $\mathbf{A}$ are the square roots of the eigenvalues of $\mathbf{A}^\top\mathbf{A}$. As $\mathbf{A}^\top\mathbf{A}$ is positive semi-definite, the singular values are non-negative real numbers. Moreover, if $\mathbf{A}$ is symmetric, we have the singular values are just the absolute value of its eigenvalues. In the case of positive semi-definite matrices where eigenvalues are non-negative, the singular values and eigenvalues coincide. However, this is not the case for general square matrices.

A.2.2 Norms of Matrices

Several matrix norms will be of interest to us. The concept of singular values will be important to the definition of some of these norms.

A.2.3 Induced Matrix Norms

Viewing a matrix as a linear operator from $\mathbb{R}^M \to \mathbb{R}^A$, we see that any pair of norms on $\mathbb{R}^M$ and $\mathbb{R}^N$ induces a norm on matrices in $\mathbb{R}^{M\times N}$:

Definition A.6 (Induced Matrix Norm)**.** *Given $\mathfrak{N}^M$ on $\mathbb{R}^M$ and $\mathfrak{N}^N$ on $\mathbb{R}^N$, and $A \in \mathbb{R}^{N\times M}$*

$$\|\mathbf{A}\|_{\mathfrak{N}^N,\mathfrak{N}^M} = \sup_{v\neq 0\in\mathbb{R}^M} \frac{\mathfrak{N}^N(\mathbf{Av})}{\mathfrak{N}^M(\mathbf{v})}. \tag{A.5}$$

It can be checked that $\|\mathbf{A}\|_{\mathfrak{N}^N,\mathfrak{N}^M}$ is a norm on $\mathbb{R}^{N\times M}$, referred to as an *induced norm*. The only induced norm that we will generally make use of is the case when $M = N$ and $\mathfrak{N}^N, \mathfrak{N}^M$ are taken to be the ℓ^2-norm. We will denote this by $\|\mathbf{A}\|_{\mathrm{Op}}$ and refer to it as the operator norm. In general, the operator norm is equal to the largest singular value. For positive semi-definite matrices, we therefore have,

$$\|\mathbf{A}\|_{\mathrm{Op}} = \lambda_1(\mathbf{A}). \tag{A.6}$$

A.2.4 Schatten Norms

We will also be interested in Schatten norms.

Definition A.7 (Schatten-p Norm)**.** *For $\mathbf{A} \in \mathbb{R}^{N\times M}$ the Schatten-p norm of $\mathbf{A}$ is the ℓ^p-norm of its singular values.*

We will denote the Schatten-p norm as $\|\mathbf{A}\|_{\mathrm{Sc},p}$. Only two cases will be of interest to us. When $p = \infty$, and $M = N$ we have $\|\mathbf{A}\|_{\mathrm{Sc},\infty} = \|\mathbf{A}\|_{\mathrm{Op}}$. If additionally $\mathbf{A}$ is positive semi-definite, we have $\|\mathbf{A}\|_{\mathrm{Sc},\infty} = \lambda_1(\mathbf{A})$, where $\lambda_1(\mathbf{A})$ denotes the largest eigenvalue of $\mathbf{A}$.

The other case of interest is $\|\mathbf{A}\|_{\mathrm{Sc},1}$, also referred to as the *nuclear norm*. In the case when $\mathbf{A} \in S^N_+$ using the connection between eigenvalues and singular values we have,

$$\|\mathbf{A}\|_{\mathrm{Sc},1} = \sum_{n=1}^{N} \lambda_n(\mathbf{A}). \tag{A.7}$$

A.3 Trace and Determinant

Definition A.8 (Trace of Square matrix)**.** *For* $\mathbf{A} \in \mathbb{R}^{N\times N}$*, let*

$$\operatorname{tr}\mathbf{A} = \sum_{n=1}^{N} \mathbf{A}_{nn} \tag{A.8}$$

denote the trace of $\mathbf{A}$.

The trace of a matrix has several important properties that we use throughout the text.

Proposition A.9 (Trace and Eigenvalues)**.** *For* $\mathbf{A} \in S^N$, $\operatorname{tr}\mathbf{A} = \sum_{n=1}^{N} \lambda_n(\mathbf{A})$*, where* $\{\lambda_n(\mathbf{A})\}_{n=1}^{N}$ *denote the eigenvalues of* $\mathbf{A}$ *counted with multiplicity.*

Allowing for complex-valued eigenvalues and counting with algebraic multiplicity, proposition A.9 holds for general square matrices; see Horn and Johnson (2012, Page 50) for a proof.

Another property of trace that we will use is the cyclic property:

Proposition A.10 (Cyclic property of the trace)**.** *For* $\mathbf{A} \in \mathbb{R}^{N\times M}, \mathbf{B} \in \mathbb{R}^{M\times N}$, $\operatorname{tr}(\mathbf{AB}) = \operatorname{tr}(\mathbf{BA})$.

Proposition A.10 can be verified by noting that

$$\operatorname{tr}(\mathbf{AB}) = \sum_{n=1}^{N}\sum_{n'=1}^{N} A_{nn'}B_{n'n} \tag{A.9}$$

is the entry-wise inner product of $\mathbf{A}$ and $\mathbf{B}^\top$ viewed as vectors in $\mathbb{R}^{(N\times M)}$.

A.3.1 Determinant

Definition A.11 (Determinant of Square Matrix)**.** *For* $\mathbf{A} \in \mathbb{R}^{N\times N}$ *the determinant of A is defined by*

$$\det\mathbf{A} = \begin{cases} A_{11} & N = 1 \\ \sum_{n=1}^{N}(-1)^{n+n'}\mathbf{A}_{nn'}\det\mathbf{A}_{\backslash n\backslash n'} & N > 1, \end{cases} \tag{A.10}$$

where $\mathbf{A}_{\backslash n\backslash n'} \in \mathbb{R}^{N-1\times N-1}$ *is the submatrix of* $\mathbf{A}$ *formed by deleting row n and column* n' *from* $\mathbf{A}$.

This definition is recursive, and generally difficult to work with. However, the determinant has several nice properties.

Proposition A.12 (Multiplicativity of Determinant, Horn and Johnson, 2012, 0.3.5)**.** *For* $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$ $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$.

Proposition A.13 (Determinant and Eigenvalues, Horn and Johnson, 2012, page 50)**.** *For* $\mathbf{A} \in S^N$, $\det\mathbf{A} = \prod_{n=1}^{N}\lambda_n(\mathbf{A})$ *where* $\{\lambda_n(\mathbf{A})\}_{n=1}^{N}$ *denote the eigenvalues of* $\mathbf{A}$ *counted with multiplicity.*

Proposition A.13 holds more generally for matrices in $\mathbb{R}^{N\times N}$ allowing for complex-valued eigenvalues and counting with algebraic multiplicity.

A.4 Block Matrices and Low-Rank Matrices

A.4.1 Block Matrices and the Schur Complement

Let $\mathbf{K} \in \mathbb{R}^{(N+M)\times(N+M)}$ be a block-matrix of the form,

$$\mathbf{K} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}, \tag{A.11}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}, \mathbf{B} \in \mathbb{R}^{N\times M}, \mathbf{C} \in \mathbb{R}^{M\times N}$ and $\mathbf{D} \in \mathbb{R}^{M\times M}$. We will generally be interested in the case when $\mathbf{K} \in S^{N+M}$, in which case $\mathbf{B} = \mathbf{C}^\top$ and $\mathbf{A}$ and $\mathbf{D}$ are symmetric.

If $\mathbf{D}$ is invertible, the *Schur complement of* $\mathbf{D}$ *in* $\mathbf{K}$, is the $N \times N$ matrix $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$. We recall two noteworthy properties of Schur complements.

Proposition A.14 (Horn and Johnson, 2012, 0.8.5)**.** *Let* $\mathbf{K} \in \mathbb{R}^{(N+M)\times(N+M)}$ *be a block-matrix of the form,*

$$\mathbf{K} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}. \tag{A.12}$$

Then $\det(\mathbf{K}) = \det(\mathbf{D})\det(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})$.

Proposition A.15 (Schur Complement of Positive Semi-definite Matrices, Horn and Johnson, 2012, Page 495)**.** *Let* $\mathbf{K} \in S^{N+M}$ *be of the form*

$$\mathbf{K} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\top & \mathbf{D} \end{pmatrix}. \tag{A.13}$$

Then $\mathbf{K}$ *is positive definite if and only if both* $\mathbf{D}$ *and* $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ *are positive definite. Moreover, if* $\mathbf{D}$ *is invertible, then* $\mathbf{K}$ *is positive semi-definite if and only if* $\mathbf{D}$ *is positive definite and* $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{B}^\top$ *is positive semi-definite.*

A.4.2 Low-Rank matrices

Proposition A.16 (Woodbury Matrix Lemma, Horn and Johnson, 2012, 0.7.4)**.** *For* $\mathbf{A} \in \mathbb{R}^{N\times N}, \mathbf{U} \in \mathbb{R}^{N\times M}, \mathbf{C} \in \mathbb{R}^{M\times M}, \mathbf{V} \in \mathbb{R}^{M\times N}$*, with* $\mathbf{A}$ *and* $\mathbf{C}$ *and* $(\mathbf{A} + \mathbf{U}\mathbf{C}\mathbf{V})^{-1}$ *invertible,*

$$(\mathbf{A} + \mathbf{U}\mathbf{C}\mathbf{V})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{C}^{-1} + \mathbf{V}\mathbf{A}^{-1}\mathbf{U})\mathbf{V}\mathbf{A}^{-1} \tag{A.14}$$

Proposition A.17 (Matrix Determinant Lemma)**.** *For* $\mathbf{A} \in \mathbb{R}^{N\times N}$*,* $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{N\times M}$*,* $\det(\mathbf{A} + \mathbf{U}\mathbf{V}^\top) = \det(\mathbf{I}_M + \mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U})\det(\mathbf{A})$.

Proposition A.17 can be deduced by applying proposition A.14 to the block matrix

$$\mathbf{K}_1 = \begin{pmatrix} \mathbf{A} & -\mathbf{U} \\ \mathbf{V}^\top & \mathbf{I} \end{pmatrix}. \tag{A.15}$$

A.5 Partial Ordering

There is a natural partial ordering on the space of symmetric matrices, often referred to as the *Loewner ordering*.

Definition A.18 (Partial Order on Symmetric Matrices)**.** *For* $\mathbf{A}, \mathbf{B} \in S^N$*, define*

$$\mathbf{A} \succ \mathbf{B} \iff \mathbf{A} - \mathbf{B} \in S^N_+. \tag{A.16}$$

By checking definitions, we see that $\succ$ is a non-strict partial order. We often use properties of Loewner order when bounding terms of interest in the main text.

Proposition A.19 (Properties of Partial Order, Horn and Johnson 2012, Corollary 7.7.4)**.** *For* $\mathbf{A}, \mathbf{B} \in S^N$*, satisfying* $\mathbf{A} \succ \mathbf{B}$*:*

- *For all* $\mathbf{v} \in \mathbb{R}^N$*,* $\|\mathbf{v}\|_\mathbf{A} \geq \|\mathbf{v}\|_\mathbf{B}$*;*
- *If* $\mathbf{A}, \mathbf{B} \in S^N_{++}$*,* $\mathbf{A}^{-1} \prec \mathbf{B}^{-1}$*;*
- $\lambda_n(\mathbf{A}) \geq \lambda_n(\mathbf{B})$ *for* $1 \leq n \leq N$*;*
- $\det(\mathbf{A}) \geq \det(\mathbf{B}) \geq 0$*.*

Proposition A.20. *Let* $\mathbf{A} \in S^N_+$*. Then* $\mathbf{A} \preceq \lambda_1(\mathbf{A})\mathbf{I}$*.*

Proof of proposition A.20. For any $\mathbf{v} \in \mathbb{R}^N$, we have

$$\mathbf{v}^\top(\lambda_1(\mathbf{A})\mathbf{I} - \mathbf{A})\mathbf{v} = \lambda_1(\mathbf{A})\|\mathbf{v}\|^2 - \|\mathbf{v}\|^2_\mathbf{A} \geq \lambda_1(\mathbf{A})\|\mathbf{v}\|^2 - \|\mathbf{A}\|_{\mathrm{Op}}\|\mathbf{v}\|^2 = 0, \tag{A.17}$$

where the inequality uses definition of the operator norm (definition A.6). □