Imperial College London

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

Orthogonal Sparse Variational Approximations for Gaussian Processes

Author:
Oskar Fernlund

Supervisor:
Mark van der Wilk

Submitted in partial fulfillment of the requirements for the MSc degree in Artificial Intelligence of Imperial College London

September 2022

Acknowledgements

I would like to thank Mark van der Wilk and Anish Dhir for all their support and guidance throughout this project; you pushed me to think deeply and to never settle for a superficial understanding.

Abstract

Gaussian processes are flexible nonparametric models which can represent distributions over functions. They have many desirable properties, like robustness against overfitting, but suffer from cubic computational complexity in the number of training datapoints. To be of practical use, Gaussian processes therefore require approximations. Sparse variational methods are one example of such an approximation, and are based on variational learning of *inducing points* which "summarise" the training dataset. In this thesis, we provide a thorough overview of the original sparse variational approximation, specifically in its collapsed form applied to regression (SGPR), then introduce more contemporary approximations like ODVGP and SOLVE-GP, which are based on orthogonal decomposition of the Gaussian process in the *reproducing kernel Hilbert space* induced by its kernel function. We make novel contribtions by deriving an orthogonal collapsed bound for regression, as well as a stable and efficient implementation of orthogonal sparse Gaussian process regression with a computational complexity which is 25-50% lower than that of the SGPR. We finish by probing the properties and behaviours of our orthogonal model and conclude that while it is cheaper than SGPR, it generally has sparser covariance, except for datasets comprised of well-separated clusters of datapoints, in which case it is capable of achieveing a near equivalent approximation to SGPR at lower cost.

Contents

Chapter 1

Introduction

Gaussian processes (GP's) are a class of stochastic processes which represent a generalisation of Gaussian distributions over finite dimensional vector spaces to infinite dimensional function spaces [1]. The elegance, flexibility and robustness to overfitting of Gaussian processes has led to their widespread adoption for a range of tasks including supervised learning, uncertainty quantification and optimisation. However, they suffer from $\mathcal{O}(N^3)$ computational complexity given N training datapoints, making them prohibitively expensive to train on even modestly sized datasets. This limitation has motivated extensive work on sparse approximations for Gaussian processes to improve scalability, e.g. [2], [3], [4], [5] and [6].

Sparse variational GP (SVGP) approximations [5], [6] based on variational learning of *inducing points*, or pseudo-datapoints which "summarise" the training dataset, have shown promise in mitigating the computational limitations of exact GP inference. Such methods maintain the original GP prior while enforcing sparse structures in the posterior, reducing the associated computational complexity to $\mathcal{O}(M^2N + M^3)$, where M is the number of inducing points. Furthermore, they allow for the use of minibatch gradient descent by subsampling the training data. SVGP methods have allowed for highly scalable GP models to be trained on as many as a billion datapoints [7], but as their computational complexity is still cubic in the number of inducing points, improving the flexibility of the posterior approximations remains an ongoing challenge [8].

Recently, new SVGP methods have been proposed in which the Gaussian process is reparameterised via orthogonal decomposition in the *reproducing kernel Hilbert space* (RKHS) induced by its kernel function [9], [10]. The authors of these "orthogonal" methods argue that they offer improved scalability in the number of inducing points as compared with standard SVGP methods and while impressive empirical results have been shown, insufficient work has been done investigating their limitations and linking empirical results with theory.

In this thesis we will illustrate, probe and benchmark various orthogonal sparse variational GP approximations under a unifying "structured covariance" view, show equivalencies and investigate when these methods excel and when they do not.

Chapter 2

Probabilistic Inference

This chapter provides a concise introduction to probabilistic inference, which is an important foundational topic for understanding Gaussian processes and sparse approximations thereof. Namely, we will review some common terminology and rules of probability theory which will come up in subsequent chapters, as well as explore approximate inference methods and why they are often necessary.

2.1 Thinking and Talking Like a Bayesian

In machine learning (ML), we construct models which "learn" from data in order to make predictions or decisions. These models are often comprised of parameters or latent variables which help describe the state of the system or process we are trying to capture. The process of drawing conclusions about these parameters or latent variables based on data is called *inference*. To an extent, inference is synonymous with learning; given increasing amounts of data, we can become more certain in our model's parameters or latent variables, and therefore it can make better predictions or decisions. Under the Bayesian statistical paradigm, performing the tasks of inference and prediction simply require manipulating random variables and their probability distributions [11].

2.1.1 Building and Manipulating Probabilistic Models

When building a probabilistic model, it can be useful to begin by considering the *joint distribution* over all random variables. Here, we will consider a very simple model comprised of some observed data $\mathbf{y}$ and unobserved parameters $\boldsymbol{\theta}$. The joint is therefore $p(\mathbf{y}, \boldsymbol{\theta})$. We can factorise our joint into a product of conditional and marginal distributions using the *product rule*:

$$p(\mathbf{y}, \boldsymbol{\theta}) = p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y})p(\mathbf{y}). \tag{2.1}$$

Note that some factorisations may be more useful or meaningful than others. If we are not interested in a variable, we can *marginalise* it out of our model using the *sum rule*:

$$p(\mathbf{y}) = \int p(\mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\theta} = \int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) d\boldsymbol{\theta}. \tag{2.2}$$

If we observe a variable, we can rearrange the product rule to *condition* on it:

$$p(\boldsymbol{\theta}|\mathbf{y}) = \frac{p(\mathbf{y}, \boldsymbol{\theta})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{2.3}$$

Note that the rightmost two formulations above are known as *Bayes' theorem*. These rules of probability theory form the basis of many of the calculations and manipulations we will do in this thesis.

2.1.2 Making Predictions

When making predictions, we generally want to account for the uncertainty in our model parameters. Conditioning on point values of parameters would result in predictions which are overconfident, so it is better to marginalise out the parameters from our model. If we want to predict the values of some new datapoints $\mathbf{y}^*$, we first add them to our joint, then condition on $\mathbf{y}$ and marginalise out $\boldsymbol{\theta}$:

$$p(\mathbf{y}^*|\mathbf{y}) = \int \frac{p(\mathbf{y}^*, \mathbf{y}, \boldsymbol{\theta})}{p(\mathbf{y})} d\boldsymbol{\theta} = \int \frac{p(\mathbf{y}^*|\boldsymbol{\theta})p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y})} d\boldsymbol{\theta} = \int p(\mathbf{y}^*|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}. \tag{2.4}$$

2.1.3 Common Terminology

Before continuing, it is worth going over some common terminology used in probabilistic inference. In our simple probabilistic model:

- $p(\mathbf{y}|\boldsymbol{\theta})$ is called the *likelihood* of the parameters. It describes the probability of the observed data as a function of the parameters. Likelihood is crucially *not* synonymous with probability – for fixed $\boldsymbol{\theta}$, $p(\mathbf{y}|\boldsymbol{\theta})$ describes the probability of $\mathbf{y}$; for fixed $\mathbf{y}$, it describes the likelihood of $\boldsymbol{\theta}$.
- $p(\mathbf{y})$ is called the *marginal likelihood*. It is a likelihood that has been integrated over the parameter space and has useful properties that make it a robust maximisation objective when optimising models.
- $p(\boldsymbol{\theta})$ is called the *prior* over the parameters. It represents our initial beliefs about the parameters before observing any data.
- $p(\boldsymbol{\theta}|\mathbf{y})$ is called the *posterior* over the parameters. It represents our beliefs about the parameters after observing the data.
- $p(\mathbf{y}^*|\mathbf{y})$ is called the *predictive* distribution. It describes our beliefs about possible unobserved values given the observed data.

2.2 Approximate Inference

In probabilistic inference, the key quantities in which we are interested are usually the marginal likelihood, posterior and predictive distributions, all of which involve some degree of integration. Except in select cases, these integrals are generally not computable in closed-form and thus exact inference is not analytically tractable. Instead, approximations must be made. Some of the most widely-used approximate inference techniques include sampling-based methods like *Markov chain Monte Carlo* (MCMC) and gradient-based methods like *variational inference* (VI) [1].

2.2.1 Variational Inference

Due largely to advancements in automatic differentiation and gradient-based optimisation, variational inference has become increasingly popular in recent years. Variational inference is an approximate inference method whereby an approximate (variational) posterior is restricted to belong to a class of tractable distributions and gradient-based optimisation routines are performed on the variational parameters with the objective of minimising the Kullback-Leibler (KL) divergence from the true posterior.

Kullback-Leibler divergence: The *Kullback-Leibler (KL) divergence* of one distribution $q(\mathbf{x})$ from another $p(\mathbf{x})$ is a statistical distance, or measure of how one distribution differs from the other, and is given by:

$$\mathrm{KL}\left[q(\mathbf{x})||p(\mathbf{x})\right] = \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x} = \mathbb{E}_{q(\mathbf{x})}\left[\log \frac{q(\mathbf{x})}{p(\mathbf{x})}\right]. \tag{2.5}$$

Note that $\mathrm{KL}\left[q(\mathbf{x})||p(\mathbf{x})\right] \geq 0$ and $\mathrm{KL}\left[q(\mathbf{x})||p(\mathbf{x})\right] = 0 \leftrightarrow q(\mathbf{x}) = p(\mathbf{x})$.

Directly computing the KL divergence of the variational posterior from the true posterior is generally not possible, as it requires knowledge of the quantity we wish to approximate to begin with. Instead, a lower bound can be constructed on the (log) marginal likelihood for which the slack is equal to the desired (intractable) KL divergence. The lower bound can be derived by writing out the KL divergence of the variational posterior from the true posterior and applying Bayes' rule. Let us once again consider our simple probabilistic model from before, and allow $q(\boldsymbol{\theta})$ to be a tractable approximation to the true posterior, $p(\boldsymbol{\theta}|\mathbf{y})$:

$$\begin{aligned}
\mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})] &= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathbf{y})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log \frac{q(\boldsymbol{\theta})p(\mathbf{y})}{p(\mathbf{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}\right] \\
&= \log p(\mathbf{y}) - \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathbf{y}|\boldsymbol{\theta})] + \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})].
\end{aligned} \tag{2.6}$$

Rearranging for the log marginal likelihood:

$$\log p(\mathbf{y}) = \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\mathbf{y})] + \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathbf{y}|\boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})]. \tag{2.7}$$

Recognising that the KL divergence of the variational posterior from the true posterior must satisfy Gibbs' inequality [1], we can construct a lower bound on the log marginal likelihood consisting of an expected log likelihood term and KL divergence term of the variational posterior from the prior:

$$\log p(\mathbf{y}) \geq \mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathbf{y}|\boldsymbol{\theta})] - \mathrm{KL}[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})] = \mathcal{L}. \tag{2.8}$$

The slack in the lower bound (often referred to as the *evidence lower bound*, or ELBO) is exactly equal to the KL divergence of the variational posterior from the true posterior, with equality if and only if this KL divergence is zero (or rather, the variational posterior exactly equals the true posterior). By maximising the ELBO using gradient-based optimisation routines, we can get an approximation of both the posterior and the marginal likelihood.

Chapter 3

Gaussian Processes

In this chapter, we will introduce Gaussian processes as a generalisation of the multivariate Gaussian distribution. We will review useful properties of Gaussian distributions and see how these properties make exact inference in Gaussian processes analytically tractable. We will finish by discussing the limitations of Gaussian processes and when approximate methods can be necessary.

3.1 Properties of Gaussian Distributions

The Gaussian distribution is the most widely used distribution for continuous random variables [12]. It has many convenient computational properties and is a foundational building block for understanding Gaussian processes. A multivariate Gaussian random variable (random vector) $x \in \mathbb{R}^N$ is fully characterised by its mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ and is denoted $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ or $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. It has a probability density function (pdf) given by:

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-\frac{N}{2}} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{3.1}$$

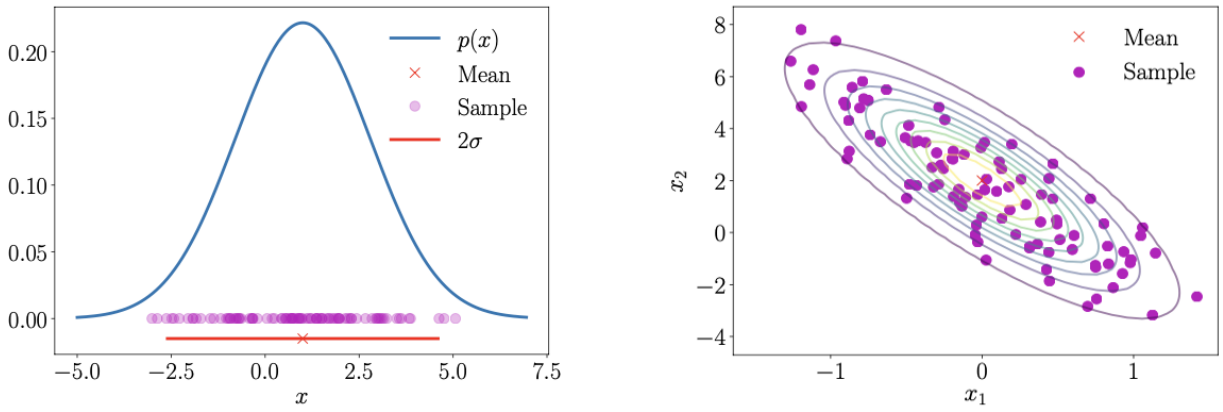

Figure 3.1: Left: univariate Gaussian distribution. The red cross indicates the mean and the red line indicates two standard deviations. Right: bivariate Gaussian distribution (viewed from the top down). The red cross indicates the mean and the coloured lines indicate density contours. Image source: [12].

We often work with log densities; the log pdf for a multivariate Gaussian is:

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

Each element in the vector $\boldsymbol{\mu}$ represents the expectation of the corresponding element in $\mathbf{x}$. The diagonals of the matrix $\boldsymbol{\Sigma}$ represent the variance of the corresponding elements in in $\mathbf{x}$; the off-diagonals represent covariance:

$$\boldsymbol{\mu} = \begin{bmatrix} \mathbb{E}_{p(\mathbf{x})}[x_1] \\ \mathbb{E}_{p(\mathbf{x})}[x_2] \\ \vdots \\ \mathbb{E}_{p(\mathbf{x})}[x_N] \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} \mathbb{V}_{p(\mathbf{x})}[x_1] & \mathbb{C}_{p(\mathbf{x})}[x_1, x_2] & \dots & \mathbb{C}_{p(\mathbf{x})}[x_1, x_N] \\ \mathbb{C}_{p(\mathbf{x})}[x_2, x_1] & \mathbb{V}_{p(\mathbf{x})}[x_2] & \dots & \mathbb{C}_{p(\mathbf{x})}[x_2, x_N] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}_{p(\mathbf{x})}[x_N, x_1] & \mathbb{C}_{p(\mathbf{x})}[x_N, x_2] & \dots & \mathbb{V}_{p(\mathbf{x})}[x_N] \end{bmatrix}, \tag{3.3}$$

where $\mathbb{E}$, $\mathbb{V}$ and $\mathbb{C}$ denote expectation, variance and covariance, respectively:

$$\begin{aligned} \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}] &= \int p(\mathbf{x})\mathbf{x} d\mathbf{x}, \\ \mathbb{V}_{p(\mathbf{x})}[\mathbf{x}] &= \mathbb{E}_{p(\mathbf{x})}\left[\mathbf{x}\mathbf{x}^\top\right] - \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]\,\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]^\top, \\ \mathbb{C}_{p(\mathbf{x},\mathbf{y})}[\mathbf{x},\mathbf{y}] &= \mathbb{E}_{p(\mathbf{x},\mathbf{y})}\left[\mathbf{x}\mathbf{y}^\top\right] - \mathbb{E}_{p(\mathbf{x},\mathbf{y})}[\mathbf{x}]\,\mathbb{E}_{p(\mathbf{x},\mathbf{y})}[\mathbf{y}]^\top. \end{aligned} \tag{3.4}$$

Note that the covariance between a random variable and itself equals its variance, and that covariance is symmetrical and positive definite by definition.

3.1.1 Marginalisation and Conditioning

Gaussian distributions have the convenient property of being closed under marginalisation and conditioning; that is to say that marginals and conditionals of Gaussians are also Gaussian. Let us explicitly write our Gaussian distribution in terms of concatenated states $[\mathbf{x}, \mathbf{y}]^\top$ using block matrices:

$$p\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_\mathbf{x}\\ \boldsymbol{\mu}_\mathbf{y}\end{bmatrix}, \begin{bmatrix}\boldsymbol{\Sigma}_{\mathbf{xx}} & \boldsymbol{\Sigma}_{\mathbf{xy}} \\ \boldsymbol{\Sigma}_{\mathbf{yx}} & \boldsymbol{\Sigma}_{\mathbf{yy}}\end{bmatrix}\right). \tag{3.5}$$

In the above expression, $\boldsymbol{\mu}_\mathbf{x} = \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]$ and $\boldsymbol{\mu}_\mathbf{y} = \mathbb{E}_{p(\mathbf{y})}[\mathbf{y}]$ are the marginal means of $\mathbf{x}$ and $\mathbf{y}$, respectively; $\boldsymbol{\Sigma}_{\mathbf{xx}} = \mathbb{C}_{p(\mathbf{x})}[\mathbf{x}, \mathbf{x}]$ and $\boldsymbol{\Sigma}_{\mathbf{yy}} = \mathbb{C}_{p(\mathbf{y})}[\mathbf{y}, \mathbf{y}]$ are the marginal covariances and $\boldsymbol{\Sigma}_{\mathbf{xy}} = \mathbb{C}_{p(\mathbf{x},\mathbf{y})}[\mathbf{x}, \mathbf{y}] = \boldsymbol{\Sigma}_{\mathbf{yx}}^\top$ is the cross covariance between $\mathbf{x}$ and $\mathbf{y}$. The marginals of our joint Gaussian are obtained by simply removing the unwanted rows and columns from the mean and covariance:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \boldsymbol{\Sigma}_{\mathbf{xx}}). \tag{3.6}$$

For conditionals, we apply the following formula:

$$p(\mathbf{x}|\mathbf{y}) = \mathcal{N}\left(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x} + \boldsymbol{\Sigma}_{\mathbf{xy}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_\mathbf{y}), \boldsymbol{\Sigma}_{\mathbf{xx}} - \boldsymbol{\Sigma}_{\mathbf{xy}}\boldsymbol{\Sigma}_{\mathbf{yy}}^{-1}\boldsymbol{\Sigma}_{\mathbf{yx}}\right). \tag{3.7}$$

3.1.2 Linear Transformations

Gaussian distributions have the additional convenient property of being closed under linear transformations, i.e. a linear transformation of a Gaussian random variable is also Gaussian. For example, if $\mathbf{x} \sim \mathcal{N}(\mathbf{a}, \mathbf{A})$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{b}, \mathbf{B})$ are independent Gaussian random vectors, then $\mathbf{z} = \mathbf{L}\mathbf{x} + \mathbf{y}$ is also a Gaussian random vector for a constant matrix $\mathbf{L}$. We can obtain the distribution on $\mathbf{z}$ by computing the expectation and variance of the transformation:

$$\begin{aligned} \mathbb{E}_{p(\mathbf{z})}[\mathbf{z}] &= \mathbb{E}_{p(\mathbf{x},\mathbf{y})}[\mathbf{L}\mathbf{x} + \mathbf{y}] = \mathbf{L}\,\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}] + \mathbb{E}_{p(\mathbf{y})}[\mathbf{y}] = \mathbf{L}\mathbf{a} + \mathbf{b}, \\ \mathbb{V}_{p(\mathbf{z})}[\mathbf{z}] &= \mathbb{V}_{p(\mathbf{x},\mathbf{y})}[\mathbf{L}\mathbf{x} + \mathbf{y}] = \mathbf{L}\,\mathbb{V}_{p(\mathbf{x})}[\mathbf{x}]\mathbf{L}^\top + \mathbb{V}_{p(\mathbf{y})}[\mathbf{y}] = \mathbf{L}\mathbf{A}\mathbf{L}^\top + \mathbf{B}, \\ \Rightarrow p(\mathbf{z}) &= \mathcal{N}(\mathbf{z}; \mathbf{L}\mathbf{a} + \mathbf{b}, \mathbf{L}\mathbf{A}\mathbf{L}^\top + \mathbf{B}). \end{aligned} \tag{3.8}$$

Linear transformation rules give us the following useful result for marginalising products of Gaussians:

$$\int \mathcal{N}(\mathbf{z}; \mathbf{L}\mathbf{x} + \mathbf{b}, \mathbf{B})\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})d\mathbf{x} = \mathcal{N}(\mathbf{z}; \mathbf{L}\mathbf{a} + \mathbf{b}, \mathbf{L}\mathbf{A}\mathbf{L}^\top + \mathbf{B}). \tag{3.9}$$

We arrive at this result by recognising that the distribution over $\mathbf{z}$ is equivalent to $\mathbf{z} = \mathbf{L}\mathbf{x} + \mathcal{N}(\mathbf{b}, \mathbf{B})$, which we know from eq. 3.8 may be expressed without explicit dependence on $\mathbf{x}$ and therefore can be pulled outside of the integral [13] (recall that probability densities must integrate to 1 by definition):

$$\int \mathcal{N}(\mathbf{z}; \mathbf{L}\mathbf{x} + \mathbf{b}, \mathbf{B})\mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})d\mathbf{x} = \mathcal{N}(\mathbf{z}; \mathbf{L}\mathbf{a} + \mathbf{b}, \mathbf{L}\mathbf{A}\mathbf{L}^\top + \mathbf{B}) \cancelto{1}{\int \mathcal{N}(\mathbf{x}; \mathbf{a}, \mathbf{A})d\mathbf{x}}. \tag{3.10}$$

3.1.3 Kullback-Leibler Divergence

The KL divergence between two N-dimensional Gaussian distributions $q(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ and $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p)$ has an analytic form:

$$\mathrm{KL}\left[q(\mathbf{x})||p(\mathbf{x})\right] = \frac{1}{2}\left\{\mathrm{Tr}\left(\boldsymbol{\Sigma}_p^{-1}\boldsymbol{\Sigma}_q\right) + (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^\top \boldsymbol{\Sigma}_p^{-1}(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q) - N + \log\frac{|\boldsymbol{\Sigma}_p|}{|\boldsymbol{\Sigma}_q|}\right\}. \tag{3.11}$$

3.2 Learning Functions with Gaussian Processes

If we wish to do inference on function values, one possibility is to do so indirectly by representing the function as a weighted sum of *basis functions* and specifying prior distributions over the weights. This is the approach we take in standard basis function regression. E.g. for some basis functions $\phi_m(x)$:

$$f(x) = \sum_{m=1}^{M} w_m \phi_m(x), \qquad p(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \sigma^2\mathbf{I}). \tag{3.12}$$

Another possibility, which has many computational and representational benefits, is to directly specify prior distributions over function values using a *Gaussian process* [14].

3.2.1 From Vector-Space to Function-Space

Gaussian processes (GP's) are a class of stochastic processes which represent a generalisation of Gaussian distributions over finite dimensional vector spaces to infinite dimensional function spaces [1]. This generalisation is motivated by the *Kolmogorov extension theorem*, which informally states that if marginally consistent joint distributions may be defined on finite subsets of an input space, then a probability measure may be defined on the entire space [11]. Marginal consistency is satisfied if we obtain the same result in specifying a joint distribution over some variables directly as when specifying joint distribution over a larger set and marginalising out the unwanted variables, i.e.:

$$p_{\text{marginal}}(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{y}) d\mathbf{y} = p_{\text{direct}}(\mathbf{x}). \tag{3.13}$$

The Kolmogorov extension theorem implies that it is valid to directly specify a distribution over functions, provided its finite dimensional marginals are consistent. A Gaussian process is such a distribution over functions where all finite dimensional marginals are Gaussian distributed [14]. Whereas a (multivariate) Gaussian distribution is completely defined by a mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, a Gaussian process is completely defined by a mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}_{p(f(\mathbf{x}))}[f(\mathbf{x})], \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{C}_{p(f(\mathbf{x}), f(\mathbf{x}'))}[f(\mathbf{x}), f(\mathbf{x}')]. \end{aligned} \tag{3.14}$$

A function distributed according to a GP is denoted $p(f(\mathbf{x})) = \mathcal{N}(f(\mathbf{x}); m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ or $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. Usually, we take the mean function to be zero for notational simplicity, although it is not necessary to do so [15]. The covariance function (or *kernel*) specifies the covariance between pairs of random variables; there are many possible choices of kernel, and selection is often problem-dependent, but the most widely used is the infinitely differentiable *squared-exponential* (SE) kernel:

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{||\mathbf{x} - \mathbf{x}'||}{2l^2}\right). \tag{3.15}$$

A Gaussian process with a squared exponential kernel can be shown to be equivalent to a Bayesian linear regression model with an *infinite* number of basis functions [15]. The characteristics of a kernel and how it restricts covariance are governed by its *hyperparameters* $\boldsymbol{\theta}$. For the squared exponential kernel above, we have two hyperparameters: the variance σ^2 and lengthscale l. The KL divergence between two Gaussian processes $\mathrm{KL}[f(\mathbf{x})||g(\mathbf{x})]$ is well-defined if and only if $f(\mathbf{x})$ and $g(\mathbf{x})$ share the same kernel hyperparameters [16].

We can probe our Gaussian process at any finite number of inputs; the marginal distribution over N function evaluations $f(\mathbf{X})$ at inputs $\mathbf{X}$ is Gaussian distributed:

$$f(\mathbf{X}) = \begin{bmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} m(\mathbf{x}_1) \\ m(\mathbf{x}_2) \\ \vdots \\ m(\mathbf{x}_N) \end{bmatrix}, \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_N) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_2) & \dots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix} \right). \tag{3.16}$$

3.2.2 Exact Inference

Gaussian processes may be used for both regression and classification tasks, but we will focus on regression in this thesis. In a simple regression setting, we assume there is a noiseless latent function f which maps input values $\mathbf{x} \in \mathbb{R}^D$ to target values $y \in \mathbb{R}^1$ that we wish to capture, but our observations y are corrupted by independent and identically distributed (i.i.d.) Gaussian noise:

$$y = f(\mathbf{x}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2). \tag{3.17}$$

We stack N input vectors $\{\mathbf{x}_n\}_{n=1}^N$ in the matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, and denote the vector of function values at these points $f(\mathbf{X})$ or $\mathbf{f}$ for short. We similarly stack the N noisy observations corresponding to the inputs $\{y_n\}_{n=1}^N$ in the vector $\mathbf{y} \in \mathbb{R}^{N \times 1}$. For notational brevity, we will drop conditioning on inputs, i.e. we write $p(\mathbf{y}|\mathbf{f}, \mathbf{X})$ as $p(\mathbf{y}|\mathbf{f})$. Due to our i.i.d. assumption, our likelihood may be factorised as:

$$p(\mathbf{y}|\mathbf{f}) = \prod_{n=1}^{N} p(y_n|f_n) = \prod_{n=1}^{N} \mathcal{N}(y_n; f_n, \sigma^2). \tag{3.18}$$

We put a zero-mean GP prior on the latent function. For further notational brevity, we denote covariance of the distribution of the N function values $\mathbf{f}$ as $\mathbf{K_{ff}}$ (note that while $\mathbf{K_{ff}}$ is the covariance of $\mathbf{f}$, it is a function of $\mathbf{X}$ as the kernel uses input values to compute covariances):

$$p(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \mathbf{0}, \mathbf{K_{ff}}). \tag{3.19}$$

If we are interested in predicting the latent function at some new inputs $\mathbf{X}_*$, we add $f(\mathbf{X}_*) = \mathbf{f}_*$ to our model. We can express our joint prior over the observations $\mathbf{y}$ and function values at the training locations $\mathbf{f}$ and test locations $\mathbf{f}_*$ in block form:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{f} \\ \mathbf{f}_* \end{bmatrix}; \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K_{ff}} + \sigma^2 \mathbf{I_N} & \mathbf{K_{ff}} & \mathbf{K_{f*}} \\ \mathbf{K_{ff}} & \mathbf{K_{ff}} & \mathbf{K_{f*}} \\ \mathbf{K_{*f}} & \mathbf{K_{*f}} & \mathbf{K_{**}} \end{bmatrix} \right). \tag{3.20}$$

We can marginalise out the function values at the training locations $\mathbf{f}$ by simply dropping the corresponding rows and columns from the mean and covariance:

$$p(\mathbf{y}, \mathbf{f}_*) = \mathcal{N}\left(\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix}; \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K_{ff}} + \sigma^2 \mathbf{I_N} & \mathbf{K_{f*}} \\ \mathbf{K_{*f}} & \mathbf{K_{**}} \end{bmatrix} \right). \tag{3.21}$$

We then obtain the functional posterior at the test locations $p(\mathbf{f}_*|\mathbf{y})$ using Gaussian conditioning rules:

$$p(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}\left(\mathbf{f}_*; \mathbf{K_{*f}}[\mathbf{K_{ff}} + \sigma^2 \mathbf{I_N}]^{-1}\mathbf{y}, \mathbf{K_{**}} - \mathbf{K_{*f}}[\mathbf{K_{ff}} + \sigma^2 \mathbf{I_N}]^{-1}\mathbf{K_{f*}}\right). \tag{3.22}$$

Note that in basis function regression, we do inference on parameters and thus the posterior is a distribution over parameters. Making predictions then involves computing a separate predictive distribution by marginalising over the product of the likelihood and posterior, as shown in eq. 2.4. In Gaussian process regression, we do inference directly on function values and thus our posterior is

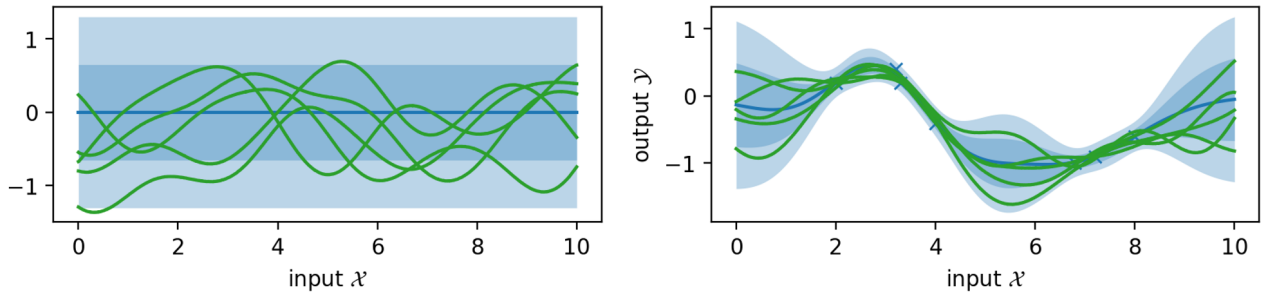

Figure 3.2: Samples from a GP prior (right) and posterior (left) using a squared exponential kernel. The mean is shown in blue and samples are shown in green. The blue shaded regions indicate $\pm$ 2 standard deviations from the mean. Image source: [11].

a distribution over functions which can be used to make predictions directly. If we wish to predict the values of some noisy observations $\mathbf{y}_*$ rather than the noiseless latent function, we can obtain the predictive distribution $p(\mathbf{y}_*|\mathbf{y})$ by simply adding the data noise to the posterior covariance:

$$p(\mathbf{y}_*|\mathbf{y}) = \mathcal{N}\left(\mathbf{f}_*; \mathbf{K}_{*\mathbf{f}}[\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I}_\mathbf{N}]^{-1}\mathbf{y}, \mathbf{K}_{**} + \sigma^2\mathbf{I}_\mathbf{N} - \mathbf{K}_{*\mathbf{f}}[\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I}_\mathbf{N}]^{-1}\mathbf{K}_{\mathbf{f}*}\right). \tag{3.23}$$

Training a Gaussian process involves maximising the marginal likelihood of the training data by adjusting the kernel hyperparameters $\boldsymbol{\theta}$ using gradient-based optimisation. For regression, the marginal likelihood and its gradient with respect to the kernel hyperparameters are available in closed-form:

$$\begin{aligned} \log p(\mathbf{y}) &= -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I}_\mathbf{N}| - \frac{1}{2}\mathbf{y}^\top(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I}_\mathbf{N})^{-1}\mathbf{y}, \\ \frac{\partial}{\partial\boldsymbol{\theta}}\log p(\mathbf{y}) &= -\frac{1}{2}\operatorname{Tr}\left[\left(\mathbf{K}_{\mathbf{ff}}^{-1}\mathbf{y}\mathbf{y}^\top\mathbf{K}_{\mathbf{ff}} - \mathbf{K}_{\mathbf{ff}}^{-1}\right)\frac{\partial\mathbf{K}_{\mathbf{ff}}}{\partial\boldsymbol{\theta}}\right]. \end{aligned} \tag{3.24}$$

A stable and efficient implementation of Gaussian process regression is shown in algorithm 1.

Algorithm 1 Log marginal likelihood and predictions for exact Gaussian process regression via Cholesky decomposition. $\mathbf{A} = \mathbf{B}\backslash\mathbf{C}$ denotes the solution to $\mathbf{BA} = \mathbf{C}$.

Input: $\mathbf{X}$ (training inputs), $\mathbf{y}$ (targets), k (kernel), σ^2 (noise), $\mathbf{X}_*$ (test inputs)

1: $\mathbf{K}_{\mathbf{ff}} = k(\mathbf{X}, \mathbf{X})$
2: $\mathbf{L} = \text{Cholesky}(\mathbf{K}_{\mathbf{ff}} + \sigma^2\mathbf{I}_\mathbf{N})$
3: $\boldsymbol{\alpha} = \mathbf{L}\backslash\mathbf{y}$
4: $\log p(\mathbf{y}) = -\frac{1}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha} - \sum_i \log \mathrm{L}_{\mathrm{ii}} - \frac{N}{2}\log(2\pi)$
5: $\mathbf{K}_{**} = k(\mathbf{X}_*, \mathbf{X}_*)$, $\mathbf{K}_{\mathbf{f}*} = k(\mathbf{X}, \mathbf{X}_*)$
6: $\boldsymbol{\beta} = \mathbf{L}\backslash\mathbf{K}_{\mathbf{f}*}$
7: $\boldsymbol{\mu}_* = \boldsymbol{\beta}^\top\boldsymbol{\alpha}$
8: $\boldsymbol{\Sigma}_{**} = \mathbf{K}_{**} - \boldsymbol{\beta}^\top\boldsymbol{\beta}$
9: **return:** $\log p(\mathbf{y})$ (log marginal likelihood), $\boldsymbol{\mu}_*$ (mean), $\boldsymbol{\Sigma}_{**}$ (covariance)

In practice, we generally use *Cholesky decompositions* to improve the efficiency and stability of our implementation, as while decompositions are computationally expensive, they make follow-up operations like inverses and determinants cheaper.

Cholesky decomposition: The Cholesky decomposition of a symmetric positive-definite matrix $\mathbf{A}$ is a decomposition into the product of a lower triangular matrix $\mathbf{L}$ and its transpose.

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top. \tag{3.25}$$

An in-depth discussion of exact inference in Gaussian processes including practical tips and tricks for implementation may be found in [15].

3.2.3 Sparse Approximations

Although Gaussian processes have many desirable properties, they suffer from computational limitations. Interestingly, the source of these limitations is different from the usual reasons that make exact inference challenging (as discussed in the previous chapter). While the key quantities of interest (i.e. the posterior, predictive, marginal likelihood and its gradient) are all *analytically* tractable for Gaussian process regression, they involve matrix inversions of $\mathcal{O}(N^3)$ computational complexity where N is the number of training datapoints, making them *computationally* intractable for even modestly sized datasets. Many approximate methods have been proposed to address these computational limitations. The simplest of these methods are based on low-rank kernel matrix approximations, Nyström kernel matrix approximations and subsampling [15]. Better methods have been proposed using a set of noiseless "inducing points" to summarise the dataset.

In inducing point approximations, the M quantities learned by our Gaussian process are function values (*inducing outputs*) at specific input locations (*inducing inputs*). The intuition behind inducing point approximations is that there is generally a lot of redundant information in learning the distribution of a latent function from many noisy observations, and since the GP prior places strong constraints on the values of neighbouring outputs, we can effectively "summarise" our dataset with a small number of strongly constrained input/output pairs (i.e. $M \ll N$) to obtain an approximate posterior that is close to the exact GP posterior. The learning objective becomes: "which M input/output pairs would result in an approximate posterior which is close to the exact GP posterior?".

Inducing point methods can be divided into two primary categories: methods involving approximations of the exact GP model (e.g. DTC, FITC [3]) and sparse variational methods [5], [6]. Sparse variational methods are some of the most promising Gaussian process approximations, as they retain many of the advantages of the exact model [11], and are the focus of this thesis. A comprehensive overview of approximate methods for Gaussian processes may be found in [11].

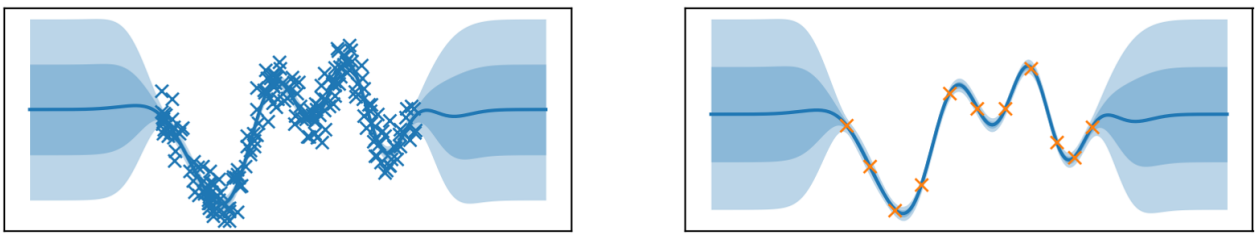

Figure 3.3: An example of a GP posterior obtained from learning many noisy observations (left) and few noiseless function values (right) on Snelson and Gharahmani's dataset [4]. The posteriors are plotted with their means $\pm$ 2 standard deviations and are very similar. Image source: [11].

Chapter 4

Sparse Variational Gaussian Processes

Sparse variational Gaussian processes (SVGP's) are based on variational learning of inducing points and in many cases can result in high quality posterior approximations at significantly reduced computational cost as compared with exact GP inference. In this chapter, we will provide an in-depth summary of the original SVGP model as proposed by Titsias in 2009 [5]. We will give a detailed outline of some of the key calculations shown by Titsias in [17], as well as Hensman and Matthews' derivation of a stable and efficient implementation for regression in [18], including intermediate steps where possible. Our aim is to provide the reader with a rigorous understanding of sparse variational GP's before introducing more complex orthogonal variations in the next chapter.

4.1 Model Formulation and Approximate Posterior

To formulate a sparse variational Gaussian process, we introduce a set of M inducing points $\mathbf{u} = f(\mathbf{Z})$ to our joint probability model:

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u})p(\mathbf{u}). \tag{4.1}$$

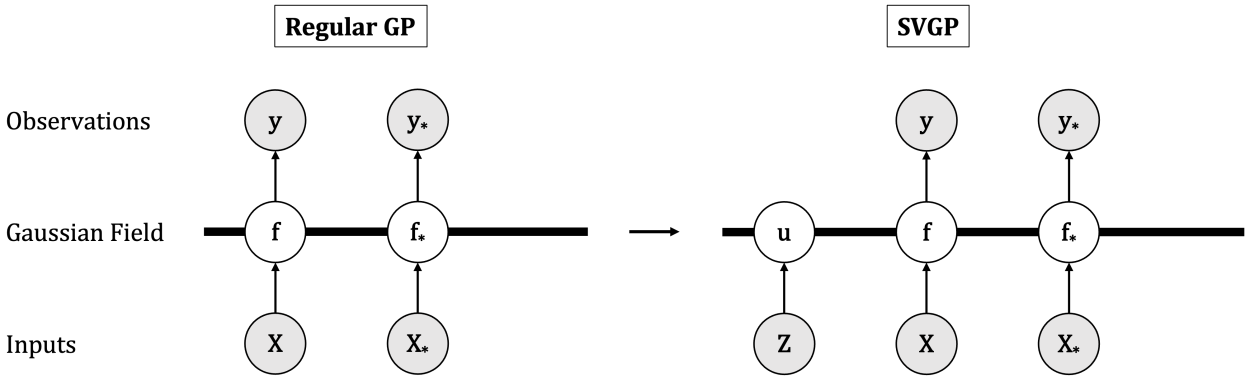

Figure 4.1: Directed graphical model for a regular Gaussian process (right) and a sparse variational Gaussian process (left) using "thick bar notation" to denote a Gaussian field. Observable variables are shaded in grey; unobservable variables are left unshaded.

We create an approximate (variational) posterior over the function values and inducing outputs $q(\mathbf{f}, \mathbf{u}) \approx p(\mathbf{f}, \mathbf{u}|\mathbf{y})$ by assuming the following factorisation:

$$q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f}|\mathbf{u})q(\mathbf{u}). \tag{4.2}$$

I.e., we keep the prior conditional $p(\mathbf{f}|\mathbf{u})$ but put a tractable variational distribution $q(\mathbf{u})$ over the inducing outputs; specifically, we let $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m_u}, \mathbf{S_u})$. The prior conditional is obtained through Gaussian conditioning rules on the joint prior over the function values and inducing outputs $p(\mathbf{f}, \mathbf{u})$:

$$\begin{aligned} p(\mathbf{f}, \mathbf{u}) &= \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{u}\end{bmatrix}; \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K_{ff}} & \mathbf{K_{fu}}\\ \mathbf{K_{uf}} & \mathbf{K_{uu}}\end{bmatrix}\right), \\ \Rightarrow p(\mathbf{f}|\mathbf{u}) &= \mathcal{N}(\mathbf{f}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}). \end{aligned} \tag{4.3}$$

Noticing that the prior conditional can be expressed as a linear transformation of the inducing outputs, the approximate functional posterior $q(\mathbf{f})$ can be obtained using Gaussian linear transformation rules:

$$\begin{aligned} q(\mathbf{f}) &= \int q(\mathbf{f}, \mathbf{u}) d\mathbf{u} = \int p(\mathbf{f}|\mathbf{u}) q(\mathbf{u}) d\mathbf{u} \\ &= \int \mathcal{N}(\mathbf{f}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}) \mathcal{N}(\mathbf{u}; \mathbf{m_u}, \mathbf{S_u}) d\mathbf{u} \\ &= \mathcal{N}(\mathbf{f}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u}, \mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}). \end{aligned} \tag{4.4}$$

This result implies a Gaussian process:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(\mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{m_u}, k(\mathbf{x}, \mathbf{x}) - \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{k_u}(\mathbf{x})\right). \tag{4.5}$$

4.1.1 Posterior Decomposition

We observe that the approximate GP posterior can be decomposed as a sum of two independent Gaussian processes [11] (noted similarly in [19]):

$$\begin{aligned} f(\mathbf{x}) &= g(\mathbf{x}) + h(\mathbf{x}), \\ g(\mathbf{x}) &\sim \mathcal{GP}\left(\mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{m_u}, \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{S_u} \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x})\right), \\ h(\mathbf{x}) &\sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}) - \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x})\right). \end{aligned} \tag{4.6}$$

For a fixed set of inducing points, $g(\mathbf{x})$ has a similar predictive distribution to a parametric regression model with M basis functions and contains the degrees of freedom that the approximate posterior is able to learn. It can be adjusted after observing data and it is the sole contributor to the posterior mean. $h(\mathbf{x})$ represents the degrees of freedom which the approximate posterior is not able to learn, and provides the nonparametric uncertainty in the prior. It remains fixed after observing data. The approximate posterior variance can only be reduced by observations which are near inducing points, which highlights the importance of proper inducing point placement when training a sparse variational Gaussian process.

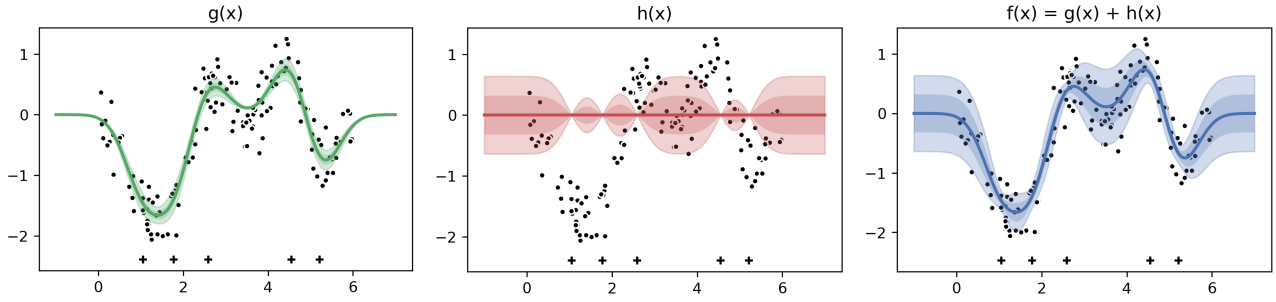

Figure 4.2: SVGP posterior with $M = 5$ inducing points (right) split into its constituent processes $g(\mathbf{x})$ (left) and $h(\mathbf{x})$ (centre) on Snelson and Gharahmani's dataset [4]. Inducing inputs are shown as black "plus" symbols. $g(\mathbf{x})$ is adjustable, while $h(\mathbf{x})$ remains unchanged after observing data.

4.2 Variational Lower Bound

We can construct a variational lower bound on the marginal likelihood by writing out the KL divergence of the approximate posterior $q(\mathbf{f}, \mathbf{u})$ from the true posterior $p(\mathbf{f}, \mathbf{u}|\mathbf{y})$ and applying Bayes' rule:

$$\begin{aligned} \mathrm{KL}[q(\mathbf{f},\mathbf{u})||p(\mathbf{f},\mathbf{u}|\mathbf{y})] &= \mathbb{E}_{q(\mathbf{f},\mathbf{u})}\left[\log \frac{q(\mathbf{f},\mathbf{u})}{p(\mathbf{f},\mathbf{u}|\mathbf{y})}\right] \\ &= \mathbb{E}_{q(\mathbf{f},\mathbf{u})}\left[\log \frac{\cancel{p(\mathbf{f}|\mathbf{u})}q(\mathbf{u})p(\mathbf{y})}{p(\mathbf{y}|\mathbf{f})\cancel{p(\mathbf{f}|\mathbf{u})}p(\mathbf{u})}\right] \\ &= \log p(\mathbf{y}) - \mathbb{E}_{q(\mathbf{f},\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})] + \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]. \end{aligned} \tag{4.7}$$

The main computational benefit of SVGP derives from the fact that most expensive term – the prior conditional $p(\mathbf{f}|\mathbf{u})$ involving an $\mathcal{O}(N^3)$ matrix inversion – cancels out. Rearranging for the (log) marginal likelihood, we arrive at the following lower bound:

$$\log p(\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{f},\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \mathcal{L}. \tag{4.8}$$

Writing the expected log likelihood term in integral form and factorising the approximate posterior $q(\mathbf{f}, \mathbf{u})$, we see that $\mathbf{u}$ can be integrated out:

$$\begin{aligned} \mathbb{E}_{q(\mathbf{f},\mathbf{u})}[\log p(\mathbf{y}|\mathbf{f})] &= \iint p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} d\mathbf{u} \\ &= \int \left\{ \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u}) d\mathbf{u} \right\} \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\ &= \int q(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\ &= \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})]. \end{aligned} \tag{4.9}$$

The lower bound can therefore be expressed equivalently as:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]. \tag{4.10}$$

If we factorise the likelihood, the expected log likelihood term can be expressed as a sum:

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(f_n)}[\log p(y_n|f_n)] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]. \tag{4.11}$$

The above expression of the lower bound works with general likelihoods and is conducive to stochastic optimisation using minibatches by subsampling over the N datapoints [6], making it suitable for classification tasks and large datasets. If the expected log likelihood term is analytically intractable, it may be approximated using quadrature or Monte Carlo integration. The lower bound is a function of the inducing input locations $\mathbf{Z}$ and the variational distribution $q(\mathbf{u})$, i.e. $\mathcal{L} = \mathcal{L}(\mathbf{Z}, q(\mathbf{u}))$[1].

4.3 Collapsed Bound for Regression

For regression (i.i.d. Gaussian likelihood), Titsias showed that we can derive the optimal form of the variational distribution $q(\mathbf{u})$, resulting in a tighter "collapsed" formulation of the lower bound [5]. While Titsias used calculus of variations to show that the optimal $q(\mathbf{u})$ is Gaussian, we will assume a Gaussian and use gradient methods to find closed-form expressions of the optimal parameters.

4.3.1 Finding the Optimal Variational Parameters

To begin, we find the analytic form of the expected log likelihood term in eq. 4.10:

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] &= \mathbb{E}_{q(\mathbf{f})}[\log \mathcal{N}(\mathbf{y}; \mathbf{f}, \sigma^2\mathbf{I_N})] \\
&= \mathbb{E}_{q(\mathbf{f})}\left[-\frac{1}{2}\left(\log|\sigma^2\mathbf{I_N}| + (\mathbf{y}-\mathbf{f})^\top(\sigma^2\mathbf{I_N})^{-1}(\mathbf{y}-\mathbf{f}) + N\log 2\pi\right)\right] \\
&= \mathbb{E}_{q(\mathbf{f})}\left[-\frac{N}{2}\log 2\pi\sigma^2 - \frac{1}{2\sigma^2}(\mathbf{y}^\top\mathbf{y} - 2\mathbf{y}^\top\mathbf{f} + \mathbf{f}^\top\mathbf{f})\right] \\
&= -\frac{1}{2\sigma^2}\,\mathbb{E}_{q(\mathbf{f})}\left[\mathrm{Tr}(\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\mathbf{f}^\top + \mathbf{f}\mathbf{f}^\top)\right] - \frac{N}{2}\log 2\pi\sigma^2 \\
&= -\frac{1}{2\sigma^2}\,\mathrm{Tr}\left(\mathbb{E}_{q(\mathbf{f})}\left[\mathbf{y}\mathbf{y}^\top\right] - 2\,\mathbb{E}_{q(\mathbf{f})}\left[\mathbf{y}\mathbf{f}^\top\right] + \mathbb{E}_{q(\mathbf{f})}\left[\mathbf{f}\mathbf{f}^\top\right]\right) - \frac{N}{2}\log 2\pi\sigma^2.
\end{aligned} \tag{4.12}$$

From the definition of covariance, we have:

$$\mathbb{C}_{p(\mathbf{x})}[\mathbf{x},\mathbf{x}] = \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^\top] - \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]\,\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]^\top \Rightarrow \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}\mathbf{x}^\top] = \mathbb{C}_{p(\mathbf{x})}[\mathbf{x},\mathbf{x}] + \mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]\,\mathbb{E}_{p(\mathbf{x})}[\mathbf{x}]^\top. \tag{4.13}$$

Using this rearranged covariance formula, we can compute the expectations in eq. 4.12:

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{f})}[\mathbf{y}\mathbf{y}^\top] &= \mathbb{C}_{q(\mathbf{f})}[\mathbf{y},\mathbf{y}] + \mathbb{E}_{q(\mathbf{f})}[\mathbf{y}]\,\mathbb{E}_{q(\mathbf{f})}[\mathbf{y}]^\top = \mathbf{0} + \mathbf{y}\mathbf{y}^\top, \\
\mathbb{E}_{q(\mathbf{f})}[\mathbf{y}\mathbf{f}^\top] &= \mathbb{C}_{q(\mathbf{f})}[\mathbf{y},\mathbf{f}] + \mathbb{E}_{q(\mathbf{f})}[\mathbf{y}]\,\mathbb{E}_{q(\mathbf{f})}[\mathbf{f}]^\top = \mathbf{0} + \mathbf{y}\boldsymbol{\mu}^\top, \\
\mathbb{E}_{q(\mathbf{f})}[\mathbf{f}\mathbf{f}^\top] &= \mathbb{C}_{q(\mathbf{f})}[\mathbf{f},\mathbf{f}] + \mathbb{E}_{q(\mathbf{f})}[\mathbf{f}]\,\mathbb{E}_{q(\mathbf{f})}[\mathbf{f}]^\top = \boldsymbol{\Sigma} + \boldsymbol{\mu}\boldsymbol{\mu}^\top.
\end{aligned} \tag{4.14}$$

[1] To be completely rigorous, it also depends on the kernel hyperparameters $\boldsymbol{\theta}$ and in practice we optimise with respect to these as well, but we will drop them for notational simplicity.

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the posterior mean and covariance in eq. 4.4, i.e. $q(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Substituting these results back into eq. 4.12, we obtain a Gaussian log density minus a trace term:

$$\begin{aligned}\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] &= -\frac{1}{2\sigma^2}\operatorname{Tr}\left(\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\boldsymbol{\mu}^\top + \boldsymbol{\mu}\boldsymbol{\mu}^\top + \boldsymbol{\Sigma}\right) - \frac{N}{2}\log 2\pi\sigma^2 \\ &= \log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \sigma^2\mathbf{I_N}) - \frac{1}{2\sigma^2}\operatorname{Tr}(\boldsymbol{\Sigma}).\end{aligned} \tag{4.15}$$

We can generalise this result for all Gaussian i.i.d. expected log likelihoods:

$$\mathbb{E}_{\mathcal{N}(\mathbf{f};\boldsymbol{\alpha},\boldsymbol{\beta})}\left[\log \mathcal{N}\big(\mathbf{y}; \mathbf{f}, \sigma^2\mathbf{I_N}\big)\right] = \log \mathcal{N}\big(\mathbf{y}; \boldsymbol{\alpha}, \sigma^2\mathbf{I_N}\big) - \frac{1}{2\sigma^2}\operatorname{Tr}\left(\boldsymbol{\beta}\right). \tag{4.16}$$

This is a very useful result, and one which we will repeat this several times in this thesis. Writing out the full expressions for the posterior mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ and expanding the log density term in eq. 4.15, we get the complete analytic form of the expected log likelihood term:

$$\begin{aligned}\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] = &-\frac{1}{2\sigma^2}\operatorname{Tr}(\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}\mathbf{m}_\mathbf{u}^\top\mathbf{K}_\mathbf{uu}^{-1}\mathbf{K}_\mathbf{uf} + \mathbf{K}_\mathbf{fu}\mathbf{K}_\mathbf{uu}^{-1}\mathbf{m}_\mathbf{u}\mathbf{m}_\mathbf{u}^\top\mathbf{K}_\mathbf{uu}^{-1}\mathbf{K}_\mathbf{uf}) \\ &-\frac{1}{2\sigma^2}\operatorname{Tr}(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}) - \frac{N}{2}\log 2\pi\sigma^2.\end{aligned} \tag{4.17}$$

The KL term in eq. 4.10 has an analytic form as both the variational distribution over inducing outputs $q(\mathbf{u})$ and the prior $p(\mathbf{u})$ are Gaussian:

$$\begin{aligned}\mathrm{KL}\left[q(\mathbf{u})||p(\mathbf{u})\right] &= \mathrm{KL}\left[\mathcal{N}(\mathbf{u}; \mathbf{m_u}, \mathbf{S_u})||\mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{K_{uu}})\right] \\ &= \frac{1}{2}\left\{\operatorname{Tr}\left(\mathbf{K_{uu}^{-1}}\mathbf{S_u}\right) + \mathbf{m}_\mathbf{u}^\top\mathbf{K_{uu}^{-1}}\mathbf{m_u} - M + \log\frac{|\mathbf{K_{uu}}|}{|\mathbf{S_u}|}\right\}.\end{aligned} \tag{4.18}$$

We can now compute the gradients of the lower bound (eq. 4.15 less eq. 4.18) with respect to the variational parameters (see [20] for a thorough overview of vector and matrix differentiation rules, specifically traces and determinants):

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{m_u}} &= \sigma^{-2}\left(\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} - \mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{y}\right) - \mathbf{K_{uu}^{-1}}\mathbf{m_u}, \\ \frac{\partial\mathcal{L}}{\partial\mathbf{S_u}} &= \sigma^{-2}\left(\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\right) + \mathbf{K_{uu}^{-1}} - \mathbf{S_u^{-1}}.\end{aligned} \tag{4.19}$$

Solving for zero gradients and rearranging terms, we arrive at the following closed-form expressions for the optimal variational parameters (which may be substituted into eq. 4.4 to make predictions):

$$\begin{aligned}\mathbf{m}_\mathbf{u}^* &= \sigma^{-2}\mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uf}}\mathbf{y}, \\ \mathbf{S}_\mathbf{u}^* &= \mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uu}}.\end{aligned} \tag{4.20}$$

4.3.2 Collapsing the Lower Bound

Substituting the optimal variational parameters back into lower bound in eq. 4.10 results in the *collapsed bound* for regression. Doing so is, however, algebraically tedious. Fortunately, there is a more elegant way of deriving the collapsed using a trick shown by Tisias in [5] and [17]. To begin, we note that eq. 4.8 may be expressed equivalently in integral form as:

$$\mathcal{L} = \int q(\mathbf{u}) \left\{ \int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right\} d\mathbf{u}. \tag{4.21}$$

The inner integral with respect to the function values $\mathbf{f}$ can be evaluated using eq. 4.16, resulting in a Gaussian log density minus a trace term:

$$\int p(\mathbf{f}|\mathbf{u}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} = \log \mathcal{N}(\mathbf{y}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \sigma^2\mathbf{I_N}) - \frac{1}{2\sigma^2} \operatorname{Tr}\left(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\right). \tag{4.22}$$

Substituting the above back into eq. 4.21, we obtain:

$$\mathcal{L} = \int q(\mathbf{u}) \log \left[\frac{\mathcal{N}(\mathbf{y}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \sigma^2\mathbf{I_N}) p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u} - \frac{1}{2\sigma^2} \operatorname{Tr}\left(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\right). \tag{4.23}$$

Before proceeding, we recall *Jensen's inequality* for concave functions.

Jensen's inequality: For a concave function f and a random variable x, we have

$$f(\mathbb{E}[x]) \geq \mathbb{E}[f(x)] \Rightarrow \mathbb{E}[f(x)] \leq f(\mathbb{E}[x]). \tag{4.24}$$

We now have a lower bound on the log marginal likelihood comprised of an expectation term and a trace term. Let us call the expectation term A and the trace term B, i.e. we have $\log p(\mathbf{y}) \geq \mathcal{L} = A - B$. Now for the trick. If we reverse Jensen's inequality on the expectation term (A), we can obtain an upper bound on the lower bound. In this case, if $\log p(\mathbf{y}) \geq A - B$ and $A \leq C$, then $A - B \leq C - B$ for $B \geq 0$. Reversing Jensen's inequality on the expectation term involves pulling the logarithm outside of the integral, allowing the variational distribution $q(\mathbf{u})$ to cancel out. The resulting integration can be done using Gaussian linear transformation rules:

$$\begin{aligned} \int q(\mathbf{u}) \log \left[\frac{\mathcal{N}(\mathbf{y}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \sigma^2\mathbf{I_N}) p(\mathbf{u})}{q(\mathbf{u})} \right] d\mathbf{u} &\leq \log \int \cancel{q(\mathbf{u})} \frac{\mathcal{N}(\mathbf{y}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \sigma^2\mathbf{I_N}) p(\mathbf{u})}{\cancel{q(\mathbf{u})}} d\mathbf{u} \\ &= \log \int \mathcal{N}(\mathbf{y}; \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \sigma^2\mathbf{I_N}) \mathcal{N}(\mathbf{u}; \mathbf{0}, \mathbf{K_{uu}}) d\mathbf{u} \\ &= \log \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \sigma^2\mathbf{I_N}). \end{aligned} \tag{4.25}$$

Subtracting the trace term from eq. 4.23, we arrive at the collapsed bound for regression:

$$\mathcal{L}^* = \log \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \sigma^2\mathbf{I_N}) - \frac{1}{2\sigma^2} \operatorname{Tr}(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}). \tag{4.26}$$

This result is identical to what would have been obtained via substitution of the optimal variational parameters in eq. 4.20 into 4.10. The collapsed bound is only a function of the inducing input locations $\mathbf{Z}$, i.e. $\mathcal{L}^* = \mathcal{L}^*(\mathbf{Z})$, as the optimal variational distribution is given in closed-form. As shown in eq. 4.25, the collapsed bound is an upper bound on all lower bounds and is therefore guaranteed to be equivalent to or tighter than the uncollapsed bound, i.e. $\mathcal{L}^* \geq \mathcal{L}$ [5]. The collapsed bound is not, however, conducive to stochastic optimisation using minibatches as there is no summation over which to subsample, making it most suitable to mid-size datasets. Collapsed bound SVGP regression is often referred to as *Sparse Gaussian Process Regression* (SGPR). We will focus on collapsed bound regression in this thesis, as it serves as a good basis of comparison between sparse variational approximations and removes some of the potential variability that can arise due to optimisation issues.

4.4 Stable and Efficient Implementation

Now that we have derived the collapsed bound and the optimal variational parameters for regression, we can come up with a numerically stable and efficient SGPR implementation. Using Cholesky decompositions, Hensman and Matthews showed that we can derive expressions of the lower bound and predictive equations which are better conditioned for inversion and less likely to result in optimisation issues [18]. We will go through their derivation and show a practical algorithmic implementation.

4.4.1 Stable and Efficient Lower Bound

If we expand the log density term in eq. 4.26, we get:

$$-\frac{1}{2}\left(\log|\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}+\sigma^2\mathbf{I_N}|+\mathbf{y}^\top\left[\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}+\sigma^2\mathbf{I_N}\right]^{-1}\mathbf{y}+N\log 2\pi\right). \tag{4.27}$$

The most expensive terms to compute are the log determinant and inverse. Fortunately, we can reduce the computational cost by applying *matrix inversion lemma* and *matrix determinant lemma*.

Matrix inversion lemma: If $\mathbf{A}$ is an invertible $N \times N$ matrix, $\mathbf{C}$ is an invertible $M \times M$ matrix, $\mathbf{U}$ is an $N \times M$ matrix and $\mathbf{V}$ is a $M \times N$ matrix, then

$$[\mathbf{A}+\mathbf{UCV}]^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{VA}^{-1}\mathbf{U}\right)^{-1}\mathbf{VA}^{-1}. \tag{4.28}$$

Matrix determinant lemma: If $\mathbf{A}$ is an invertible $N \times N$ matrix, $\mathbf{W}$ is an invertible $M \times M$ matix and $\mathbf{U}$, $\mathbf{V}$ are $N \times M$ matrices, then

$$|\mathbf{A}+\mathbf{UWV}^\top| = |\mathbf{W}^{-1}+\mathbf{V}^\top\mathbf{A}^{-1}\mathbf{U}||\mathbf{W}||\mathbf{A}|. \tag{4.29}$$

We apply matrix inversion lemma to the inverse term in eq. 4.27:

$$\left[\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}+\sigma^2\mathbf{I_N}\right]^{-1} = \sigma^{-2}\mathbf{I_N} - \sigma^{-4}\mathbf{K_{fu}}\left[\mathbf{K_{uu}}+\sigma^{-2}\mathbf{K_{uf}K_{fu}}\right]^{-1}\mathbf{K_{uf}}. \tag{4.30}$$

Next, we rotate by the Cholesky decomposition $\mathbf{L}$ of $\mathbf{K_{uu}}$, i.e. $\mathbf{K_{uu}} = \mathbf{LL}^\top$, to obtain a better condi-

tioned matrix for inversion. This puts stable upper and lower bounds on the eigenvalues [15]:

$$\begin{aligned} &= \sigma^{-2}\mathbf{I_N} - \sigma^{-4}\mathbf{K_{fu}}\mathbf{L}^{-\top}\mathbf{L}^{\top}\left[\mathbf{L}\mathbf{L}^{\top} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{L}\mathbf{L}^{-1}\mathbf{K_{uf}} \\ &= \sigma^{-2}\mathbf{I_N} - \sigma^{-4}\mathbf{K_{fu}}\mathbf{L}^{-\top}\left[\mathbf{I_M} + \sigma^{-2}\mathbf{L}^{-1}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{L}^{-\top}\right]^{-1}\mathbf{L}^{-1}\mathbf{K_{uf}}. \end{aligned} \tag{4.31}$$

To simplify our expression, we let $\mathbf{A} = \sigma^{-1}\mathbf{L}^{-1}\mathbf{K_{uf}}$ and $\mathbf{B} = \mathbf{I_M} + \mathbf{A}\mathbf{A}^{\top}$ where $\mathbf{B} = \mathbf{L_B}\mathbf{L_B^{\top}}$:

$$\begin{aligned} &= \sigma^{-2}\mathbf{I_N} - \sigma^{-2}\mathbf{A}^{\top}\left[\mathbf{I_M} + \mathbf{A}\mathbf{A}^{\top}\right]^{-1}\mathbf{A} \\ &= \sigma^{-2}\left(\mathbf{I_N} - \mathbf{A}^{\top}\mathbf{B}^{-1}\mathbf{A}\right) \\ &= \sigma^{-2}\left(\mathbf{I_N} - \mathbf{A}^{\top}\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A}\right). \end{aligned} \tag{4.32}$$

Next, we apply matrix determinant lemma to the log determinant term in eq. 4.27:

$$\log|\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \sigma^2\mathbf{I_N}| = \log(|\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}||\mathbf{K_{uu}}^{-1}||\sigma^2\mathbf{I_N}|). \tag{4.33}$$

Substituting in $\mathbf{K_{uu}} = \mathbf{L}\mathbf{L}^{\top}$:

$$\begin{aligned} &= \log(|\mathbf{L}\mathbf{L}^{\top} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}||\mathbf{L}^{-\top}||\mathbf{L}^{-1}||\sigma^2\mathbf{I_N}|) \\ &= \log(|\mathbf{I_M} + \sigma^{-2}\mathbf{L}^{-1}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{L}^{-\top}||\sigma^2\mathbf{I_N}|). \end{aligned} \tag{4.34}$$

Using our intermediate matrix definitions from before:

$$\begin{aligned} &= \log(|\mathbf{I_M} + \mathbf{A}\mathbf{A}^{\top}||\sigma^2\mathbf{I_N}|) \\ &= \log(|\mathbf{B}||\sigma^2\mathbf{I_N}|) \\ &= \log(|\mathbf{L_B}||\mathbf{L_B^{\top}}||\sigma^2\mathbf{I_N}|). \end{aligned} \tag{4.35}$$

We recall that the determinant of a triangular matrix is simply the product of the diagonals:

$$\begin{aligned} &= \log\Big(\Big[\prod_i \mathrm{L_{B_{ii}}}\Big]\Big[\prod_i \mathrm{L_{B_{ii}}}\Big]\sigma^{2N}\Big) \\ &= 2\sum_i \log \mathrm{L_{B_{ii}}} + N\log\sigma^2. \end{aligned} \tag{4.36}$$

Finally, we let $\mathbf{c} = \sigma^{-1}\mathbf{L_B}^{-1}\mathbf{A}\mathbf{y}$. The collapsed bound in eq. 4.26 becomes:

$$\begin{aligned}\mathcal{L}^* &= -\frac{1}{2}\left(2\sum_i \log \mathrm{L}_{\mathrm{B}_{\mathrm{ii}}} + N\log\sigma^2 + \mathbf{y}^\top\sigma^{-2}\left(\mathbf{I_N} - \mathbf{A}^\top\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A}\right)\mathbf{y} + N\log 2\pi\right) - \frac{1}{2\sigma^2}\mathrm{Tr}(\mathbf{K_{ff}} - \sigma^2\mathbf{A}\mathbf{A}^\top) \\ &= -\frac{N}{2}\log 2\pi\sigma^2 - \sum_i \log \mathrm{L}_{\mathrm{B}_{\mathrm{ii}}} - \frac{1}{2}\sigma^{-2}\mathbf{y}^\top\mathbf{y} + \frac{1}{2}\sigma^{-2}\mathbf{y}^\top\mathbf{A}^\top\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A}\mathbf{y} - \frac{1}{2}\sigma^{-2}\,\mathrm{Tr}(\mathbf{K_{ff}}) + \frac{1}{2}\,\mathrm{Tr}(\mathbf{A}\mathbf{A}^\top) \\ &= -\frac{N}{2}\log 2\pi\sigma^2 - \sum_i \log \mathrm{L}_{\mathrm{B}_{\mathrm{ii}}} - \frac{1}{2}\sigma^{-2}\mathbf{y}^\top\mathbf{y} + \frac{1}{2}\mathbf{c}^\top\mathbf{c} - \frac{1}{2}\sigma^{-2}\,\mathrm{Tr}(\mathbf{K_{ff}}) + \frac{1}{2}\,\mathrm{Tr}(\mathbf{A}\mathbf{A}^\top). \end{aligned} \tag{4.37}$$

4.4.2 Stable and Efficient Predictive Equations

Grouping and rearranging terms in eq. 4.20, we can arrive at the following equivalent expressions for the optimal variational parameters:

$$\begin{aligned}\mathbf{m_u^*} &= \sigma^{-2}\left[\mathbf{K_{uu}^{-1}} + \sigma^{-2}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\right]^{-1}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{y}, \\ \mathbf{S_u^*} &= \left[\mathbf{K_{uu}^{-1}} + \sigma^{-2}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\right]^{-1}.\end{aligned} \tag{4.38}$$

Substituting the optimal variational mean in eq. 4.38 into the posterior in eq. 4.4, we get the following expression for the posterior mean $\boldsymbol{\mu_*} = \mu(\mathbf{X_*})$ at some test points $\mathbf{X_*}$:

$$\begin{aligned}\boldsymbol{\mu_*} &= \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} \\ &= \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\left(\sigma^{-2}\left[\mathbf{K_{uu}^{-1}} + \sigma^{-2}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\right]^{-1}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{y}\right).\end{aligned} \tag{4.39}$$

Substituting $\mathbf{K_{uu}} = \mathbf{L}\mathbf{L}^\top$:

$$\begin{aligned}&= \mathbf{K_{*u}}\mathbf{L}^{-\top}\mathbf{L}^{-1}\left(\sigma^{-2}\left[\mathbf{L}^{-\top}\mathbf{L}^{-1} + \sigma^{-2}\mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{L}^{-\top}\mathbf{L}^{-1}\right]^{-1}\mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{K_{uf}}\mathbf{y}\right) \\ &= \mathbf{K_{*u}}\mathbf{L}^{-\top}\left(\sigma^{-2}\left[\mathbf{I_M} + \sigma^{-2}\mathbf{L}^{-1}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{L}^{-\top}\right]^{-1}\mathbf{L}^{-1}\mathbf{K_{uf}}\mathbf{y}\right).\end{aligned} \tag{4.40}$$

Using our intermediate matrix definitions from before:

$$\begin{aligned}&= \mathbf{K_{*u}}\mathbf{L}^{-\top}\left(\sigma^{-1}\left[\mathbf{I_M} + \mathbf{A}\mathbf{A}^\top\right]^{-1}\mathbf{A}\mathbf{y}\right) \\ &= \mathbf{K_{*u}}\mathbf{L}^{-\top}\left(\sigma^{-1}\mathbf{B}^{-1}\mathbf{A}\mathbf{y}\right) \\ &= \mathbf{K_{*u}}\mathbf{L}^{-\top}\left(\sigma^{-1}\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A}\mathbf{y}\right) \\ &= \mathbf{K_{*u}}\mathbf{L}^{-\top}\mathbf{L_B^{-\top}}\mathbf{c}.\end{aligned} \tag{4.41}$$

Similarly, to obtain the posterior covariance $\boldsymbol{\Sigma_{**}} = \Sigma(\mathbf{X_*}, \mathbf{X_*})$ we substitute the optimal variational covariance in eq. 4.38 into eq. 4.4 (note that if we wish to predict some new observations $\mathbf{y_*}$ rather than function values $\mathbf{f_*}$, we need to add the likelihood variance σ^2 to the diagonals):

$$\begin{aligned}\boldsymbol{\Sigma}_{**} &= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}(\mathbf{K}_{\mathbf{uu}} - \mathbf{S}_{\mathbf{u}})\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}\mathbf{K}_{\mathbf{u}*} \\ &= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}\left(\mathbf{K}_{\mathbf{uu}} - \left[\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}} + \sigma^{-2}\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}\right]^{-1}\right)\mathbf{K}_{\mathbf{uu}}^{-\mathbf{1}}\mathbf{K}_{\mathbf{u}*}.\end{aligned} \tag{4.42}$$

Substituting $\mathbf{K_{uu}} = \mathbf{L}\mathbf{L}^\top$:

$$\begin{aligned}&= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{L}^{-\top}\mathbf{L}^{-1}\Big(\mathbf{L}\mathbf{L}^\top - \Big[\mathbf{L}^{-\top}\mathbf{L}^{-1} + \sigma^{-2}\mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}\mathbf{L}^{-\top}\mathbf{L}^{-1}\Big]^{-1}\Big)\mathbf{L}^{-\top}\mathbf{L}^{-1}\mathbf{K}_{\mathbf{u}*} \\ &= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{L}^{-\top}\Big(\mathbf{I}_{\mathbf{M}} - \Big[\mathbf{I}_{\mathbf{M}} + \sigma^{-2}\mathbf{L}^{-1}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}\mathbf{L}^{-\top}\Big]^{-1}\Big)\mathbf{L}^{-1}\mathbf{K}_{\mathbf{u}*}.\end{aligned} \tag{4.43}$$

Using our intermediate matrix definitions from before:

$$\begin{aligned}&= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{L}^{-\top}\Big(\mathbf{I}_{\mathbf{M}} - \Big[\mathbf{I}_{\mathbf{M}} + \mathbf{A}\mathbf{A}^\top\Big]^{-1}\Big)\mathbf{L}^{-1}\mathbf{K}_{\mathbf{u}*} \\ &= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{L}^{-\top}\Big(\mathbf{I}_{\mathbf{M}} - \mathbf{B}^{-1}\Big)\mathbf{L}^{-1}\mathbf{K}_{\mathbf{u}*} \\ &= \mathbf{K}_{**} - \mathbf{K}_{*\mathbf{u}}\mathbf{L}^{-\top}\Big(\mathbf{I}_{\mathbf{M}} - \mathbf{L}_{\mathbf{B}}^{-\top}\mathbf{L}_{\mathbf{B}}^{-\mathbf{1}}\Big)\mathbf{L}^{-1}\mathbf{K}_{\mathbf{u}*}.\end{aligned} \tag{4.44}$$

4.4.3 SGPR Algorithm

A stable and efficient implementation of the SGPR lower bound and predictive equations is shown in algorithm 2. Note that in practice it is often useful to add "jitter" (a small float value, e.g. 10^{-6}) to the diagonals of $\mathbf{K_{uu}}$ and $\mathbf{C_{vv}}$ to make Cholesky decomposition and follow up operations more numerically stable.

Algorithm 2 ELBO and predictions for sparse Gaussian process regression via Cholesky decomposition. $\mathbf{A} = \mathbf{B}\backslash\mathbf{C}$ denotes the solution to $\mathbf{BA} = \mathbf{C}$.

Input: $\mathbf{X}$ (training inputs), $\mathbf{y}$ (targets), $\mathbf{Z}$ (inducing inputs), k (kernel), σ^2 (noise), $\mathbf{X}_*$ (test inputs)

$\mathbf{K_{ff}} = k(\mathbf{X}, \mathbf{X})$, $\mathbf{K_{uu}} = k(\mathbf{Z}, \mathbf{Z})$, $\mathbf{K_{uf}} = k(\mathbf{Z}, \mathbf{X})$
$\mathbf{L} = \text{Cholesky}(\mathbf{K_{uu}})$
$\mathbf{A} = \sigma^{-1}\mathbf{L}\backslash\mathbf{K_{uf}}$
$\mathbf{AAT} = \mathbf{A}\mathbf{A}^\top$
$\mathbf{B} = \mathbf{I_M} + \mathbf{AAT}$
$\mathbf{L_B} = \text{Cholesky}(\mathbf{B})$
$\mathbf{c} = \sigma^{-1}\mathbf{L_B}\backslash(\mathbf{A}\mathbf{y})$
$\mathcal{L}^* = -\frac{N}{2}\log 2\pi\sigma^2 - \sum_i \log \mathrm{L_{B_{ii}}} - \frac{1}{2}\sigma^{-2}\mathbf{y}^\top\mathbf{y} + \frac{1}{2}\mathbf{c}^\top\mathbf{c} - \frac{1}{2}\sigma^{-2}\,\mathrm{Tr}(\mathbf{K_{ff}}) + \frac{1}{2}\,\mathrm{Tr}(\mathbf{AAT})$
$\mathbf{K}_{**} = k(\mathbf{X}_*, \mathbf{X}_*)$, $\mathbf{K_{u*}} = k(\mathbf{Z}, \mathbf{X}_*)$
$\boldsymbol{\alpha} = \mathbf{L}\backslash\mathbf{K_{u*}}$
$\boldsymbol{\beta} = \mathbf{L_B}\backslash\boldsymbol{\alpha}$
$\boldsymbol{\mu}_* = \boldsymbol{\beta}^\top\mathbf{c}$
$\boldsymbol{\Sigma}_{**} = \mathbf{K}_{**} - \boldsymbol{\alpha}^\top\boldsymbol{\alpha} + \boldsymbol{\beta}^\top\boldsymbol{\beta}$
return: $\mathcal{L}^*$ (evidence lower bound), $\boldsymbol{\mu}_*$ (mean), $\boldsymbol{\Sigma}_{**}$ (covariance)

Chapter 5

Orthogonal Sparse Variational Gaussian Processes

While the original sparse variational Gaussian process model as proposed by Titsias [5] can offer significant computational cost reduction over exact GP inference (at the expense of reduced accuracy in modelling the posterior), it doesn't fully resolve computational bottlenecks in tasks involving high dimensional input spaces (e.g. images) or complex datasets which may not be able to be effectively summarised with a small number of inducing points.

In eq. 4.6, we showed that the SVGP posterior could be decomposed as a sum of two independent constituent Gaussian processes. Recently, "orthogonal" variations of the original SVGP model have been proposed based on *further* decomposition of the posterior through the introduction of a second set of M_2 inducing points $\mathbf{v} = f(\mathbf{O})$, called *orthogonal* inducing points. These orthogonal SVGP variations are the primary focus of this thesis; they can be represented by a structured covariance between the two sets of inducing points $\mathbf{u}$ and $\mathbf{v}$, or by orthogonal decomposition of the Gaussian process in the *reproducing kernel Hilbert space* (RKHS) induced by the kernel. While the latter representation makes it more apparent why these formulations are called "orthogonal", we will focus on the former as it is an intuitive unifying viewpoint from which to compare different SVGP parameterisations. The key to understanding the connection between these two representations is that orthogonality in a reproducing kernel Hilbert space equates to statistical independence in function space.

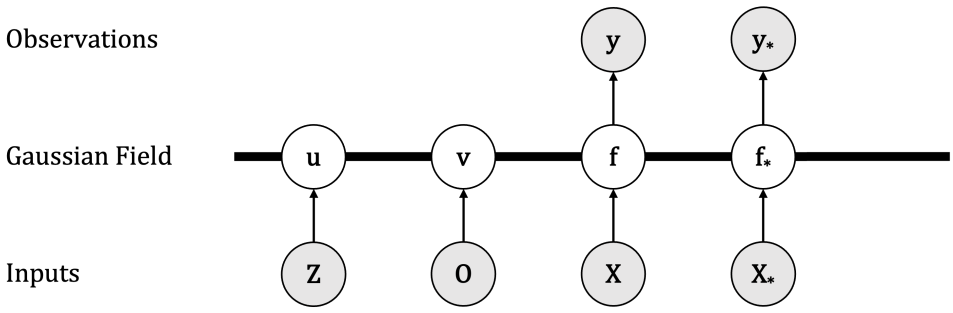

Figure 5.1: Directed graphical model for an orthogonal sparse variational Gaussian process using "thick bar notation" to denote a Gaussian field. Observable variables are shaded in grey; unobservable variables are left unshaded.

Two notable orthogonal SVGP parameterisations are "Orthogonally Decoupled Variational Gaussian Processes" (ODVGP) [9] and "Sparse OrthogonaL Variational infErence for Gaussian Processes" (SOLVE-GP) [10]. The authors of both models argue that their approach reduces the computational cost per inducing point, thus more inducing points can be used for a fixed computational budget, resulting in superior flexibility in modelling the posterior as compared with standard SVGP. In this chapter, we will summarise the ODVGP and SOLVE-GP parameterisations before making our first novel contributions: deriving an orthogonal collapsed bound and a stable and efficient implementation of ODVGP and SOLVE-GP for regression, taking inspiration from the work of Titsias [17] and Hensman/Matthews [18] shown in the previous chapter.

5.1 Model Formulation and Approximate Posterior

To formulate an orthogonal sparse variational Gaussian process, we introduce a set of M_2 orthogonal inducing points $\mathbf{v} = f(\mathbf{O})$ to the SVGP joint probability model (in addition to the M_1 regular inducing points $\mathbf{u} = f(\mathbf{Z})$):

$$p(\mathbf{y}, \mathbf{f}, \mathbf{u}, \mathbf{v}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u}, \mathbf{v})p(\mathbf{u}, \mathbf{v}) = p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{u}, \mathbf{v})p(\mathbf{v}|\mathbf{u})p(\mathbf{u}). \tag{5.1}$$

Similarly to SVGP, we create an approximate (variational) posterior over the function values and inducing outputs $q(\mathbf{f}, \mathbf{u}, \mathbf{v}) \approx p(\mathbf{f}, \mathbf{u}, \mathbf{v}|\mathbf{y})$ by assuming a factorisation which retains the prior conditional $p(\mathbf{f}|\mathbf{u}, \mathbf{v})$ and puts a tractable (Gaussian) variational distribution on the inducing outputs $q(\mathbf{u}, \mathbf{v})$:

$$q(\mathbf{f}, \mathbf{u}, \mathbf{v}) = p(\mathbf{f}|\mathbf{u}, \mathbf{v})q(\mathbf{u}, \mathbf{v}) = p(\mathbf{f}|\mathbf{u}, \mathbf{v})q(\mathbf{v}|\mathbf{u})q(\mathbf{u}). \tag{5.2}$$

The prior conditional $p(\mathbf{f}|\mathbf{u}, \mathbf{v})$ is obtained through Gaussian conditioning rules on the joint prior over the function values and the inducing outputs $p(\mathbf{f}, \mathbf{u}, \mathbf{v})$:

$$\begin{aligned} p(\mathbf{f}, \mathbf{u}, \mathbf{v}) &= \mathcal{N}\left(\begin{bmatrix}\mathbf{f}\\ \mathbf{u}\\ \mathbf{v}\end{bmatrix}; \begin{bmatrix}\mathbf{0}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}, \begin{bmatrix}\mathbf{K_{ff}} & \mathbf{K_{fu}} & \mathbf{K_{fv}}\\ \mathbf{K_{uf}} & \mathbf{K_{uu}} & \mathbf{K_{uv}}\\ \mathbf{K_{vf}} & \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix}\right), \\ \Rightarrow p(\mathbf{f}|\mathbf{u}, \mathbf{v}) &= \mathcal{N}\left(\mathbf{f}; \begin{bmatrix}\mathbf{K_{fu}}\mathbf{K_{fv}}\end{bmatrix}\begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}}\\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}, \mathbf{K_{ff}} - \begin{bmatrix}\mathbf{K_{fu}}\mathbf{K_{fv}}\end{bmatrix}\begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}}\\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{K_{uf}}\\ \mathbf{K_{vf}}\end{bmatrix}\right) \end{aligned}. \tag{5.3}$$

The block matrix inversion in the above can be written explicitly [21]:

$$\begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}}\\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix}^{-1} = \begin{bmatrix}\mathbf{K_{uu}^{-1}} + \mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}\mathbf{C_{vv}^{-1}}\mathbf{K_{vu}}\mathbf{K_{uu}^{-1}} & -\mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}\mathbf{C_{vv}^{-1}}\\ -\mathbf{C_{vv}^{-1}}\mathbf{K_{vu}}\mathbf{K_{uu}^{-1}} & \mathbf{C_{vv}^{-1}}\end{bmatrix}, \tag{5.4}$$

where $\mathbf{C_{vv}}$ denotes the *Schur complement* [21] of $\mathbf{K_{vv}}$ in the above and is also the covariance of the prior conditional $p(\mathbf{v}|\mathbf{u})$. Let us define it explicitly, along with another expression which will later help simplify the form our posterior:

$$\begin{aligned} \mathbf{C_{vv}} &= \mathbf{K_{vv}} - \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}, \\ \mathbf{C_{fv}} &= \mathbf{K_{fv}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}. \end{aligned} \tag{5.5}$$

We will refer to these as orthogonal covariances; they closely approximate the true covariances $\mathbf{K_{vv}}$ and $\mathbf{K_{fv}}$ when the covariance between the two sets of inducing points $\mathbf{K_{uv}}$ is small. To simplify our notation, let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ denote the prior conditional mean and covariance in eq. 5.3, i.e. $p(\mathbf{f}|\mathbf{u},\mathbf{v}) = \mathcal{N}(\mathbf{f};\boldsymbol{\alpha},\boldsymbol{\beta})$, and let $\mathbf{m}$ and $\mathbf{S}$ denote the mean and covariance of the variational distribution, i.e. $q(\mathbf{u},\mathbf{v}) = \mathcal{N}([\mathbf{u},\mathbf{v}]^\top;\mathbf{m},\mathbf{S})$. Letting $\mathbf{A}$ represent the first two block terms in the prior conditional mean which premultiply the inducing points, we can write $\boldsymbol{\alpha} = \mathbf{A}[\mathbf{u},\mathbf{v}]^\top$, and the approximate posterior $q(\mathbf{f})$ can be obtained using Gaussian linear transformation rules:

$$\begin{aligned} q(\mathbf{f}) &= \iint q(\mathbf{f},\mathbf{u},\mathbf{v})d\mathbf{u}d\mathbf{v} = \int p(\mathbf{f}|\mathbf{u},\mathbf{v})q(\mathbf{u},\mathbf{v})d\mathbf{u}d\mathbf{v} \\ &= \iint \mathcal{N}\left(\mathbf{f};\mathbf{A}[\mathbf{u},\mathbf{v}]^\top,\boldsymbol{\beta}\right)\mathcal{N}\left([\mathbf{u},\mathbf{v}]^\top;\mathbf{m},\mathbf{S}\right)d\mathbf{u}d\mathbf{v} \\ &= \mathcal{N}\left(\mathbf{f};\mathbf{A}\mathbf{m},\mathbf{A}\mathbf{S}\mathbf{A}^\top+\boldsymbol{\beta}\right). \end{aligned} \tag{5.6}$$

ODVGP and SOLVE-GP use different parameterisations of $q(\mathbf{u},\mathbf{v})$ and thus have different posteriors.

5.1.1 ODVGP

Proposed by Salimbeni, Cheng, Boots and Deisenroth in their 2019 paper [9], Orthogonally Decoupled Variational Gaussian Processes (ODVGP) introduces an additional variational parameter $\mathbf{m_v}$ to help learn the posterior mean. ODVGP uses the following parameterisation of $q(\mathbf{u},\mathbf{v})$:

$$q(\mathbf{u},\mathbf{v}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{u}\\\mathbf{v}\end{bmatrix};\begin{bmatrix}\mathbf{m_u}\\\mathbf{m_v}+\mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u}\end{bmatrix},\begin{bmatrix}\mathbf{S_u} & \mathbf{S_u}\mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}\\ \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{S_u} & \mathbf{C_{vv}}+\mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{S_u}\mathbf{K_{uu}^{-1}}\mathbf{K_{uv}}\end{bmatrix}\right). \tag{5.7}$$

Noting the terms on the off-diagonals of the block covariance, we see that this is different from the mean field parameterisation $q(\mathbf{u},\mathbf{v}) = q(\mathbf{u})q(\mathbf{v})$; covariance between the two sets of inducing points is enforced through a structured parameterisation. Using Gaussian marginalisation rules, we can see that the marginal on $\mathbf{u}$ is the same as for SVGP, i.e. $q(\mathbf{u}) = \mathcal{N}(\mathbf{u};\mathbf{m_u},\mathbf{S_u})$. Using Gaussian conditioning rules, we obtain the following conditional $q(\mathbf{v}|\mathbf{u})$:

$$q(\mathbf{v}|\mathbf{u}) = \mathcal{N}(\mathbf{v};\mathbf{m_v}+\mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{u},\mathbf{C_{vv}}). \tag{5.8}$$

I.e. $q(\mathbf{v}|\mathbf{u}) = \mathbf{m_v} + p(\mathbf{v}|\mathbf{u})$, meaning we keep the same covariance as the prior. We obtain the posterior from eq. 5.6 by substituting in the block mean and covariance from eq. 5.7 for $\mathbf{m}$ and $\mathbf{S}$, as well as the values of $\mathbf{A}$ and $\boldsymbol{\beta}$ from eq. 5.3, then performing the block matrix multiplications using the explicit form of the inverse in eq. 5.4. Sparing the grisly algebraic details, we arrive at the following expression for the posterior:

$$q(\mathbf{f}) = \mathcal{N}(\mathbf{f};\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u}+\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{m_v},\mathbf{K_{ff}}-\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}}-\mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}). \tag{5.9}$$

This result implies the following Gaussian process:

$$f(\mathbf{x}) \sim \mathcal{GP}\Big(\mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}\mathbf{m_u}+\mathbf{c_v}(\mathbf{x})^\top\mathbf{C_{vv}^{-1}}\mathbf{m_v}, k(\mathbf{x},\mathbf{x})-\mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}}-\mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{k_u}(\mathbf{x})\Big). \tag{5.10}$$

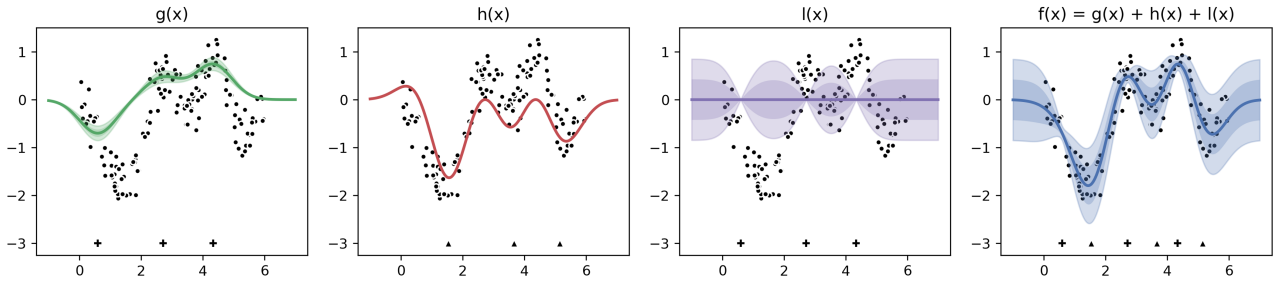

Figure 5.2: ODVGP posterior with $(M_1, M_2) = (3, 3)$ inducing points (far right) split into its constituent processes $g(\mathbf{x})$ (far left), $h(\mathbf{x})$ (centre left) and $l(\mathbf{x})$ (centre right) on Snelson and Gharahmani's dataset [4]. Inducing inputs are shown as black "plus" symbols; orthogonal inducing inputs are shown as black triangles. $g(\mathbf{x})$ and $h(\mathbf{x})$ are adjustable, while $l(\mathbf{x})$ remains unchanged after observing data. The orthogonal inducing points only contribute to the mean, thus $h(\mathbf{x})$ has no capacity to learn covariance.

Posterior Decomposition

The ODVGP posterior can be decomposed as a sum of three independent Gaussian processes:

$$\begin{aligned} f(\mathbf{x}) &= g(\mathbf{x}) + h(\mathbf{x}) + l(\mathbf{x}), \\ g(\mathbf{x}) &\sim \mathcal{GP}\left(\mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{m_u}, \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{S_u} \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x})\right), \\ h(\mathbf{x}) &\sim \mathcal{GP}\left(\mathbf{c_v}(\mathbf{x})^\top \mathbf{C_{vv}^{-1}} \mathbf{m_v}, 0\right), \\ l(\mathbf{x}) &\sim \mathcal{GP}\left(0, k(\mathbf{x}, \mathbf{x}) - \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}} \mathbf{k_u}(\mathbf{x})\right). \end{aligned} \tag{5.11}$$

This posterior decomposition is similar to eq. 4.6, with the addition of a third constituent process which uses the orthogonal inducing points $\mathbf{v}$ to help learn the posterior mean. Note that the contribution of the orthogonal inducing points to the posterior mean involves the orthogonal covariances we defined in eq. 5.5, whereas contribution of the regular inducing points involves true covariances. Also note that the orthogonal inducing points do not contribute to the posterior covariance in any way.

5.1.2 SOLVE-GP

Proposed by Shi, Titsias, and Mnih in their 2020 paper [10], Sparse OrthogonaL Variational infErence for Gaussian Processes (SOLVE-GP) extends the ODVGP parameterisation by introducing an additional variational parameter $\mathbf{S_v}$ to help learn the posterior covariance. SOLVE-GP uses the following parameterisation of the variational distribution $q(\mathbf{u}, \mathbf{v})$:

$$q(\mathbf{u}, \mathbf{v}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{u} \\ \mathbf{v}\end{bmatrix}; \begin{bmatrix}\mathbf{m_u} \\ \mathbf{m_v} + \mathbf{K_{vu}} \mathbf{K_{uu}^{-1}} \mathbf{m_u}\end{bmatrix}, \begin{bmatrix}\mathbf{S_u} & \mathbf{S_u} \mathbf{K_{uu}^{-1}} \mathbf{K_{uv}} \\ \mathbf{K_{vu}} \mathbf{K_{uu}^{-1}} \mathbf{S_u} & \mathbf{S_v} + \mathbf{K_{vu}} \mathbf{K_{uu}^{-1}} \mathbf{S_u} \mathbf{K_{uu}^{-1}} \mathbf{K_{uv}}\end{bmatrix}\right). \tag{5.12}$$

SOLVE-GP's parameterisation implies the same marginal on $\mathbf{u}$ as ODVGP and SVGP, i.e. $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m_u}, \mathbf{S_u})$, but a different conditional $q(\mathbf{v}|\mathbf{u})$:

$$q(\mathbf{v}|\mathbf{u}) = \mathcal{N}(\mathbf{v}; \mathbf{m_v} + \mathbf{K_{vu}} \mathbf{K_{uu}^{-1}} \mathbf{u}, \mathbf{S_v}). \tag{5.13}$$

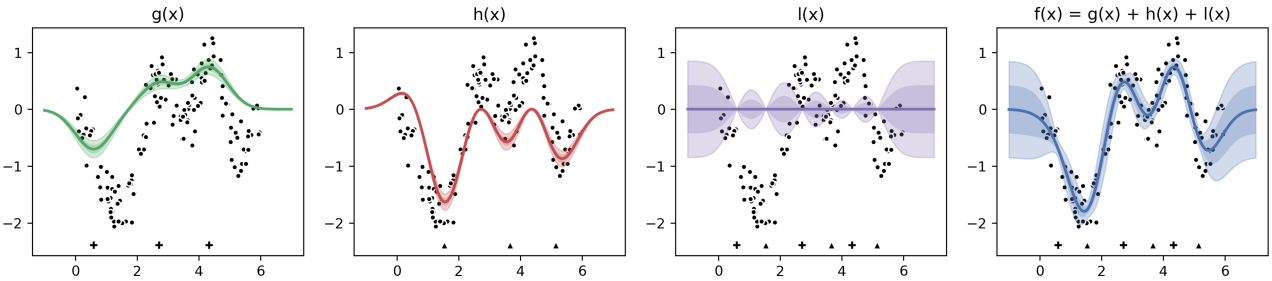

Figure 5.3: SOLVE-GP posterior with $(M_1, M_2) = (3, 3)$ inducing points (far right) split into its constituent processes $g(\mathbf{x})$ (far left), $h(\mathbf{x})$ (centre left) and $l(\mathbf{x})$ (centre right) on Snelson and Gharahmani's dataset [4]. Inducing inputs are shown as black "plus" symbols; orthogonal inducing inputs are shown as black triangles. $g(\mathbf{x})$ and $h(\mathbf{x})$ are adjustable, while $l(\mathbf{x})$ remains unchanged after observing data. The orthogonal inducing points contribute to both the mean and covariance.

If we restrict $\mathbf{S_v}$ to equal the covariance of the prior conditional $p(\mathbf{v}|\mathbf{u})$, i.e. $\mathbf{S_v} = \mathbf{C_{vv}}$, SOLVE-GP becomes equivalent to ODVGP; thus ODVGP can be viewed as a special case of SOLVE-GP. We again obtain the posterior from eq. 5.6 by substituting in the block mean and covariance from eq. 5.12 for $\mathbf{m}$ and $\mathbf{S}$, as well as the values of $\mathbf{A}$ and $\boldsymbol{\beta}$ from eq. 5.3, then performing the block matrix multiplications using the explicit form of the inverse in eq. 5.4, arriving at the following expression:

$$\begin{aligned} q(\mathbf{f}) = \mathcal{N}(\mathbf{f}; &\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} + \mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{m_v}, \\ &\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} - \mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}(\mathbf{C_{vv}} - \mathbf{S_v})\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}). \end{aligned} \tag{5.14}$$

This result implies the following Gaussian process:

$$\begin{aligned} f(\mathbf{x}) \sim \mathcal{GP}\Big(&\mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}\mathbf{m_u} + \mathbf{c_v}(\mathbf{x})^\top\mathbf{C_{vv}^{-1}}\mathbf{m_v}, \\ &k(\mathbf{x},\mathbf{x}) - \mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{k_u}(\mathbf{x}) - \mathbf{c_v}(\mathbf{x})^\top\mathbf{C_{vv}^{-1}}(\mathbf{C_{vv}} - \mathbf{S_v})\mathbf{C_{vv}^{-1}}\mathbf{c_v}(\mathbf{x})\Big). \end{aligned} \tag{5.15}$$

Posterior Decomposition

The SOLVE-GP posterior can be decomposed as a sum of three independent Gaussian processes:

$$\begin{aligned} f(\mathbf{x}) &= g(\mathbf{x}) + h(\mathbf{x}) + l(\mathbf{x}), \\ g(\mathbf{x}) &\sim \mathcal{GP}\left(\mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}\mathbf{m_u}, \mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}\mathbf{S_u}\mathbf{K_{uu}^{-1}}\mathbf{k_u}(\mathbf{x})\right), \\ h(\mathbf{x}) &\sim \mathcal{GP}\left(\mathbf{c_v}(\mathbf{x})^\top\mathbf{C_{vv}^{-1}}\mathbf{m_v}, \mathbf{c_v}(\mathbf{x})^\top\mathbf{C_{vv}^{-1}}\mathbf{S_v}\mathbf{C_{vv}^{-1}}\mathbf{c_v}(\mathbf{x})\right), \\ l(\mathbf{x}) &\sim \mathcal{GP}\left(0, k(\mathbf{x},\mathbf{x}) - \mathbf{k_u}(\mathbf{x})^\top\mathbf{K_{uu}^{-1}}\mathbf{k_u}(\mathbf{x}) - \mathbf{c_v}(\mathbf{x})^\top\mathbf{C_{vv}^{-1}}\mathbf{c_v}(\mathbf{x})\right). \end{aligned} \tag{5.16}$$

SOLVE-GP's parameterisation of the variational distribution $q(\mathbf{u}, \mathbf{v})$ results in a posterior in which the orthogonal inducing points contribute to both the mean and covariance. As such, SOLVE-GP has greater flexibility in approximating the posterior as compared with ODVGP. Note that once again, all contributions of the orthogonal inducing points to the posterior involve orthogonal covariances.

5.2 Variational Lower Bound

We can construct a variational lower bound on the orthogonal SVGP marginal likelihood by writing out the KL divergence of the approximate posterior $q(\mathbf{f}, \mathbf{u}, \mathbf{v})$ from the true posterior $p(\mathbf{f}, \mathbf{u}, \mathbf{v}|\mathbf{y})$ and applying Bayes' rule:

$$\begin{aligned}
\mathrm{KL}[q(\mathbf{f}, \mathbf{u}, \mathbf{v})||p(\mathbf{f}, \mathbf{u}, \mathbf{v}|\mathbf{y})] &= \mathbb{E}_{q(\mathbf{f},\mathbf{u},\mathbf{v})}\left[\log \frac{q(\mathbf{f}, \mathbf{u}, \mathbf{v})}{p(\mathbf{f}, \mathbf{u}, \mathbf{v}|\mathbf{y})}\right] \\
&= \mathbb{E}_{q(\mathbf{f},\mathbf{u},\mathbf{v})}\left[\log \frac{\cancel{p(\mathbf{f}|\mathbf{u}, \mathbf{v})}q(\mathbf{v}|\mathbf{u})q(\mathbf{u})p(\mathbf{y})}{p(\mathbf{y}|\mathbf{f})\cancel{p(\mathbf{f}|\mathbf{u}, \mathbf{v})}p(\mathbf{v}|\mathbf{u})p(\mathbf{u})}\right] \\
&= \log p(\mathbf{y}) - \mathbb{E}_{q(\mathbf{f},\mathbf{u},\mathbf{v})}[\log p(\mathbf{y}|\mathbf{f})] + \mathbb{E}_{q(\mathbf{u})}[\mathrm{KL}[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})]] + \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})].
\end{aligned} \tag{5.17}$$

Note that this bound is independent of the parameterisation of $q(\mathbf{u}, \mathbf{v})$, i.e. it is shared by both ODVGP and SOLVE-GP as their posteriors factorise in the same way. As with SVGP, the expensive $\mathcal{O}(N^3)$ term $p(\mathbf{f}|\mathbf{u}, \mathbf{v})$ cancels out. Rearranging for the marginal likelihood, we arrive at the following lower bound:

$$\log p(\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{f},\mathbf{u},\mathbf{v})}[\log p(\mathbf{y}|\mathbf{f})] - \mathbb{E}_{q(\mathbf{u})}[\mathrm{KL}[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})]] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \mathcal{L}. \tag{5.18}$$

Writing the expected log likelihood term in integral form and factorising the approximate posterior $q(\mathbf{f}, \mathbf{u}, \mathbf{v})$, we see that $\mathbf{u}$ and $\mathbf{v}$ can be integrated out:

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{f},\mathbf{u},\mathbf{v})}[\log p(\mathbf{y}|\mathbf{f})] &= \iiint p(\mathbf{f}|\mathbf{u}, \mathbf{v})q(\mathbf{u}, \mathbf{v}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} d\mathbf{u} d\mathbf{v} \\
&= \int \left\{ \iint p(\mathbf{f}|\mathbf{u}, \mathbf{v})q(\mathbf{u}, \mathbf{v}) d\mathbf{u} d\mathbf{v} \right\} \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\
&= \int q(\mathbf{f}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} \\
&= \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})].
\end{aligned} \tag{5.19}$$

We can therefore express the lower bound equivalently as:

$$\mathcal{L} = \mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] - \mathbb{E}_{q(\mathbf{u})}\left[\mathrm{KL}[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})]\right] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]. \tag{5.20}$$

If we factorise the likelihood, the expected log likelihood term can be expressed as a sum:

$$\mathcal{L} = \sum_{n=1}^{N} \mathbb{E}_{q(f_n)}[\log p(y_n|f_n)] - \mathbb{E}_{q(\mathbf{u})}\left[\mathrm{KL}[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})]\right] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]. \tag{5.21}$$

The orthogonal lower bound is equivalent to the SVGP lower bound in eq. 4.10 minus a conditional KL divergence term (the analytic form of which depends on the orthogonal parameterisation, i.e. ODVGP vs. SOLVE-GP). It is a function of the inducing input locations ($\mathbf{Z}$, $\mathbf{O}$) and the variational distribution $q(\mathbf{u}, \mathbf{v})$, i.e. $\mathcal{L} = \mathcal{L}((\mathbf{Z}, \mathbf{O}), q(\mathbf{u}, \mathbf{v}))$[1].

[1] In addition to the kernel hyperparameters $\boldsymbol{\theta}$, which we have dropped for notational simplicity.

5.3 Collapsed Bound for Regression

The authors of SOLVE-GP claim to derive a collapsed bound for SOLVE-GP in appendix C of [10], but their bound is only partially collapsed as the final result is an explicit function of the variational parameters $\mathbf{m_v}$ and $\mathbf{S_v}$ and one of the KL terms is left in its abstract form. We go a step further by deriving a fully collapsed bound, taking inspiration from Titsias' derivation in [17].

5.3.1 Finding the Optimal Variational Parameters

We will show the derivation of the optimal variational parameters using the SOLVE-GP parameterisation, as ODVGP can be viewed as a special case of SOLVE-GP with one less variational parameter. To begin, we write out the analytic form of the expected log likelihood term in eq. 5.20 explicitly. This can be done using eq. 4.16 to obtain a Gaussian log density minus a trace term:

$$\mathbb{E}_{q(\mathbf{f})}[\log p(\mathbf{y}|\mathbf{f})] = \log \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \sigma^2 \mathbf{I_N}) - \frac{1}{2\sigma^2} \operatorname{Tr}(\boldsymbol{\Sigma}). \tag{5.22}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the posterior mean and covariance in eq. 5.14, i.e. $q(\mathbf{f}) = \mathcal{N}(\mathbf{f}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$. Writing out the full expressions for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and expanding the log density term:

$$\begin{aligned}
=&-\frac{1}{2\sigma^2}\operatorname{Tr}\Big(\mathbf{y}\mathbf{y}^\top - 2\mathbf{y}(\mathbf{m_u}^\top \mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \mathbf{m_v}^\top \mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}) + (\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} + \mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{m_v})(\mathbf{m_u}^\top \mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \mathbf{m_v}^\top \mathbf{C_{vv}^{-1}}\mathbf{C_{vf}})\Big) \\
&-\frac{1}{2\sigma^2}\operatorname{Tr}\Big(\mathbf{K_{ff}} - \mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} - \mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}(\mathbf{C_{vv}} - \mathbf{S_v})\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}\Big) - \frac{N}{2}\log 2\pi\sigma^2.
\end{aligned} \tag{5.23}$$

The conditional KL term in eq. 5.20 has an analytic form as both conditionals are Gaussian:

$$\begin{aligned}
\operatorname{KL}\left[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})\right] &= \operatorname{KL}\left[\mathcal{N}(\mathbf{v}; \mathbf{m_v} + \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{S_v})||\mathcal{N}(\mathbf{v}; \mathbf{K_{vu}}\mathbf{K_{uu}^{-1}}\mathbf{u}, \mathbf{C_{vv}})\right] \\
&= \frac{1}{2}\left\{\operatorname{Tr}\left(\mathbf{C_{vv}^{-1}}\mathbf{S_v}\right) + \mathbf{m_v}^\top \mathbf{C_{vv}^{-1}}\mathbf{m_v} - M + \log\frac{|\mathbf{C_{vv}}|}{|\mathbf{S_v}|}\right\}.
\end{aligned} \tag{5.24}$$

Though both conditionals $q(\mathbf{v}|\mathbf{u})$ and $p(\mathbf{v}|\mathbf{u})$ depend on $\mathbf{u}$, the terms containing $\mathbf{u}$ get subtracted out in the expression of the KL divergence, and thus the expectation over $q(\mathbf{u})$ in eq. 5.20 has no effect, i.e. $\mathbb{E}_{q(\mathbf{u})}[\operatorname{KL}\left[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})\right]] = \operatorname{KL}\left[q(\mathbf{v}|\mathbf{u})||p(\mathbf{v}|\mathbf{u})\right]$. Note, however, that this is merely a result of SOLVE-GP's parameterisation of $q(\mathbf{v}|\mathbf{u})$ (although the same holds for ODVGP) and not a fundamental result for orthogonal SVGP's. The second KL term is the same as in the SVGP lower bound and thus has the same analytic form as eq. 4.18. We can now compute the gradients of the orthogonal lower bound (eq. 5.22 less eq.'s 5.24 and 4.18) with respect to SOLVE-GP's variational parameters:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{m_u}} &= -\sigma^{-2}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\left(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} + \mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{m_v} - \mathbf{y}\right) - \mathbf{K_{uu}^{-1}}\mathbf{m_u}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{m_v}} &= -\sigma^{-2}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}\left(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} + \mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{m_v} - \mathbf{y}\right) - \mathbf{C_{vv}^{-1}}\mathbf{m_v}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{S_u}} &= -\frac{1}{2}\sigma^{-2}\left(\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\right) - \frac{1}{2}\mathbf{K_{uu}^{-1}} + \frac{1}{2}\mathbf{S_u^{-1}}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{S_v}} &= -\frac{1}{2}\sigma^{-2}\left(\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\right) - \frac{1}{2}\mathbf{C_{vv}^{-1}} + \frac{1}{2}\mathbf{S_v^{-1}}.
\end{aligned} \tag{5.25}$$

Solving this system of equations for zero gradients, rearranging terms and applying matrix inversion lemma to simplify some grouped inversions, we arrive at the following closed-form expressions for the optimal variational parameters (which may be substituted into eq. 5.9 or 5.14 to make predictions):

$$\begin{aligned}
\mathbf{m}_\mathbf{u}^* &= \mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \mathbf{K_{uf}}\left(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}} + \sigma^2\mathbf{I_N}\right)^{-1}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uf}}\left(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}} + \sigma^2\mathbf{I_N}\right)^{-1}\mathbf{y}, \\
\mathbf{m}_\mathbf{v}^* &= \mathbf{C_{vv}}\left[\mathbf{C_{vv}} + \mathbf{C_{vf}}\left(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \sigma^2\mathbf{I_N}\right)^{-1}\mathbf{C_{fv}}\right]^{-1}\mathbf{C_{vf}}\left(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}} + \sigma^2\mathbf{I_N}\right)^{-1}\mathbf{y}, \\
\mathbf{S}_\mathbf{u}^* &= \mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uu}}, \\
\mathbf{S}_\mathbf{v}^* &= \mathbf{C_{vv}}\left[\mathbf{C_{vv}} + \sigma^{-2}\mathbf{C_{vf}}\mathbf{C_{fv}}\right]^{-1}\mathbf{C_{vv}}.
\end{aligned} \tag{5.26}$$

Identical expressions of the common variational parameters ($\mathbf{m_u}$, $\mathbf{m_v}$, $\mathbf{S_u}$) would be obtained if we repeated this derivation using the ODVGP parameterisation, but $\mathbf{S_v}$ would be restricted to equal $\mathbf{C_{vv}}$.

5.3.2 Collapsing the Lower Bound

Substituting the optimal variational parameters into the orthogonal lower bound in eq. 5.20 results in the orthogonal collapsed bound for regression; however, doing so involves a good deal of rather painful algebra and repeated applications of matrix inversion lemma. Instead, we will derive an equivalent expression for the orthogonal collapsed bound using Titsias' Jensen's inequality reversal trick [5], shown previously for SVGP. Note that the collapsed bound is independent of the parameterisation of the variational distribution $q(\mathbf{u}, \mathbf{v})$ and is therefore shared by both ODVGP and SOLVE-GP. To begin, we note that eq. 5.18 may be expressed equivalently in integral form as:

$$\mathcal{L} = \iint q(\mathbf{u}, \mathbf{v}) \left\{ \int p(\mathbf{f}|\mathbf{u}, \mathbf{v}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} + \log \frac{p(\mathbf{u}, \mathbf{v})}{q(\mathbf{u}, \mathbf{v})} \right\} d\mathbf{u} d\mathbf{v}. \tag{5.27}$$

The inner integral with respect to the function values $\mathbf{f}$ can be evaluated using eq. 4.16 to obtain a Gaussian log density minus a trace term:

$$\int p(\mathbf{f}|\mathbf{u}, \mathbf{v}) \log p(\mathbf{y}|\mathbf{f}) d\mathbf{f} = \log \mathcal{N}\left(\mathbf{y}; \boldsymbol{\alpha}, \sigma^2\mathbf{I_N}\right) - \frac{1}{2\sigma^2}\operatorname{Tr}\left(\boldsymbol{\beta}\right). \tag{5.28}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ denote the mean and covariance of the prior conditional $p(\mathbf{f}|\mathbf{u}, \mathbf{v})$ in eq. 5.3, i.e. $p(\mathbf{f}|\mathbf{u}, \mathbf{v}) = \mathcal{N}(\mathbf{f}; \boldsymbol{\alpha}, \boldsymbol{\beta})$. Substituting this back into eq. 5.27, we obtain:

$$\mathcal{L} = \iint q(\mathbf{u}, \mathbf{v}) \log \left[\frac{\mathcal{N}(\mathbf{y}; \boldsymbol{\alpha}, \sigma^2\mathbf{I_N}) p(\mathbf{u}, \mathbf{v})}{q(\mathbf{u}, \mathbf{v})} \right] d\mathbf{u} d\mathbf{v} - \frac{1}{2\sigma^2}\operatorname{Tr}\left(\boldsymbol{\beta}\right). \tag{5.29}$$

We now repeat the trick from eq. 4.25 and reverse Jensen's inequality on the expectation term above, pulling the logarithm outside the integral and allowing the variational distribution $q(\mathbf{u}, \mathbf{v})$ to cancel:

$$\iint q(\mathbf{u}, \mathbf{v}) \log \left[\frac{\mathcal{N}(\mathbf{y}; \boldsymbol{\alpha}, \sigma^2\mathbf{I_N}) p(\mathbf{u}, \mathbf{v})}{q(\mathbf{u}, \mathbf{v})} \right] d\mathbf{u} d\mathbf{v} \leq \log \iint \cancel{q(\mathbf{u}, \mathbf{v})} \frac{\mathcal{N}(\mathbf{y}; \boldsymbol{\alpha}, \sigma^2\mathbf{I_N}) p(\mathbf{u}, \mathbf{v})}{\cancel{q(\mathbf{u}, \mathbf{v})}} d\mathbf{u} d\mathbf{v}. \tag{5.30}$$

The resulting integration can be done using Gaussian linear transformation rules. Writing the prior conditional mean $\boldsymbol{\alpha}$ as $\mathbf{A}[\mathbf{u},\mathbf{v}]^\top$ as we did in eq. 5.6 and letting $\mathbf{K}$ denote the prior covariance (the block matrix we inverted in eq. 5.4), we get:

$$\begin{aligned}\iint \mathcal{N}(\mathbf{y};\boldsymbol{\alpha},\sigma^2\mathbf{I_N})p(\mathbf{u},\mathbf{v})d\mathbf{u}d\mathbf{v} &= \iint \mathcal{N}(\mathbf{y};\mathbf{A}[\mathbf{u},\mathbf{v}]^\top,\sigma^2\mathbf{I_N})\mathcal{N}([\mathbf{u},\mathbf{v}]^\top;\mathbf{0},\mathbf{K})d\mathbf{u}d\mathbf{v}\\ &= \mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{AKA}^\top+\sigma^2\mathbf{I_N}).\end{aligned} \tag{5.31}$$

Bringing back the trace term from eq. 5.29:

$$\mathcal{L}^* = \mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{AKA}^\top+\sigma^2\mathbf{I_N}) - \frac{1}{2\sigma^2}\operatorname{Tr}(\boldsymbol{\beta}). \tag{5.32}$$

Substituting in the prior covariance $\mathbf{K}$ as well as $\mathbf{A}$ and $\boldsymbol{\beta}$ from the prior conditional in eq. 5.3 and completing the block matrix multiplications, we arrive at the orthogonal collapsed bound for regression:

$$\mathcal{L}^* = \mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}+\mathbf{C_{fv}C_{vv}^{-1}C_{vf}}+\sigma^2\mathbf{I_N}) - \frac{1}{2\sigma^2}\operatorname{Tr}\left(\mathbf{K_{ff}}-\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}-\mathbf{C_{fv}C_{vv}^{-1}C_{vf}}\right). \tag{5.33}$$

The orthogonal collapsed bound has a similar form to the SVGP collapsed bound, with orthogonal covariance terms appearing in both the log density and trace terms. We reaffirm that this is an identical result as would have been obtained substituting the optimal variational parameters in eq. 5.26 into the uncollapsed bound in eq. 5.20, and that the same bound is shared by ODVGP and SOLVE-GP. The orthogonal collapsed bound is only a function of the inducing input locations $(\mathbf{Z},\mathbf{O})$, i.e. $\mathcal{L}^* = \mathcal{L}^*((\mathbf{Z},\mathbf{O}))$. Just as we distinguish between SGPR and SVGP, we will use the acronyms ODGPR and SOLVE-GPR to indicate when we are doing orthogonal collapsed bound regression (or orthogonal SGPR) using the ODVGP and SOLVE-GP parameterisations.

5.4 Stable and Efficient Implementation

Now that we have derived the orthogonal collapsed bound and the optimal variational parameters for ODVGP and SOLVE-GP, we derive a novel numerically stable and efficient implementations of ODGPR and SOLVE-GPR using Cholesky decompositions, taking inspiration from the Hensman and Matthews' SGPR implementation [18].

5.4.1 Stable and Efficient Lower Bound

We begin by expanding the log density term in eq. 5.33:

$$-\frac{1}{2}\Big(\log|\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}+\mathbf{C_{fv}C_{vv}^{-1}C_{vf}}+\sigma^2\mathbf{I_N}| + \mathbf{y}^\top\left[\mathbf{K_{fu}K_{uu}^{-1}K_{uf}}+\mathbf{C_{fv}C_{vv}^{-1}C_{vf}}+\sigma^2\mathbf{I_N}\right]^{-1}\mathbf{y} + N\log 2\pi\Big). \tag{5.34}$$

A key observation is that $\mathbf{K_{fu}K_{uu}^{-1}K_{uf}} + \mathbf{C_{fv}C_{vv}^{-1}C_{vf}} + \sigma^2\mathbf{I_N}$ can be expressed as a product of block matrices, allowing us to use matrix inversion lemma and matrix determinant lemma to obtain cheaper and better conditioned expressions for the inverse and log determinant terms:

$$\mathbf{K_{fu}K_{uu}^{-1}K_{uf}} + \mathbf{C_{fv}C_{vv}^{-1}C_{vf}} + \sigma^2\mathbf{I_N} = \begin{bmatrix}\mathbf{K_{fu}} & \mathbf{C_{fv}}\end{bmatrix} \begin{bmatrix}\mathbf{K_{uu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_{vv}}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{K_{uf}} \\ \mathbf{C_{vf}}\end{bmatrix} + \sigma^2\mathbf{I_N}. \tag{5.35}$$

Using this block expression of $\mathbf{K_{fu}K_{uu}^{-1}K_{uf}} + \mathbf{C_{fv}C_{vv}^{-1}C_{vf}} + \sigma^2\mathbf{I_N}$, we can repeat Hensman and Matthews' derivation of the SGPR lower bound in [18] as shown in the previous chapter, only using a different rotation matrix and different definitions of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{c}$. We use the following block diagonal Cholesky matrix for rotating the inverse term (after applying matrix inversion lemma):

$$\mathbf{L} = \begin{bmatrix}\mathbf{L_u} & \mathbf{0} \\ \mathbf{0} & \mathbf{L_v}\end{bmatrix}, \tag{5.36}$$

where $\mathbf{L_u}$ and $\mathbf{L_v}$ are the lower Cholesky factors of $\mathbf{K_{uu}}$ and $\mathbf{C_{vv}}$, respectively, i.e. $\mathbf{K_{uu}} = \mathbf{L_u}\mathbf{L_u}^\top$ and $\mathbf{C_{vv}} = \mathbf{L_v}\mathbf{L_v}^\top$. If we let $\mathbf{A} = \sigma^{-1}\mathbf{L}^{-1}[\mathbf{K_{uf}}, \mathbf{C_{vf}}]^\top$, we can then let $\mathbf{B} = \mathbf{I_M} + \mathbf{AA}^\top = \mathbf{L_B}\mathbf{L_B}^\top$ and $\mathbf{c} = \sigma^{-1}\mathbf{L_B^{-1}Ay}$ as we did when deriving the SGPR equations and conveniently arrive at an expression for the orthogonal lower bound which is identical to eq. 4.37. Computing $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{L_B}$ directly has a computational complexity which is roughly equivalent to the SGPR lower bound. Fortunately, we can exploit the block diagonal structure of our Cholesky matrix $\mathbf{L}$ to find some shortcuts which make our implementation more efficient. First, we notice that the calculation of the intermediate matrix $\mathbf{A}$ can be broken into two cheaper calculations:

$$\mathbf{A} = \sigma^{-1}\begin{bmatrix}\mathbf{L_u} & \mathbf{0} \\ \mathbf{0} & \mathbf{L_v}\end{bmatrix}^{-1} \begin{bmatrix}\mathbf{K_{uf}} \\ \mathbf{C_{vf}}\end{bmatrix} = \begin{bmatrix}\sigma^{-1}\mathbf{L_u^{-1}K_{uf}} \\ \sigma^{-1}\mathbf{L_v^{-1}C_{vf}}\end{bmatrix} = \begin{bmatrix}\mathbf{A_u} \\ \mathbf{A_v}\end{bmatrix}. \tag{5.37}$$

I.e., we can obtain $\mathbf{A}$ by computing $\mathbf{A_u}$ and $\mathbf{A_v}$ separately. We then notice that the matrix multiplication $\mathbf{AA}^\top$ in the definition of $\mathbf{B}$ can be split into three cheaper multiplications:

$$\mathbf{AA}^\top = \begin{bmatrix}\mathbf{A_u} \\ \mathbf{A_v}\end{bmatrix}\begin{bmatrix}\mathbf{A_u}^\top & \mathbf{A_v}^\top\end{bmatrix} = \begin{bmatrix}\mathbf{A_u}\mathbf{A_u}^\top & \mathbf{A_u}\mathbf{A_v}^\top \\ \mathbf{A_v}\mathbf{A_u}^\top & \mathbf{A_v}\mathbf{A_v}^\top\end{bmatrix} = \begin{bmatrix}\mathbf{B'_u} & \mathbf{B_{uv}} \\ \mathbf{B_{uv}}^\top & \mathbf{B'_v}\end{bmatrix}. \tag{5.38}$$

I.e., we can obtain $\mathbf{AA}^\top$ by computing $\mathbf{B'_u}$, $\mathbf{B'_v}$ and $\mathbf{B_{uv}}$ separately. We can then obtain $\mathbf{B}$ as follows:

$$\mathbf{B} = \mathbf{I_M} + \mathbf{AA}^\top = \begin{bmatrix}\mathbf{I_{M_1}} + \mathbf{B'_u} & \mathbf{B_{uv}} \\ \mathbf{B_{uv}}^\top & \mathbf{I_{M_2}} + \mathbf{B'_v}\end{bmatrix} = \begin{bmatrix}\mathbf{B_u} & \mathbf{B_{uv}} \\ \mathbf{B_{uv}}^\top & \mathbf{B_v}\end{bmatrix}. \tag{5.39}$$

The Cholesky decomposition of $\mathbf{B}$ can also be split into cheaper sub-operations by writing out the lower Cholesky factor in block form [22]:

$$\mathbf{L_B} = \begin{bmatrix}\mathbf{L_{B_u}} & \mathbf{0} \\ (\mathbf{L_{B_u}^{-1}B_{uv}})^\top & \mathbf{L_Q}\end{bmatrix}. \tag{5.40}$$

where $\mathbf{B_u} = \mathbf{L_{B_u}}\mathbf{L_{B_u}}^\top$ and $\mathbf{Q} = \mathbf{B_v} - \mathbf{B_{vu}B_u^{-1}B_{uv}} = \mathbf{B_v} - (\mathbf{L_{B_u}^{-1}B_{uv}})^\top(\mathbf{L_{B_u}^{-1}B_{uv}}) = \mathbf{L_Q}\mathbf{L_Q}^\top$ is the Schur complement of $\mathbf{B_u}$. Therefore, the Cholesky factor of $\mathbf{B}$ can be obtained by computing the Cholesky factor $\mathbf{L_{B_u}}$, computing $\mathbf{L_{B_u}^{-1}B_{uv}}$, then computing the Schur complement $\mathbf{Q}$ and its Cholesky factor. While splitting the calculations of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{c}$ into sub-operations makes the lower bound im-

plementation look more complex, it is precisely this splitting up of operations which gives orthogonal SGPR its computational speedup over SGPR. A full analysis of the computational complexities of SGPR vs. orthogonal SGPR can be found in the following chapter.

5.4.2 Stable and Efficient Predictive Equations

While we were able to come up with a stable expression for the lower bound which matched the corresponding expression for SGPR (adding a few computational tricks along the way to make it more efficient), obtaining a stable expression for the predictive equations is a bit more challenging. Although the optimal covariance parameters $\mathbf{S_u}$ and $\mathbf{S_v}$ have the same form as the optimal covariance parameter for SGPR, the optimal mean parameters $\mathbf{m_u}$ and $\mathbf{m_v}$ have more complex expressions involving additional inversions. Nevertheless, we can derive a stable expression for the posterior mean through repeated Cholesky decompositions, rotations and applications of matrix inversion lemma. We begin with the formula for the posterior mean $\boldsymbol{\mu}_* = \mu(\mathbf{X}_*)$ at some test points $\mathbf{X_*}$ (recall that ODVGP and SOLVE-GP share a common posterior mean):

$$\boldsymbol{\mu}_* = \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} + \mathbf{C_{*v}}\mathbf{C_{vv}^{-1}}\mathbf{m_v}. \tag{5.41}$$

Let us examine the first term in the summation above. Substituting in the optimal $\mathbf{m_u}$ from eq. 5.26, a couple terms cancel and we get:

$$\begin{aligned}\mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{m_u} &= \mathbf{K_{*u}}\cancel{\mathbf{K_{uu}^{-1}}}\cancel{\mathbf{K_{uu}}}\Big[\mathbf{K_{uu}}+\mathbf{K_{uf}}\big(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N}\big)^{-1}\mathbf{K_{fu}}\Big]^{-1}\mathbf{K_{uf}}\big(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N}\big)^{-1}\mathbf{y} \\ &= \mathbf{K_{*u}}\Big[\mathbf{K_{uu}}+\mathbf{K_{uf}}\big(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N}\big)^{-1}\mathbf{K_{fu}}\Big]^{-1}\mathbf{K_{uf}}\big(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N}\big)^{-1}\mathbf{y}.\end{aligned} \tag{5.42}$$

There are two grouped inversions in above expression, which we will deal with individually. First, we apply matrix inversion lemma on the inversion in square brackets:

$$\begin{aligned}&\Big[\mathbf{K_{uu}}+\mathbf{K_{uf}}\Big(\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N}\Big)^{-1}\mathbf{K_{fu}}\Big]^{-1} = \\ &\qquad\mathbf{K_{uu}^{-1}}-\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\Big(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}+\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N}\Big)^{-1}\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}.\end{aligned} \tag{5.43}$$

We recognise that the inversion in round brackets above is the same inversion from the log density term in the collapsed bound (eq. 5.34). In eq. 4.32, we showed that the inversion in the SGPR collapsed bound log density $(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}+\sigma^2\mathbf{I_N})^{-1}$ could be expressed as $\sigma^{-2}(\mathbf{I_N}-\mathbf{A}^\top\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A})$, and the same result holds for $(\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}+\mathbf{C_{fv}}\mathbf{C_{vv}^{-1}}\mathbf{C_{vf}}+\sigma^2\mathbf{I_N})^{-1}$ using our definitions of $\mathbf{A}$ and $\mathbf{L_B}$ from the orthogonal lower bound implementation if we apply matrix inversion lemma on the block expression in eq. 5.35 and rotate by our block diagonal Cholesky matrix $\mathbf{L}$. Substituting in this result:

$$= \mathbf{K_{uu}^{-1}} - \sigma^{-2}\mathbf{K_{uu}^{-1}}\mathbf{K_{uf}}\Big(\mathbf{I_N}-\mathbf{A}^\top\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A}\Big)\mathbf{K_{fu}}\mathbf{K_{uu}^{-1}}. \tag{5.44}$$

Substituting in the Cholesky decomposition $\mathbf{K_{uu}} = \mathbf{L_u}\mathbf{L_u}^\top$:

$$= \mathbf{L_u^{-\top}}\mathbf{L_u^{-1}} - \sigma^{-2}\mathbf{L_u^{-\top}}\mathbf{L_u^{-1}}\mathbf{K_{uf}}\Big(\mathbf{I_N}-\mathbf{A}^\top\mathbf{L_B^{-\top}}\mathbf{L_B^{-1}}\mathbf{A}\Big)\mathbf{K_{fu}}\mathbf{L_u^{-\top}}\mathbf{L_u^{-1}}. \tag{5.45}$$

Letting $\mathbf{A_u} = \sigma^{-1}\mathbf{L_u^{-1}K_{uf}}$:

$$\begin{aligned} &= \mathbf{L_u^{-\top}L_u^{-1}} - \mathbf{L_u^{-\top}A_u}\Big(\mathbf{I_N} - \mathbf{A}^\top\mathbf{L_B^{-\top}L_B^{-1}A}\Big)\mathbf{A_u^\top L_u^{-1}} \\ &= \mathbf{L_u^{-\top}}\Big[\mathbf{I_{M_1}} - \mathbf{A_u}\Big(\mathbf{I_N} - \mathbf{A}^\top\mathbf{L_B^{-\top}L_B^{-1}A}\Big)\mathbf{A_u^\top}\Big]\mathbf{L_u^{-1}}. \end{aligned} \tag{5.46}$$

Letting $\mathbf{C} = \mathbf{L_B^{-1}A}$ and $\mathbf{D} = \mathbf{I_N} - \mathbf{C}^\top\mathbf{C}$:

$$\begin{aligned} &= \mathbf{L_u^{-\top}}\Big[\mathbf{I_{M_1}} - \mathbf{A_u}\Big(\mathbf{I_N} - \mathbf{C}^\top\mathbf{C}\Big)\mathbf{A_u^\top}\Big]\mathbf{L_u^{-1}} \\ &= \mathbf{L_u^{-\top}}\Big[\mathbf{I_{M_1}} - \mathbf{A_u}\mathbf{D}\mathbf{A_u^\top}\Big]\mathbf{L_u^{-1}}. \end{aligned} \tag{5.47}$$

Finally, we let $\mathbf{E_u} = \mathbf{I_{M_1}} - \mathbf{A_u}\mathbf{D}\mathbf{A_u^\top}$:

$$= \mathbf{L_u^{-\top}E_uL_u^{-1}}. \tag{5.48}$$

Next, we apply matrix inversion lemma on the inversion in round brackets in eq. 5.42:

$$\left(\mathbf{C_{fv}C_{vv}^{-1}C_{vf}} + \sigma^2\mathbf{I_N}\right)^{-1} = \sigma^{-2}\mathbf{I_N} - \sigma^{-4}\mathbf{C_{fv}}\left[\mathbf{C_{vv}} + \sigma^{-2}\mathbf{C_{vf}C_{fv}}\right]^{-1}\mathbf{C_{vf}}. \tag{5.49}$$

We rotate by $\mathbf{L_v}$ to obtain a matrix which is better conditioned for inversion:

$$\begin{aligned} &= \sigma^{-2}\mathbf{I_N} - \sigma^{-4}\mathbf{C_{fv}L_v^{-\top}L_v^\top}\left[\mathbf{L_vL_v^\top} + \sigma^{-2}\mathbf{C_{vf}C_{fv}}\right]^{-1}\mathbf{L_vL_v^{-1}C_{vf}} \\ &= \sigma^{-2}\mathbf{I_N} - \sigma^{-4}\mathbf{C_{fv}L_v^{-\top}}\left[\mathbf{I_{M_2}} + \sigma^{-2}\mathbf{L_v^{-1}C_{vf}C_{fv}L_v^{-\top}}\right]^{-1}\mathbf{L_v^{-1}C_{vf}}. \end{aligned} \tag{5.50}$$

We define $\mathbf{A_v} = \sigma^{-1}\mathbf{L_v^{-1}C_{vf}}$ and $\mathbf{B_v} = \mathbf{I_{M_2}} + \mathbf{A_vA_v^\top} = \mathbf{L_{B_v}L_{B_v}^\top}$:

$$\begin{aligned} &= \sigma^{-2}\mathbf{I_N} - \sigma^{-2}\mathbf{A_v^\top}\left[\mathbf{I_{M_2}} + \mathbf{A_vA_v^\top}\right]^{-1}\mathbf{A_v} \\ &= \sigma^{-2}\Big(\mathbf{I_N} - \mathbf{A_v^\top B_v^{-1}A_v}\Big) \\ &= \sigma^{-2}\Big(\mathbf{I_N} - \mathbf{A_v^\top L_{B_v}^{-\top}L_{B_v}^{-1}A_v}\Big). \end{aligned} \tag{5.51}$$

Letting $\mathbf{C_v} = \mathbf{L_{B_v}^{-1}A_v}$ and $\mathbf{D_v} = \mathbf{I_N} - \mathbf{C_v^\top C_v}$:

$$\begin{aligned} &= \sigma^{-2}\Big(\mathbf{I_N} - \mathbf{C_v^\top C_v}\Big) \\ &= \sigma^{-2}\mathbf{D_v}. \end{aligned} \tag{5.52}$$

Substituting our simplified expressions for the two inversions back into eq. 5.42, we obtain:

$$\begin{aligned}\mathbf{K_{*u}K_{uu}^{-1}m_u} &= \sigma^{-2}\mathbf{K_{*u}L_u^{-\top}E_uL_u^{-1}K_{uf}D_vy} \\ &= \sigma^{-1}\mathbf{K_{*u}L_u^{-\top}E_uA_uD_vy}.\end{aligned} \tag{5.53}$$

To simplify our expression even further, we let $\mathbf{F_{uv}} = \sigma^{-1}\mathbf{E_uA_uD_v}$:

$$= \mathbf{K_{*u}L_u^{-\top}F_{uv}y}. \tag{5.54}$$

If we repeat the same procedure for the second term in the summation in eq. 5.41 with the optimal $\mathbf{m_v}$, we arrive at the following expression for the ODGPR and SOLVE-GPR posterior mean $\boldsymbol{\mu_*}$:

$$\boldsymbol{\mu_*} = \mathbf{K_{*u}L_u^{-\top}F_{uv}y} + \mathbf{C_{*v}L_v^{-\top}F_{vu}y}, \tag{5.55}$$

where $\mathbf{F_{vu}} = \sigma^{-1}\mathbf{E_vA_vD_u}$, $\mathbf{E_v} = \mathbf{I_{M_2}} - \mathbf{A_vDA_v^\top}$, $\mathbf{D_u} = \mathbf{I_N} - \mathbf{C_u^\top C_u}$, $\mathbf{C_u} = \mathbf{L_{B_u}^{-1}A_u}$ and $\mathbf{B_u} = \mathbf{I_{M_1}} + \mathbf{A_uA_u^\top} = \mathbf{L_{B_u}L_{B_u}^\top}$. Next, we will derive a stable and efficient implementation of the posterior covariance $\boldsymbol{\Sigma_{**}} = \Sigma(\mathbf{X_*}, \mathbf{X_*})$ at some test points $\mathbf{X_*}$. Fortunately, this task is much simpler and more similar to the procedure we took in deriving the SGPR covariance. ODVGP and SOLVE-GP have different posterior covariances, so we will handle them separately.

ODVGP has the same posterior covariance (eq. 5.9) as SVGP (eq. 4.4) since for ODVGP, the orthogonal inducing points only contribute to the mean. Furthermore, the optimal $\mathbf{S_u}$ has the same form for both ODVGP (eq. 5.26) and SVGP (eq. 4.20). The ODGPR posterior covariance is therefore the same as the SGPR posterior covariance and can be expressed as:

$$\boldsymbol{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*u}L_u^{-\top}}\left(\mathbf{I_{M_1}} - \mathbf{L_{B_u}^{-\top}L_{B_u}^{-1}}\right)\mathbf{L_u^{-1}K_{u*}}. \tag{5.56}$$

The SOLVE-GP posterior covariance (eq. 5.14) is the same as for SVGP and ODVGP, but with some additional terms which represent the contribution of the orthogonal inducing points. We can follow the same procedure as we did to obtain the SGPR and ODGPR posterior covariance to obtain a stable expression of the additional terms for SOLVE-GP. The full SOLVE-GPR posterior covariance can be expressed as:

$$\boldsymbol{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*u}L_u^{-\top}}\left(\mathbf{I_{M_1}} - \mathbf{L_{B_u}^{-\top}L_{B_u}^{-1}}\right)\mathbf{L_u^{-1}K_{u*}} - \mathbf{C_{*v}L_v^{-\top}}\left(\mathbf{I_{M_2}} - \mathbf{L_{B_v}^{-\top}L_{B_v}^{-1}}\right)\mathbf{L_v^{-1}C_{v*}}. \tag{5.57}$$

5.4.3 Orthogonal SGPR Algorithm

Our novel, stable and efficient implementation of the orthogonal SGPR (ODGPR and SOLVE-GPR) collapsed bound and predictive equations is shown in algorithm 3. There are two slightly redundant calculations on lines 16 and 22, but computing $\mathbf{c}$ this way makes the computational complexity of the inversion squared rather than cubic, and we want to prioritise efficiency in the lower bound over the predictive equations as the lower bound gets computed repeatedly during training. Note again that in practice it is often useful to add "jitter" (a small float value, e.g. 10^{-6}) to the diagonals of $\mathbf{K_{uu}}$ and $\mathbf{C_{vv}}$ to make Cholesky decomposition and follow up operations more numerically stable.

Algorithm 3 ELBO and predictions for orthogonal sparse Gaussian process regression (ODGPR and SOLVE-GPR) via Cholesky decomposition. $\mathbf{A} = \mathbf{B}\backslash\mathbf{C}$ denotes the solution to $\mathbf{BA} = \mathbf{C}$.

Input: $\mathbf{X}$ (training inputs), $\mathbf{y}$ (targets), $\mathbf{Z}$, $\mathbf{O}$ (inducing inputs), k (kernel), σ^2 (noise), $\mathbf{X_*}$ (test inputs)

$\mathbf{K_{ff}} = k(\mathbf{X}, \mathbf{X})$, $\mathbf{K_{uu}} = k(\mathbf{Z}, \mathbf{Z})$, $\mathbf{K_{vv}} = k(\mathbf{O}, \mathbf{O})$, $\mathbf{K_{uf}} = k(\mathbf{Z}, \mathbf{X})$, $\mathbf{K_{uv}} = k(\mathbf{Z}, \mathbf{O})$, $\mathbf{K_{vf}} = k(\mathbf{O}, \mathbf{X})$
$\mathbf{L_u} = \text{Cholesky}(\mathbf{K_{uu}})$
$\boldsymbol{\delta_f} = \mathbf{L_u}\backslash\mathbf{K_{uf}}$, $\boldsymbol{\delta_v} = \mathbf{L_u}\backslash\mathbf{K_{uv}}$
$\mathbf{C_{vf}} = \mathbf{K_{vf}} - \boldsymbol{\delta_v}^\top\boldsymbol{\delta_f}$, $\mathbf{C_{vv}} = \mathbf{K_{vv}} - \boldsymbol{\delta_v}^\top\boldsymbol{\delta_v}$
$\mathbf{L_v} = \text{Cholesky}(\mathbf{C_{vv}})$
$\mathbf{A_u} = \sigma^{-1}\boldsymbol{\delta_f}$, $\mathbf{A_v} = \sigma^{-1}\mathbf{L_v}\backslash\mathbf{C_{vf}}$
$\mathbf{A} = \begin{bmatrix}\mathbf{A_u}\\ \mathbf{A_v}\end{bmatrix}$
$\mathbf{B'_u} = \mathbf{A_u}\mathbf{A_u}^\top$, $\mathbf{B'_v} = \mathbf{A_v}\mathbf{A_v}^\top$, $\mathbf{B_{uv}} = \mathbf{A_u}\mathbf{A_v}^\top$
$\mathbf{AAT} = \begin{bmatrix}\mathbf{B'_u} & \mathbf{B_{uv}}\\ \mathbf{B_{uv}}^\top & \mathbf{B'_v}\end{bmatrix}$
$\mathbf{B_u} = \mathbf{I_{M_1}} + \mathbf{B'_u}$, $\mathbf{B_v} = \mathbf{I_{M_2}} + \mathbf{B'_v}$
$\mathbf{L_{B_u}} = \text{Cholesky}(\mathbf{B_u})$
$\boldsymbol{\gamma} = \mathbf{L_{B_u}}\backslash\mathbf{B_{uv}}$
$\mathbf{Q} = \mathbf{B_v} - \boldsymbol{\gamma}^\top\boldsymbol{\gamma}$
$\mathbf{L_Q} = \text{Cholesky}(\mathbf{Q})$
$\mathbf{L_B} = \begin{bmatrix}\mathbf{L_{B_u}} & \mathbf{0}\\ \boldsymbol{\gamma}^\top & \mathbf{L_Q}\end{bmatrix}$
$\mathbf{c} = \sigma^{-1}\mathbf{L_B}\backslash(\mathbf{Ay})$
$\mathcal{L}^* = -\frac{N}{2}\log 2\pi\sigma^2 - \sum_i \log \mathrm{L_{B_{ii}}} - \frac{1}{2}\sigma^{-2}\mathbf{y}^\top\mathbf{y} + \frac{1}{2}\mathbf{c}^\top\mathbf{c} - \frac{1}{2}\sigma^{-2}\operatorname{Tr}(\mathbf{K_{ff}}) + \frac{1}{2}\operatorname{Tr}(\mathbf{AAT})$
$\mathbf{K_{**}} = k(\mathbf{X_*}, \mathbf{X_*})$, $\mathbf{K_{u*}} = k(\mathbf{Z}, \mathbf{X_*})$, $\mathbf{K_{v*}} = k(\mathbf{O}, \mathbf{X_*})$
$\boldsymbol{\delta_*} = \mathbf{L_u}\backslash\mathbf{K_{u*}}$
$\mathbf{C_{v*}} = \boldsymbol{\delta_v}^\top\boldsymbol{\delta_*}$
$\mathbf{L_{B_v}} = \text{Cholesky}(\mathbf{B_v})$
$\mathbf{C} = \mathbf{L_B}\backslash\mathbf{A}$, $\mathbf{C_u} = \mathbf{L_{B_u}}\backslash\mathbf{A_u}$, $\mathbf{C_v} = \mathbf{L_{B_v}}\backslash\mathbf{A_v}$
$\mathbf{D} = \mathbf{I_N} - \mathbf{C}^\top\mathbf{C}$, $\mathbf{D_u} = \mathbf{I_N} - \mathbf{C_u}^\top\mathbf{C_u}$, $\mathbf{D_v} = \mathbf{I_N} - \mathbf{C_v}^\top\mathbf{C_v}$
$\mathbf{E_u} = \mathbf{I_{M_1}} - \mathbf{A_u}\mathbf{D}\mathbf{A_u}^\top$, $\mathbf{E_v} = \mathbf{I_{M_2}} - \mathbf{A_v}\mathbf{D}\mathbf{A_v}^\top$
$\mathbf{F_{uv}} = \sigma^{-1}\mathbf{E_u}\mathbf{A_u}\mathbf{D_v}$, $\mathbf{F_{vu}} = \sigma^{-1}\mathbf{E_v}\mathbf{A_v}\mathbf{D_u}$
$\boldsymbol{\alpha_u} = \mathbf{L_u}\backslash\mathbf{K_{u*}}$, $\boldsymbol{\alpha_v} = \mathbf{L_v}\backslash\mathbf{C_{v*}}$
$\boldsymbol{\beta_u} = \mathbf{L_{B_u}}\backslash\boldsymbol{\alpha_u}$, $\boldsymbol{\beta_v} = \mathbf{L_{B_v}}\backslash\boldsymbol{\alpha_v}$,
$\boldsymbol{\mu_*} = \boldsymbol{\alpha_u}^\top\mathbf{F_{uv}}\mathbf{y} + \boldsymbol{\alpha_v}^\top\mathbf{F_{vu}}\mathbf{y}$
$\boldsymbol{\Sigma_{**}} = \mathbf{K_{**}} - \boldsymbol{\alpha_u}^\top\boldsymbol{\alpha_u} + \boldsymbol{\beta_u}^\top\boldsymbol{\beta_u}$ (ODGPR)
$\quad = \mathbf{K_{**}} - \boldsymbol{\alpha_u}^\top\boldsymbol{\alpha_u} + \boldsymbol{\beta_u}^\top\boldsymbol{\beta_u} - \boldsymbol{\alpha_v}^\top\boldsymbol{\alpha_v} + \boldsymbol{\beta_v}^\top\boldsymbol{\beta_v}$ (SOLVE-GPR)
return: $\mathcal{L}^*$ (evidence lower bound), $\boldsymbol{\mu_*}$ (mean), $\boldsymbol{\Sigma_{**}}$ (covariance)

Chapter 6

Probing and Benchmarking Orthogonal Sparse Variational Gaussian Processes

In the previous two chapters, we introduced the original sparse variational Gaussian process model as proposed by Titsias [5] as well as two more recent "orthogonal" SVGP models: ODVGP [9] and SOLVE-GP [10]. The primary advantage of the orthogonal models over the original SVGP model as claimed by their authors is that more inducing points can be used for a fixed computational budget, theoretically allowing for greater flexibility in modelling the posterior. In this chapter, we will investigate the behaviour and characteristics of the orthogonal models and benchmark them against the original SVGP model when applied to regression in their collapsed forms. We will test out our novel implementation of orthogonal SGPR and explicitly analyse its computational complexity, explore posterior equivalencies with SGPR and analyse the value of orthogonal inducing points as compared with regular inducing points. Our aim is to thoroughly understand orthogonal SVGP approximations and demonstrate when they are likely to perform well in practice and when they are not. We will place emphasis on how well the various models approximate the true (exact) Gaussian process, rather than focussing on predictive performance.

6.1 Computational Complexity

We begin by explicitly analysing the computational complexity of our orthogonal SGPR implementation, benchmarking it against SGPR and exact GP regression (GPR). We specifically look at cubic-cost operations required to compute the training objective function (log marginal likelihood for GPR, ELBO for SGPR and orthogonal SGPR) as they need to be computed at every gradient update during training. In our stable and efficient implementations of GPR (algorithm 1), SGPR (algorithm 2) and orthogonal SGPR (algorithm 3), the most expensive operations are matrix multiplications, Cholesky decompositions and solving triangular systems (inverting triangular matrices). Table 6.1 shows a breakdown of the costs and frequencies of these operations in each algorithm, summarising the overall complexity of each at the bottom. The overall cost of orthogonal SGPR is cubic in the greatest number of inducing points between the regular and orthogonal sets $\max(M_1, M_2)$. For a given *total* number of inducing points $M = M_1 + M_2$, the cheapest training cost therefore occurs when inducing points are allocated evenly between the two sets, i.e. when $M_1 = M_2 = M/2$. If we add the costs of all matrix multiplications, Cholesky decompositions and triangular solves for SGPR with M inducing points, we get a total cost of $\mathcal{O}(2M^2N + 2M^3)$; if we do the same for orthogonal SGPR with $M_1 = M_2 = M/2$, we get a total cost of $\mathcal{O}(\frac{3}{2}M^2N + M^3)$. We will consider three extreme scenarios to help compare these costs:

	GPR	SGPR	ODGPR / SOLVE-GPR
Matmul	–	$\mathcal{O}(M^2N) \times 1$	$\mathcal{O}(M_1M_2N) \times 2$ $\mathcal{O}(M_1M_2^2) \times 2$ $\mathcal{O}(M_1^2N) \times 1$ $\mathcal{O}(M_2^2N) \times 1$
Cholesky	$\mathcal{O}(N^3) \times 1$	$\mathcal{O}(M^3) \times 2$	$\mathcal{O}(M_1^3) \times 2$ $\mathcal{O}(M_2^3) \times 2$
Trisolve	–	$\mathcal{O}(M^2N) \times 1$	$\mathcal{O}(M_1^2M_2) \times 2$ $\mathcal{O}(M_1^2N) \times 1$ $\mathcal{O}(M_2^2N) \times 1$
Total:	$\mathcal{O}(N^3)$	$\mathcal{O}(M^2N + M^3)$	$\mathcal{O}(\bar{M}^2N + \bar{M}^3)$

Table 6.1: Cubic-cost operations required to compute the training objective function for GPR, SGPR and orthogonal SGPR. N denotes the number of datapoints in all cases. M denotes the number of inducing points for SGPR. M_1 denotes the number of regular inducing points and M_2 the number of orthogonal inducing points for orthogonal SGPR. We use $\bar{M}$ to represent $\max(M_1, M_2)$.

- When $M \ll N$, i.e. when $M^2N \gg M^3$, the cost of SGPR becomes approximately $\mathcal{O}(2M^2N)$, while the cost of orthogonal SGPR becomes approximately $\mathcal{O}(\frac{3}{2}M^2N)$. In this case, orthogonal SGPR is 25% cheaper than SGPR.
- When $M \approx N$, i.e. when $M^2N \approx M^3$, the cost of SGPR becomes approximately $\mathcal{O}(4M^3)$, while the cost of orthogonal SGPR becomes approximately $\mathcal{O}(\frac{5}{2}M^3)$. In this case, orthogonal SGPR is 37.5% cheaper than SGPR.
- When $M \gg N$, i.e. when $M^2N \ll M^3$, the cost of SGPR becomes approximately $\mathcal{O}(2M^3)$, while the cost of orthogonal SGPR becomes approximately $\mathcal{O}(M^3)$. In this case, orthogonal SGPR is 50% cheaper than SGPR.

Accordingly, for 50/50 allocation of regular and orthogonal points, orthogonal SGPR ranges from being 25-50% cheaper than SGPR for the same *total* number of inducing points, with cost savings improving as the number of inducing points increases (although the case where $M \gg N$ resulting in a 50% cost reduction is unlikely in practice).

6.2 Visualising Predictions

To help us visualise the differences between approximate models, we show the predictions of SGPR, ODGPR and SOLVE-GPR with a squared exponential kernel on Snelson and Gharahmani's dataset [4] for various numbers of inducing points in figure 6.1. We randomly initialise the inducing points along the input domain and use the L-BFGS algorithm to optimise their placement (in addition to the kernel hyperparameters) by maximising the collapsed bound. We can see that for a fixed number of inducing points $M = M_1 + M_2$, the mean predictions are very similar for all models, but ODGPR has consistently worse covariance. This is not surprising, as the orthogonal inducing points only contribute to the

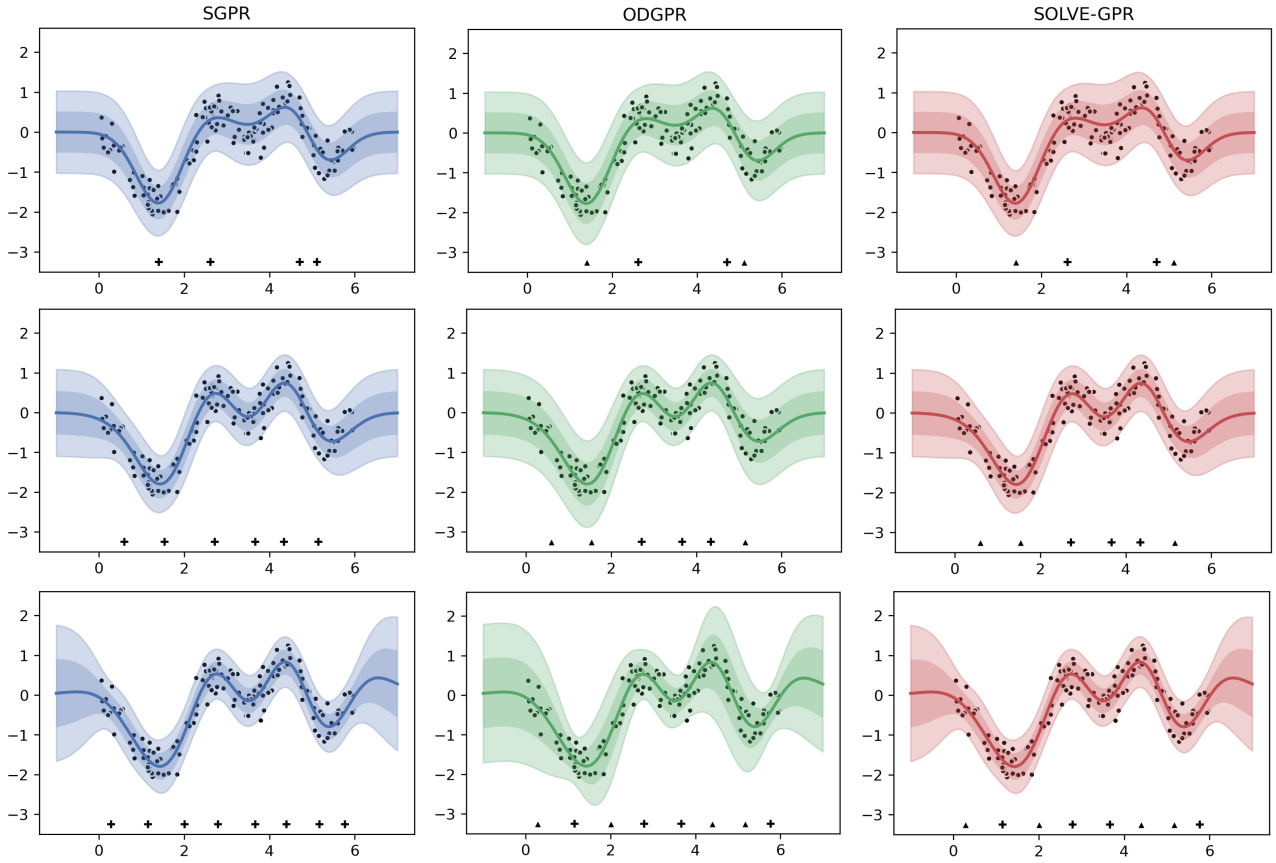

Figure 6.1: SGPR (left column), ODGPR (centre column) and SOLVE-GPR (right column) posteriors (mean $\pm$ 2 standard deviations) with a squared exponential kernel on Snelson and Gharahmani's dataset [4] for various numbers of inducing points. Top row: $M = 4$ / $(M_1, M_2) = (2, 2)$. Middle row: $M = 6$ / $(M_1, M_2) = (3, 3)$. Bottom row: $M = 8$ / $(M_1, M_2) = (4, 4)$. Inducing inputs are shown as black "plus" symbols; orthogonal inducing inputs are shown as black triangles.

posterior mean for ODGPR and as such, the posterior covariance can only be reduced near regular inducing points ("plus" symbols in the figure above). This effect becomes more visually prominent as the total number of inducing points increases since the predictions become progressively more confident near the regular inducing points, but not near the orthogonal inducing points, resulting in noticeable "pinch points" in the covariance. We will not discount ODGPR entirely, but because of ODGPR's poor covariance estimation capacity relative to SOLVE-GPR, we will use SOLVE-GPR as our benchmark orthogonal method in order to make the most compelling comparison with SGPR.

6.3 Equivalencies with SGPR

Now that we have shown that SOLVE-GPR is computationally cheaper than SGPR, it is natural to question whether there is some associated tradeoff in terms of the quality of the approximation. To help answer this question, we will compare the SGPR and SOLVE-GPR posteriors in detail to determine whether or not they are equivalent and if so, under which conditions. Although redundant, we will restate several formulas and expressions from previous chapters here to spare the reader having to flip back and forth.

6.3.1 Posterior Mean

We recall the formula for the SGPR posterior mean $\boldsymbol{\mu}_* = \mu(\mathbf{X}_*)$ at some test points $\mathbf{X}_*$:

$$\boldsymbol{\mu}_* = \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_\mathbf{u}, \tag{6.1}$$

where the optimal variational mean $\mathbf{m}_\mathbf{u}^*$ is given by:

$$\mathbf{m}_\mathbf{u}^* = \sigma^{-2}\mathbf{K}_{\mathbf{uu}}\left[\mathbf{K}_{\mathbf{uu}} + \sigma^{-2}\mathbf{K}_{\mathbf{uf}}\mathbf{K}_{\mathbf{fu}}\right]^{-1}\mathbf{K}_{\mathbf{uf}}\mathbf{y}. \tag{6.2}$$

We also recall the SOLVE-GPR posterior mean $\boldsymbol{\mu}_*$:

$$\boldsymbol{\mu}_* = \mathbf{K}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}_\mathbf{u} + \mathbf{C}_{*\mathbf{v}}\mathbf{C}_{\mathbf{vv}}^{-1}\mathbf{m}_\mathbf{v}, \tag{6.3}$$

where the optimal variational means $\mathbf{m}_\mathbf{u}^*$ and $\mathbf{m}_\mathbf{v}^*$ are given by:

$$\begin{aligned}
\mathbf{m}_\mathbf{u}^* &= \mathbf{K}_{\mathbf{uu}}\left[\mathbf{K}_{\mathbf{uu}} + \mathbf{K}_{\mathbf{uf}}\left(\mathbf{C}_{\mathbf{fv}}\mathbf{C}_{\mathbf{vv}}^{-1}\mathbf{C}_{\mathbf{vf}} + \sigma^2\mathbf{I}_\mathbf{N}\right)^{-1}\mathbf{K}_{\mathbf{fu}}\right]^{-1}\mathbf{K}_{\mathbf{uf}}\left(\mathbf{C}_{\mathbf{fv}}\mathbf{C}_{\mathbf{vv}}^{-1}\mathbf{C}_{\mathbf{vf}} + \sigma^2\mathbf{I}_\mathbf{N}\right)^{-1}\mathbf{y}, \\
\mathbf{m}_\mathbf{v}^* &= \mathbf{C}_{\mathbf{vv}}\left[\mathbf{C}_{\mathbf{vv}} + \mathbf{C}_{\mathbf{vf}}\left(\mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}} + \sigma^2\mathbf{I}_\mathbf{N}\right)^{-1}\mathbf{C}_{\mathbf{fv}}\right]^{-1}\mathbf{C}_{\mathbf{vf}}\left(\mathbf{K}_{\mathbf{fu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}} + \sigma^2\mathbf{I}_\mathbf{N}\right)^{-1}\mathbf{y}.
\end{aligned} \tag{6.4}$$

and the orthogonal covariance matrices $\mathbf{C}_{\mathbf{vf}}$ and $\mathbf{C}_{\mathbf{vv}}$ are given by:

$$\begin{aligned}
\mathbf{C}_{\mathbf{vf}} &= \mathbf{K}_{\mathbf{vf}} - \mathbf{K}_{\mathbf{vu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uf}}, \\
\mathbf{C}_{\mathbf{vv}} &= \mathbf{K}_{\mathbf{vv}} - \mathbf{K}_{\mathbf{vu}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{K}_{\mathbf{uv}}.
\end{aligned} \tag{6.5}$$

Let us consider some edge cases for SOLVE-GPR. If $\mathbf{u}$ is an empty set, i.e. we allocate all our inducing points to the orthogonal set $\mathbf{v}$, then the covariances $\mathbf{K}_{\mathbf{uf}} = \mathbf{K}_{\mathbf{uv}} = \mathbf{K}_{\mathbf{uu}} = 0$. As a result, the orthogonal covariances $\mathbf{C}_{\mathbf{vf}}$ and $\mathbf{C}_{\mathbf{vv}}$ become the real covariances $\mathbf{K}_{\mathbf{vf}}$ and $\mathbf{K}_{\mathbf{vv}}$, respectively. The posterior mean $\boldsymbol{\mu}_*$ becomes:

$$\boldsymbol{\mu}_* = \mathbf{K}_{*\mathbf{v}}\mathbf{K}_{\mathbf{vv}}^{-1}\mathbf{m}_\mathbf{v}. \tag{6.6}$$

Additionally, the optimal variational mean $\mathbf{m}_\mathbf{v}^*$ becomes:

$$\mathbf{m}_\mathbf{v}^* = \sigma^{-2}\mathbf{K}_{\mathbf{vv}}\left[\mathbf{K}_{\mathbf{vv}} + \sigma^{-2}\mathbf{K}_{\mathbf{vf}}\mathbf{K}_{\mathbf{fv}}\right]^{-1}\mathbf{K}_{\mathbf{vf}}\mathbf{y}. \tag{6.7}$$

Substituting eq. 6.7 into eq. 6.6, we perfectly recover the SGPR posterior mean. We obtain the same result in the case where $\mathbf{v}$ is an empty set, i.e. when we allocate all our inducing points to the regular set $\mathbf{u}$. Another noteworthy edge case for SOLVE-GPR is when the two sets of inducing points are equal, i.e. when $\mathbf{u} = \mathbf{v}$. In this case, the covariances $\mathbf{K}_{\mathbf{vf}} = \mathbf{K}_{\mathbf{uf}}$ and $\mathbf{K}_{\mathbf{uv}} = \mathbf{K}_{\mathbf{uu}}$. As a result, the orthogonal covariances $\mathbf{C}_{\mathbf{vf}} = \mathbf{0}$ and $\mathbf{C}_{\mathbf{vv}} = \mathbf{0}$, the optimal variational mean $\mathbf{m}_\mathbf{v}^* = \mathbf{0}$, and the optimal variational mean $\mathbf{m}_\mathbf{u}^*$ becomes equivalent to eq. 6.2. The posterior mean $\boldsymbol{\mu}_*$ consequently becomes equal to eq. 6.1 and we again recover the SGPR posterior mean.

Aside from these edge cases, it is difficult to rigorously compare the SOLVE-GPR and SGPR posterior means as the different forms of the optimal variational means $\mathbf{m_u}$ and $\mathbf{m_v}$ (which involve several additional inversions for SOLVE-GPR as compared with SGPR), make it challenging to express the means in a way which is easily comparable. What we *can* deduce is that the SOLVE-GPR posterior mean will be most similar to the SGPR posterior mean when the orthogonal inducing points behave as similarly as possible to regular inducing points, which happens when the orthogonal covariance matrices closely approximate real covariance matrices, i.e. $\mathbf{C_{vv}} \approx \mathbf{K_{vv}}$. Aside from the edge case when $\mathbf{u}$ is an empty set, another scenario in which this can occur is when $\mathbf{K_{uv}}$ is small, i.e. when there is little covariance between the two sets of inducing points. In the context of a squared-exponential kernel, which computes covariance in inverse proportion to the Euclidean distance between inputs, $\mathbf{K_{uv}}$ would be small when $\mathbf{u}$ and $\mathbf{v}$ are well-separated. We will elaborate on this point in the context of the posterior covariance, which is more easily comparable between SOLVE-GPR and SGPR.

6.3.2 Posterior Covariance

We recall the SGPR posterior covariance $\mathbf{\Sigma_{**}} = \Sigma(\mathbf{X_*}, \mathbf{X_*})$ at some test points $\mathbf{X_*}$:

$$\mathbf{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}}, \tag{6.8}$$

where the optimal variational covariance $\mathbf{S_u^*}$ is given by:

$$\mathbf{S_u^*} = \mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uu}}. \tag{6.9}$$

We also recall the SOLVE-GPR posterior covariance $\mathbf{\Sigma_{**}}$:

$$\mathbf{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}(\mathbf{K_{uu}} - \mathbf{S_u})\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} - \mathbf{C_{*v}}\mathbf{C_{vv}^{-1}}(\mathbf{C_{vv}} - \mathbf{S_v})\mathbf{C_{vv}^{-1}}\mathbf{C_{v*}}, \tag{6.10}$$

where the optimal variational covariances $\mathbf{S_u^*}$ and $\mathbf{S_v^*}$ are given by:

$$\begin{aligned}\mathbf{S_u^*} &= \mathbf{K_{uu}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{uu}},\\ \mathbf{S_v^*} &= \mathbf{C_{vv}}\left[\mathbf{C_{vv}} + \sigma^{-2}\mathbf{C_{vf}}\mathbf{C_{fv}}\right]^{-1}\mathbf{C_{vv}}.\end{aligned} \tag{6.11}$$

Let us consider the same edge cases we analysed for the posterior mean. If $\mathbf{u}$ is an empty set and consequently $\mathbf{K_{uf}} = \mathbf{K_{uv}} = \mathbf{K_{uu}} = 0$, the orthogonal covariances $\mathbf{C_{vf}}$ and $\mathbf{C_{vv}}$ become real covariances $\mathbf{K_{vf}}$ and $\mathbf{K_{vv}}$, and the posterior covariance becomes:

$$\mathbf{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*v}}\mathbf{K_{vv}^{-1}}(\mathbf{K_{vv}} - \mathbf{S_v})\mathbf{K_{vv}^{-1}}\mathbf{K_{v*}}. \tag{6.12}$$

Additionally, the optimal variational covariance $\mathbf{S_v^*}$ becomes:

$$\mathbf{S_v^*} = \mathbf{K_{vv}}\left[\mathbf{K_{vv}} + \sigma^{-2}\mathbf{K_{vf}}\mathbf{K_{fv}}\right]^{-1}\mathbf{K_{vv}}. \tag{6.13}$$

Substituting eq. 6.13 into eq. 6.12, we perfectly recover the SGPR posterior covariance. We obtain

the same result in the case where $\mathbf{v}$ is an empty set. For the additional edge case when $\mathbf{u} = \mathbf{v}$, the covariances $\mathbf{K_{vf}} = \mathbf{K_{uf}}$ and $\mathbf{K_{uv}} = \mathbf{K_{uu}}$ and consequently the orthogonal covariances $\mathbf{C_{vf}} = \mathbf{0}$ and $\mathbf{C_{vv}} = \mathbf{0}$. The posterior covariance becomes equivalent to eq. 6.8 and we again recover the SGPR posterior covariance. Having examined these edge cases for both the posterior mean and covariance, we can affirm that SOLVE-GPR perfectly recovers the SGPR posterior in the cases where 1) $\mathbf{u}$ is an empty set, 2) $\mathbf{v}$ is an empty set, or 3) $\mathbf{u} = \mathbf{v}$, i.e. the two sets of inducing points are perfectly overlapping. Next, we will explore posterior equivalencies in more common scenarios that fall outside of these edge cases.

Unlike the posterior mean, it is relatively straightforward to express the SGPR and SOLVE-GPR posterior covariances in a way that makes them easy to compare. To do this, we explicitly write out the SGPR posterior covariance in terms of concatenated states by partitioning the inducing points into two arbitrary subsets. We will use the redundant and somewhat confusing notation of referring to these subsets as $\mathbf{u}$ and $\mathbf{v}$ even though the superset which we partitioned is also $\mathbf{u}$, as it will make comparison with SOLVE-GPR clearer. We can write eq. 6.8 equivalently as:

$$\mathbf{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} + \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{S_u}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}}. \tag{6.14}$$

Substituting in the optimal variational covariance $\mathbf{S_u^*}$ from eq. 6.9, we obtain:

$$\begin{aligned}
&= \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} + \mathbf{K_{*u}}\cancel{\mathbf{K_{uu}^{-1}}\mathbf{K_{uu}}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\cancel{\mathbf{K_{uu}}\mathbf{K_{uu}^{-1}}}\mathbf{K_{u*}} \\
&= \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} + \mathbf{K_{*u}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{u*}} \\
&= \mathbf{K_{**}} + \mathbf{K_{*u}}\left(\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1} - \mathbf{K_{uu}^{-1}}\right)\mathbf{K_{u*}}.
\end{aligned} \tag{6.15}$$

Partitioning the inducing set $\mathbf{u}$ into two arbitrary subsets $\mathbf{u}$ and $\mathbf{v}$ (redundant notation), we can write the above in block form:

$$\begin{aligned}
&= \mathbf{K_{**}} + \begin{bmatrix}\mathbf{K_{*u}} & \mathbf{K_{*v}}\end{bmatrix}\left\{\left(\begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}} \\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix} + \sigma^{-2}\begin{bmatrix}\mathbf{K_{uf}} \\ \mathbf{K_{vf}}\end{bmatrix}\begin{bmatrix}\mathbf{K_{fu}} & \mathbf{K_{fv}}\end{bmatrix}\right)^{-1} - \begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}} \\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix}^{-1}\right\}\begin{bmatrix}\mathbf{K_{u*}} \\ \mathbf{K_{v*}}\end{bmatrix}, \\
&= \mathbf{K_{**}} + \begin{bmatrix}\mathbf{K_{*u}} & \mathbf{K_{*v}}\end{bmatrix}\left\{\left(\begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}} \\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix} + \sigma^{-2}\begin{bmatrix}\mathbf{K_{uf}}\mathbf{K_{fu}} & \mathbf{K_{uf}}\mathbf{K_{fv}} \\ \mathbf{K_{vf}}\mathbf{K_{fu}} & \mathbf{K_{vf}}\mathbf{K_{fv}}\end{bmatrix}\right)^{-1} - \begin{bmatrix}\mathbf{K_{uu}} & \mathbf{K_{uv}} \\ \mathbf{K_{vu}} & \mathbf{K_{vv}}\end{bmatrix}^{-1}\right\}\begin{bmatrix}\mathbf{K_{u*}} \\ \mathbf{K_{v*}}\end{bmatrix}.
\end{aligned} \tag{6.16}$$

Next, we take the SOLVE-GPR posterior covariance expressed as a summation and find an equivalent expression in block form. We begin by writing eq. 6.10 in an equivalent form:

$$\mathbf{\Sigma_{**}} = \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} + \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{S_u}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} - \mathbf{C_{*v}}\mathbf{C_{vv}^{-1}}\mathbf{C_{v*}} + \mathbf{C_{*v}}\mathbf{C_{vv}^{-1}}\mathbf{S_v}\mathbf{C_{vv}^{-1}}\mathbf{C_{v*}}. \tag{6.17}$$

Substituting in the optimal variational covariances $\mathbf{S_u^*}$ and $\mathbf{S_v^*}$ from eq. 6.11, we obtain:

$$\begin{aligned}
= \mathbf{K_{**}} &- \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} + \mathbf{K_{*u}}\cancel{\mathbf{K_{uu}^{-1}}\mathbf{K_{uu}}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\cancel{\mathbf{K_{uu}}\mathbf{K_{uu}^{-1}}}\mathbf{K_{u*}} \\
&- \mathbf{C_{*v}}\mathbf{C_{vv}^{-1}}\mathbf{C_{v*}} + \mathbf{C_{*v}}\cancel{\mathbf{C_{vv}^{-1}}\mathbf{C_{vv}}}\left[\mathbf{C_{vv}} + \sigma^{-2}\mathbf{C_{vf}}\mathbf{C_{fv}}\right]^{-1}\cancel{\mathbf{C_{vv}}\mathbf{C_{vv}^{-1}}}\mathbf{C_{v*}}.
\end{aligned}$$

$$
\begin{aligned}
&= \mathbf{K_{**}} - \mathbf{K_{*u}}\mathbf{K_{uu}^{-1}}\mathbf{K_{u*}} + \mathbf{K_{*u}}\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1}\mathbf{K_{u*}} - \mathbf{C_{*v}}\mathbf{C_{vv}^{-1}}\mathbf{C_{v*}} + \mathbf{C_{*v}}\left[\mathbf{C_{vv}} + \sigma^{-2}\mathbf{C_{vf}}\mathbf{C_{fv}}\right]^{-1}\mathbf{C_{v*}} \\
&= \mathbf{K_{**}} + \mathbf{K_{*u}}\left(\left[\mathbf{K_{uu}} + \sigma^{-2}\mathbf{K_{uf}}\mathbf{K_{fu}}\right]^{-1} - \mathbf{K_{uu}^{-1}}\right)\mathbf{K_{u*}} + \mathbf{C_{*v}}\left(\left[\mathbf{C_{vv}} + \sigma^{-2}\mathbf{C_{vf}}\mathbf{C_{fv}}\right]^{-1} - \mathbf{C_{vv}^{-1}}\right)\mathbf{C_{v*}}.
\end{aligned} \tag{6.18}
$$

We can write this summation in block form:

$$
= \mathbf{K_{**}} + \begin{bmatrix}\mathbf{K_{*u}} & \mathbf{C_{*v}}\end{bmatrix}\left\{\left(\begin{bmatrix}\mathbf{K_{uu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_{vv}}\end{bmatrix} + \sigma^{-2}\begin{bmatrix}\mathbf{K_{uf}}\mathbf{K_{fu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_{vf}}\mathbf{C_{fv}}\end{bmatrix}\right)^{-1} - \begin{bmatrix}\mathbf{K_{uu}} & \mathbf{0} \\ \mathbf{0} & \mathbf{C_{vv}}\end{bmatrix}^{-1}\right\}\begin{bmatrix}\mathbf{K_{u*}} \\ \mathbf{C_{v*}}\end{bmatrix}. \tag{6.19}
$$

Comparing the eq. 6.19 and eq. 6.16, we can see that the SOLVE-GPR posterior covariance is in general sparser than that of SGPR, due to the zero elements in the off-diagonals of the block matrices. Let us revisit the idea of "well-separated" sets of inducing points, i.e. if we place $\mathbf{u}$ and $\mathbf{v}$ in separate regions of the domain such that $\mathbf{K_{uv}} \approx \mathbf{0}$. As previously discussed, in such a case the orthogonal covariances $\mathbf{C_{vf}}$ and $\mathbf{C_{vv}}$ closely approximate the true covariances $\mathbf{K_{vf}}$ and $\mathbf{K_{vv}}$. Furthermore, the covariances between the inducing points and function values $\mathbf{K_{uf}}$ and $\mathbf{K_{vf}}$ would contain mostly zero elements in the columns corresponding to function values which are far away from the corresponding set of inducing points, and would have mirrored zero elements such that products $\mathbf{K_{uf}}\mathbf{K_{fv}}$ and $\mathbf{K_{vf}}\mathbf{K_{fu}}$ would approximately be zero matrices. In a simple case with N evenly-spaced training points, M_1 inducing points $\mathbf{u}$ allocated over one part of the function domain, denoted $\mathbf{f}_1$, and M_2 orthogonal inducing points $\mathbf{v}$ allocated over the remainder of the domain, denoted $\mathbf{f}_2$, we would roughly have:

$$
\mathbf{K_{uf}} = \begin{bmatrix}\mathbf{K_{uf}}_1 & \mathbf{0}\end{bmatrix}, \qquad \mathbf{K_{vf}} = \begin{bmatrix}\mathbf{0} & \mathbf{K_{vf}}_2\end{bmatrix}, \qquad \Rightarrow \mathbf{K_{uf}}\mathbf{K_{fv}} = \begin{bmatrix}\mathbf{K_{uf}}_1 & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{0} \\ \mathbf{K_{vf}}_2\end{bmatrix} = \mathbf{0}. \tag{6.20}
$$

If we again compare eq. 6.19 and eq. 6.16 and consider the case when $\mathbf{K_{uv}} = \mathbf{0}$ and therefore $\mathbf{C_{vf}} = \mathbf{K_{vf}}$, $\mathbf{C_{vv}} = \mathbf{K_{vv}}$ and $\mathbf{K_{vf}}\mathbf{K_{fu}} = \mathbf{0}$, we can see that the two posterior covariances are equivalent. Therefore, in addition to the edge cases considered previously, we conclude that SOLVE-GPR recovers the SGPR posterior covariance when $\mathbf{u}$ and $\mathbf{v}$ are well-separated. As this coincides with the case in which the orthogonal inducing points behave most similarly to regular inducing points (as deduced in the previous subsection), we conclude further that SOLVE-GPR most closely recovers the SGPR posterior in the case when $\mathbf{u}$ and $\mathbf{v}$ are well-separated, i.e. when $\mathbf{K_{uv}}$ is small, and diverges from SGPR the most when $\mathbf{u}$ and $\mathbf{v}$ are well-overlapping, i.e. when $\mathbf{K_{uv}}$ is large.

6.3.3 The Determining Factor: Inducing Point Separation

Let us summarise the equivalencies we have found between SOLVE-GPR and SGPR. SOLVE-GPR *fully* recovers the SGPR posterior in the following edge cases:

- $\mathbf{u}$ or $\mathbf{v}$ is an empty set, or
- $\mathbf{u} = \mathbf{v}$.

In addition, SOLVE-GPR fully recovers the SGPR posterior covariance and most closely recovers the posterior mean when $\mathbf{u}$ and $\mathbf{v}$ are well-separated and $\mathbf{K_{uv}} \approx \mathbf{0}$. Whether the posterior mean is fully recovered remains an open question which requires future work.

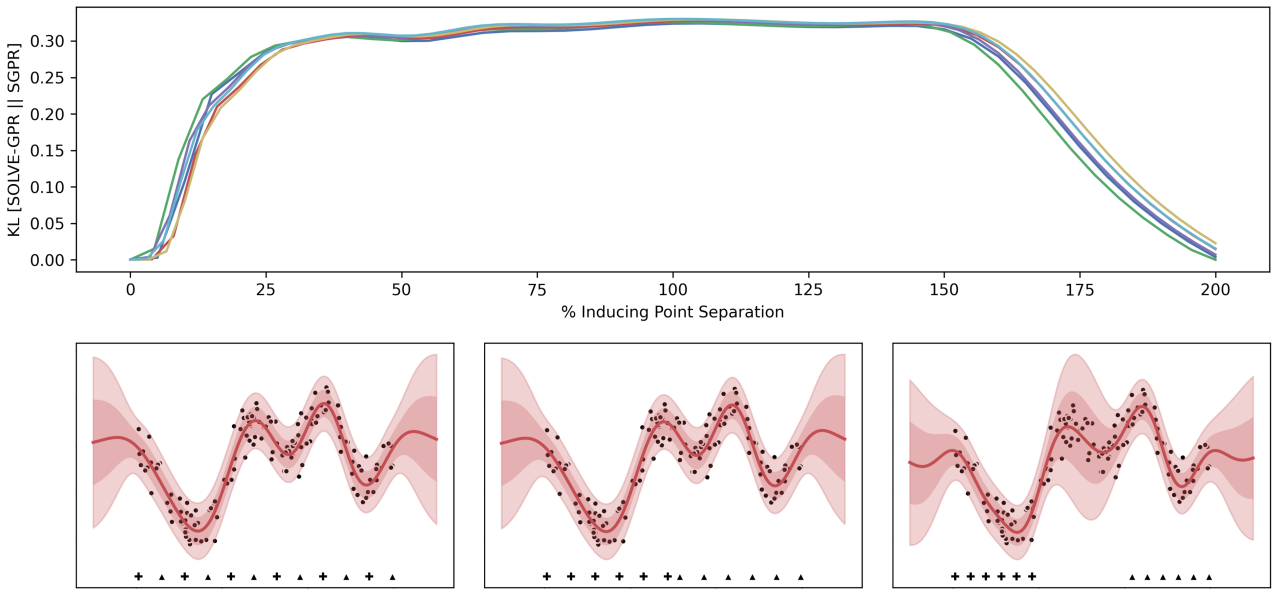

Figure 6.2: Top: KL divergence of the SOLVE-GPR posterior from the SGPR posterior with squared-exponential kernels vs. inducing point separation on Snelson and Gharahmani's dataset [4] for various total numbers of inducing points (multiple curves shown). The bottom row shows the progression of inducing point separation as we move from left to right. Bottom left: $\sim$ 0% separation. Bottom centre: $\sim$ 100% separation. Bottom right: $\sim$ 200% separation. Inducing inputs are shown as black "plus" symbols; orthogonal inducing inputs are shown as black triangles. The KL divergence is nearly zero when the inducing points are well-overlapping and well-separated.

We support our claim that the posterior equivalence is dependent on inducing point separation with a computational experiment, the results of which are shown in fig. 6.2. We begin by taking two evenly spaced and equally populated sets of inducing points which span the input domain of Snelson and Gharahmani's dataset [4] and which overlap perfectly. We then compute the KL divergence of the SOLVE-GPR posterior with the two sets from the SGPR posterior with the concatenation of the two sets, i.e. we compute $\mathrm{KL}[\text{SOLVE-GPR}(M, M)||\text{SGPR}(2M)]$. We then "slide" the two sets of inducing points apart and repeat this procedure until the two sets are very well-separated. As expected, the KL divergence is zero when the two sets of inducing points are perfectly overlapping, and (approximately) zero when the two sets are well-separated. When the inducing points are well-overlapping or partially overlapping, the KL divergence is non-zero.

6.4 Inducing Point Allocation

An important question from both a practical and theoretical standpoint is how to allocate regular and orthogonal inducing points in an optimal fashion. We know that from a computational complexity standpoint, the optimal implementation of SOLVE-GPR is when regular and orthogonal inducing points are allocated evenly, i.e. 50/50 splits. However, we hypothesise that since the orthogonal points contribute to the posterior using orthogonal covariances (e.g. $\mathbf{C_{vv}}$), they are in general less valuable than regular inducing points, having approximately equivalent value only in the edge case when $\mathbf{u}$ is an empty set or when $\mathbf{u}$ and $\mathbf{v}$ are well-separated.

We support this hypothesis with another computational experiment, the results of which are shown in fig. 6.3. We compute the KL divergence of the SOLVE-GPR posterior from the exact GP posterior for

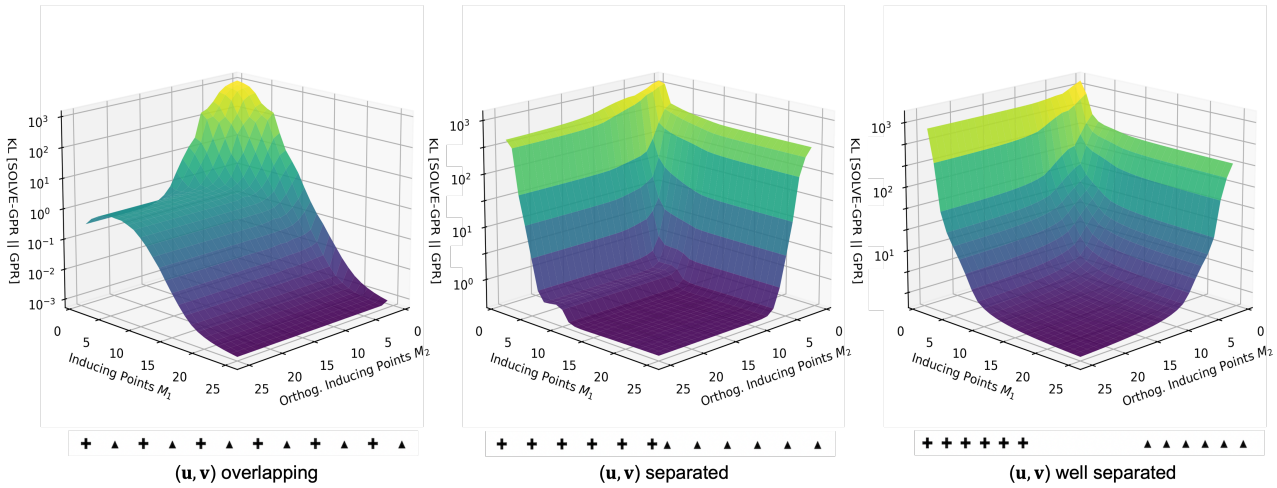

Figure 6.3: KL divergence of the SOLVE-GPR posterior from the exact GPR posterior vs. regular/orthogonal inducing point allocation on Snelson and Gharahmani's dataset [4] with a squared-exponential kernel. Left: regular and orthogonal inducing sets are strongly overlapping. Centre: regular and orthogonal inducing sets are just barely separated. Left: regular and orthogonal inducing sets are separated by a wide margin. Example inducing point allocations are shown at the bottom of each figure to illustrate approximate separation using black "plus" symbols and triangles. As the inducing sets become more separated, the plots become more symmetric.

various allocations of regular and orthogonal inducing points on Snelson and Gharahmani's dataset [4] in three distinct scenarios: overlapping inducing sets, separated inducing sets and well-separated inducing sets. Note that by overlapping we mean that the sets span the same general domain, not that the points are literally placed on top of one another. In the scenario where the inducing sets are overlapping, adding regular inducing points always improves the approximation, whereas adding orthogonal inducing points appears to have a finite capacity for improving the approximation. This scenario exhibits the most noticeable difference in behaviour of the regular and orthogonal inducing points, but also results in the lowest potential KL divergence from the true posterior. As we increase the inducing set separation, the behaviour of the orthogonal inducing points becomes increasingly similar to that of the regular inducing points, but the KL divergence from the true posterior increases.

6.5 When Orthogonal Methods Excel

We have arrived at a quandary: the scenario in which the orthogonal inducing points behave most similarly to regular inducing points (and therefore add the most value in terms of improving the approximation) is when the sets $\mathbf{u}$ and $\mathbf{v}$ are well separated. However, placing inducing variables in well separated clusters is not generally a good strategy for learning a dataset, as we will fail to capture information in the space between our inducing sets. Just because we can recover the SGPR posterior under certain conditions does not necessarily mean that the SGPR posterior under these conditions is worth recovering. The exception, and the scenario in which SOLVE-GPR really excels, is when we have a dataset consisting of two well-separated data clusters. In such a scenario, SOLVE-GPR is able to recover the SGPR posterior at a reduced computational cost, and the SGPR posterior, though sparse, is able to closely approximate the true posterior due to the nature of the data. We illustrate this through another computational experiment, the results of which are shown in fig.6.4.

Regression datasets containing well-separated data clusters are somewhat uncommon; in general,

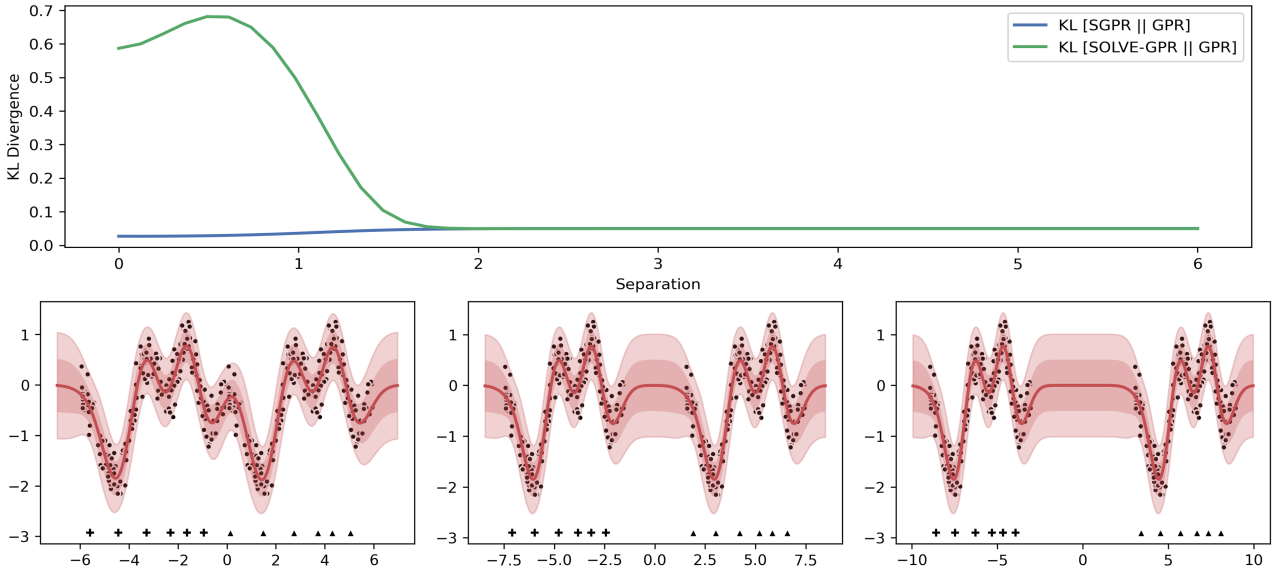

Figure 6.4: Top: KL divergence of the SOLVE-GPR posterior and SGPR posterior from the exact GPR posterior vs. data cluster separation on a duplicated Snelson and Gharahmani dataset [4] with a squared-exponential kernel. We see that as the data clusters become increasingly separated, SOLVE-GPR recovers SGPR and also makes a high quality posterior approximation. Bottom: progression of SOLVE-GPR predictions as the data cluster separation increases. Bottom left: separation = 0. Bottom centre: separation = 3. Bottom right: separation = 6. Inducing inputs are shown as black "plus" symbols; orthogonal inducing inputs are shown as black triangles.

although we recover SGPR the best when the inducing sets are well-separated, this may not be a posterior worth recovering. In practical settings, it may make sense to let the inducing sets overlap and enjoy the computational cost reduction of SOLVE-GPR with the tradeoff of a slightly less accurate posterior approximation.

Chapter 7

Conclusions and Future Work

In this thesis, we probed and benchmarked two major orthogonal sparse variational Gaussian process (SVGP) approximations, showed equivalencies between different parameterisations and demonstrated when orthogonal methods excel and when they do not.

We began by providing a brief introduction to probabilistic inference and Gaussian processes, followed by a thorough overview of the original sparse variational Gaussian process approximation as proposed by Titsias in [5] and [17] with a specific focus on collapsed bound regression (SGPR), going through many of his steps and calculations in thorough detail. Specifically, we showed showed how to derive the approximate posterior using Gaussian linear transformation rules, how to derive the variational lower bound by applying Bayes' theorem to the KL divergence of the variational posterior to the true posterior, then how to collapse the bound and find the optimal variational parameters using gradient methods. We then reviewed the work of Hensman and Matthews in deriving a stable and efficient implementation of sparse variational GP's for collapsed bound regression [18], [23]. Our aim in reviewing these topics in such depth was to provide the reader with a clear understanding of SVGP's before introducing more complex orthogonal variations, which were the main focus of this thesis.

Next, we introduced two notable orthogonal SVGP parameterisations: ODVGP [9] and SOLVE-GP [10]. We reviewed the posteriors and variational lower bound for these parameterisations as shown by their authors before making our first novel contributions: deriving a fully collapsed orthogonal bound and a stable implementation of the bound and predictive equations using Cholesky decompositions and other computational tricks. We finished by probing the behaviour and characteristics of orthogonal SVGP's by analysing the computational cost, examining equivalencies with SGPR and running a series of mathematically motivated computational experiments. Overall, we found that orthogonal SGPR methods like SOLVE-GPR can be computationally cheaper than SGPR, but the cost reduction generally comes at the expense of sparser approximations of the covariance. The exception is for datasets involving two well-separated data clusters, in which case SOLVE-GPR can approximate the true posterior nearly (if not exactly) as well as SGPR but at lower cost.

The three main analyses which remain to be conducted are 1) examining the SOLVE-GPR posterior mean with greater scrutiny to see if it may be written in a form that is directly comparable with SGPR to complete (or refute) the equivalence under the condition of well-separated inducing sets, 2) attempting to prove that adding an orthogonal inducing point can only improve the lower bound (similar analysis to [24] or [17]) and 3) analysing the costs and benefits of repeating the orthogonal decomposition of the Gaussian process again, or perhaps recursively. In [10], the authors claim that doing the decomposition a third time increases the complexity, but do not cite specific evidence.

In summation, orthogonal sparse variational Gaussian processes are a promising class of approxima-

tions which can improve the computational cost of doing collapsed bound regression with the general drawback of a sparser covariance. Empirically, the results yielded by SOLVE-GPR are very similar to those given by SGPR in many scenarios, but further analysis needs to be conducted to understand when SOLVE-GPR is or is not mathematically equivalent to SGPR, and to get a proper intuition for when a practitioner should choose to use one over the other.

References

[1] MacKay DJC. Information Theory, Inference and Learning Algorithms. Cambridge University Press; 2003. Available from: https://www.inference.org.uk/itprnn/book.pdf. pages 1, 4, 5, 10

[2] Csato L, Opper M. Sparse Online Gaussian Processes. Neural Computation. 2002;14:641–668. Available from: https://eprints.soton.ac.uk/259182/. pages 1

[3] Quiñonero-Candela J, Rasmussen CE. A Unifying View of Sparse Approximate Gaussian Process Regression. Journal of Machine Learning Research. 2005;6(65):1939–1959. Available from: http://jmlr.org/papers/v6/quinonero-candela05a.html. pages 1, 13

[4] Snelson E, Ghahramani Z. Sparse Gaussian Processes using Pseudo-inputs. In: Weiss Y, Schölkopf B, Platt J, editors. Advances in Neural Information Processing Systems. vol. 18. MIT Press; 2005. Available from: https://proceedings.neurips.cc/paper/2005/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf. pages 1, 13, 17, 28, 29, 40, 41, 46, 47, 48

[5] Titsias M. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In: Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics. vol. 5. PMLR; 2009. p. 567–574. Available from: https://proceedings.mlr.press/v5/titsias09a.html. pages 1, 13, 15, 18, 20, 21, 25, 32, 39, 49

[6] Hensman J, Fusi N, Lawrence ND. Gaussian Processes for Big Data. arXiv; 2013. Available from: https://arxiv.org/abs/1309.6835. pages 1, 13, 18

[7] Salimbeni H, Deisenroth M. Doubly Stochastic Variational Inference for Deep Gaussian Processes. arXiv; 2017. Available from: https://arxiv.org/abs/1705.08933. pages 1

[8] Shi J, Khan ME, Zhu J. Scalable Training of Inference Networks for Gaussian-Process Models. arXiv; 2019. Available from: https://arxiv.org/abs/1905.10969. pages 1

[9] Salimbeni H, Cheng CA, Boots B, Deisenroth M. Orthogonally Decoupled Variational Gaussian Processes. 2018. Available from: https://arxiv.org/abs/1809.08820. pages 1, 26, 27, 39, 49

[10] Shi J, Titsias MK, Mnih A. Sparse Orthogonal Variational Inference for Gaussian Processes. arXiv; 2019. Available from: https://arxiv.org/abs/1910.10596. pages 1, 26, 28, 31, 39, 49

[11] van der Wilk M. Sparse Gaussian Process Approximations and Applications. University of Cambridge; 2018. Available from: https://mvdw.uk/vanderwilk-thesis.pdf. pages 3, 10, 12, 13, 16

[12] Deisenroth MP, Faisal AA, Ong CS. Mathematics for Machine Learning. Cambridge University Press; 2020. Available from: https://mml-book.github.io/book/mml-book.pdf. pages 7

[13] Yi W. Sparse and Variational Gaussian Process (SVGP) — What To Do When Data is Large;. Available from: https://towardsdatascience.com/sparse-and-variational-gaussian-process-what-to-do-when-data-is-large-2d3959f430e7. pages 9

[14] Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB. Bayesian Data Analysis. 3rd ed. CRC Press; 2014. Available from: http://www.stat.columbia.edu/~gelman/book/BDA3.pdf. pages 9, 10

[15] Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. MIT Press; 2006. Available from: http://gaussianprocess.org/gpml/chapters/RW.pdf. pages 10, 13, 22

[16] Matthews A. Scalable Gaussian Process Inference Using Variational Methods. University of Cambridge; 2017. Available from: https://www.repository.cam.ac.uk/bitstream/handle/1810/278022/thesis.pdf?sequence=1&isAllowed=y. pages 10

[17] Titsias M. Variational Model Selection for Sparse Gaussian Process Regression; 2009. Available from: https://www2.aueb.gr/users/mtitsias/papers/sparseGPv2.pdf. pages 15, 20, 26, 31, 49

[18] Hensman J, Matthews A. Derivation of SGPR equations;. Available from: https://gpflow.readthedocs.io/en/v1.5.1-docs/notebooks/theory/SGPR_notes.html. pages 15, 21, 26, 33, 34, 49

[19] Hensman J, Durrande N, Solin A. Variational Fourier features for Gaussian processes. arXiv; 2016. Available from: https://arxiv.org/abs/1611.06740. pages 16

[20] Petersen KB, Pedersen MS. The Matrix Cookbook. Technical University of Denmark; 2008. Available from: http://www2.imm.dtu.dk/pubdb/p.php?3274. pages 19

[21] Block Matrix;. Available from: https://en.wikipedia.org/wiki/Block_matrix. pages 26

[22] Block LU Decomposition;. Available from: https://en.wikipedia.org/wiki/Block_LU_decomposition. pages 34

[23] Matthews AGdG, van der Wilk M, Nickson T, Fujii K, Boukouvalas A, León-Villagrá P, et al. GPflow: A Gaussian process library using TensorFlow. Journal of Machine Learning Research. 2017 apr;18(40):1–6. Available from: http://jmlr.org/papers/v18/16-537.html. pages 49

[24] Bauer M, van der Wilk M, Rasmussen CE. Understanding Probabilistic Sparse Gaussian Process Approximations. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R, editors. Advances in Neural Information Processing Systems. vol. 29. Curran Associates, Inc.; 2016. Available from: https://proceedings.neurips.cc/paper/2016/file/7250eb93b3c18cc9daa29cf58af7a004-Paper.pdf. pages 49