Imperial College London

MEng Individual Project

Imperial College London

Department of Computing

Heteroscedastic Inducing Point Selection for Sparse Gaussian Process Classification

Author:
Giovanni Passerello

Supervisor:
Dr. Mark van der Wilk

Second Marker:
Dr. Julia Ive

June 21, 2021

Abstract

Gaussian processes are a class of rich non-parametric models in the realm of probabilistic machine learning. They are expressed as probability distributions over functions, which are able to exploit the Bayesian framework in order to perform accurate inference and quantify their own predictive uncertainty.

Exact Gaussian process inference scales cubically with the number of training examples, N. As a consequence, practical inference often requires the use of sparse variational approximations which consider only a small set of M inducing points that accurately summarise the data. Performing inference through these approximations presents two problems: firstly we need to choose the inducing points of interest, and secondly we need to compute a variational distribution over these inducing points.

Gaussian process regression benefits from a conjugate model which permits the optimal variational distribution to be computed in closed-form, and further provides ground for mathematical analysis of suitable inducing point selection methods. However, due to the non-conjugate nature of the classification framework, no such analytical solution is available and gradient-based optimisation of the variational distribution is required; this is both expensive and unreliable.

In this work, we first explore the benefits of a heteroscedastic model and recommend an augmented inducing point selection method for Gaussian process regression. We then propose a novel heteroscedastic model for Gaussian process classification in which the variational distribution is analytically optimisable. In particular, we use Pólya-Gamma data augmentation to form a conditionally conjugate lower bound to the classification likelihood. With this analytical solution available, we further propose a novel inducing point selection method for Gaussian process classification.

We evaluate our methods against the current state-of-the-art, and show empirical improvements on ten datasets from the OpenML and UCI repositories. Notably, we show that our methods permit much sparser and more robust inference, where the number of inducing points required for a sufficiently accurate approximation can be automatically determined by the model.

Acknowledgements

I would like to principally thank my supervisor, Dr. Mark van der Wilk, for his guidance and support throughout the duration of this project. His constant enthusiasm and remarkable knowledge of the field made the development of this project a richly rewarding and enjoyable experience - it has been a privilege to work with him.

I would also like to thank my second marker, Dr. Julia Ive, for her invaluable feedback during the development of this report.

Finally, I would like to thank my late father Chris for his inspirational scholarship, my mother Louise for her unwavering support throughout my educational journey, and my wider network of friends and family around the world.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of machine learning (ML) encompasses a class of algorithms which aim to automatically learn patterns from data for purposes of prediction and/or decision-making. These algorithms are typically used where a hand-programmed solution might be undesirable or infeasible, and often involve finding some statistical model that accurately captures given data. For example, self-driving cars employ image recognition techniques to 'understand' their environment and then make real-time decisions based on this understanding.

One of the most prevalent issues with many modern methods is that they operate as black boxes, i.e., they provide little vision over their internal workings. With the growing adoption of ML in all sectors of industry, it is essential that we have a clear understanding of these systems and are able to reason about the decisions that they make. For these purposes, we believe that it is desirable to be able to quantify the uncertainty (or degree of belief) in both our predictions and in our chosen model.

Bayesian inference provides a natural framework to address all aspects of these uncertainties through the use of probability theory, and as surveyed by [Ghahramani (1; 2)], allows us to account for uncertainty during both prediction and model selection[1]. These uncertainties often arise due to limited model complexity or lack of training data, in which cases we may be unable to find the 'perfect' underlying model as a single point estimate. The Bayesian framework provides methodologies to account for this by constructing models as probability distributions which are updated through Bayes' rule in the presence of data[2]. In this setting, we are able to 'average' over *all* possible model parameters through integration, instead of relying on unsatisfactory point estimates - this remediates issues such as overfitting.

1.1 Gaussian Processes

Gaussian processes (GPs) [Rasmussen and Williams (5)] are probability distributions over functions which are used to represent priors and posteriors for use in Bayesian inference. They are a class of stochastic processes that exploit the Bayesian framework and can be seen as non-parametric models with infinitely-many basis functions,

[1] Bayesian inference automatically applies the principle of 'Occam's Razor', i.e., it avoids the selection of overly-complex models [Rasmussen and Ghahramani (3)].

[2] For a comprehensive overview of probability theory and inference methods, see [MacKay (4)].

or as neural network layers with infinitely-many neurons [Neal (6)]. Aside from their capacity to represent flexible models, one of their key attractions is their ability to accurately represent their own predictive uncertainty. These traits have resulted in GPs gaining a large amount of recent attention in parts of the ML community.

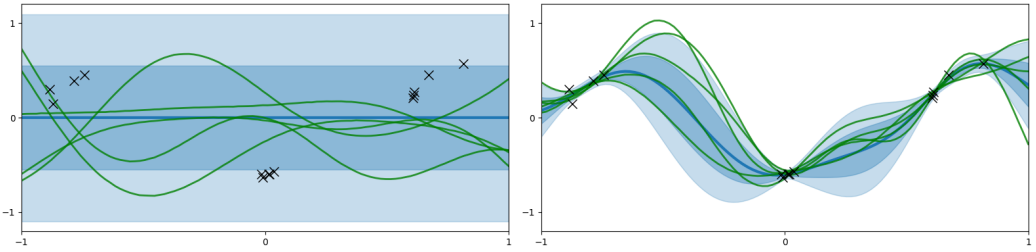

Figure 1.1: Gaussian processes as distributions over functions. Multiple function samples drawn from the prior (left) and posterior (right) distributions shown in green. Mean function shown in blue, with σ and 2σ predictive uncertainty bounds surrounding.

The largest limitation of GPs however, is that they require processing the entire training dataset at prediction time. Exact inference involves computing matrix inverses and determinants which scale cubically, $\mathcal{O}(N^3)$, with respect to the size of the training dataset, N. For this reason, it is well known that GPs are computationally intractable in the presence of even modestly sized datasets ($N \geq 10,000$). In order to maintain the benefits of Bayesian uncertainty and flexible non-parametric models, practical inference often requires the use of sparse variational approximations.

Sparse approximations involve finding some subset of quantities ('inducing points') that best explain the posterior that we are trying to model. These inducing points can either be function values evaluated at training inputs [Quiñnonero-Candela and Rasmussen (7), Titsias (8)], variable pseudo-inputs [Snelson and Ghahramani (9)] or more abstract representations. In order to perform inference, we must then compute a variational distribution over the chosen inducing points. We typically choose M inducing points, for which the cost of inference more favourably scales as $\mathcal{O}(NM^2)$. [Burt et al. (10)] place bounds on how M should grow with respect to N in order to maintain accurate approximations and show that we can typically choose $M << N$.

1.2 Objectives

The development of this project aims to address three interdependent objectives encompassing the use of Gaussian process models in practice:

- **Heteroscedastic Regression**: It is a realistic expectation that a large amount of real-word data is non-stationary, and that the amount of noise may vary across the domain, i.e., the data may be heteroscedastic. For example, measuring the distance of a spacecraft during takeoff may become more noisy as it gets further away due to atmospheric distortions. We would therefore like to incorporate heteroscedasticity into our Gaussian process regression models.

- **Analytical Classification**: As Gaussian process regression contains a conjugate Gaussian likelihood, a closed-form solution for the optimal variational distribution exists [Titsias (8)], and provides ground for the theoretical justification of inducing point selection methods [Burt et al. (10)]. However, as Gaussian

process classification contains a non-conjugate Bernoulli likelihood, no such analytical solution exists - this means that the variational distribution must be computed through gradient-based optimisation which is both expensive and unreliable. We would therefore like to find a closed-form approximation to the optimal variational distribution for Gaussian process classification.

- **Inducing Point Selection for Classification**: With the existence of a closed-form solution for Gaussian process classification, we would like to further devise a suitable inducing point selection method. For Gaussian process regression, inducing points are often chosen to spread out across the input domain so as to capture a broad range of the function well. However, we hypothesise that instead clustering the inducing points around decision boundaries may lead to increased predictive performance for Gaussian process classification.

1.3 Contributions

In an attempt to address the above objectives, we list our main contributions:

- **Heteroscedastic Regression**: We extend the work of [Titsias (8)] and derive a heteroscedastic Gaussian process regression model in which the variational distribution is analytically optimised. We further provide justifications for a novel heteroscedastic inducing point selection method, and show that our method removes the need for gradient-based optimisation of the inducing points.

- **Analytical Classification**: We address the current gap in the literature, and derive a novel heteroscedastic Gaussian process classification model, in which the variational distribution over inducing points is analytically optimised. In this model, we employ Pólya-Gamma data augmentation to form a conditionally conjugate lower bound to the likelihood. Importantly, the effective likelihood is an unnormalised Gaussian distribution with heteroscedastic variance.

- **Inducing Point Selection for Classification**: With an analytical solution for classification available, we justify the use of the same novel heteroscedastic inducing point selection method for classification. We show that our methods are able to consistently outperform the current state-of-the-art, permitting much sparser and more robust inference. Crucially, our selection method allows the number of inducing points required to be automatically determined by the model. We further show that, as per our hypothesis, the selected inducing points for classification tend to cluster around the decision boundaries.

In order to both theoretically and empirically motivate the aforementioned contributions, we provide an extensive suite of mathematical derivations and experimental results in the project appendices. We also provide robust and efficient software implementations of our contributions in a public GitHub repository, available *here*.

1.4 Report Structure

In Chapter 1 we introduced and motivated the topic of our work. In Chapter 2 we move on to cover the preliminary concepts for the development of this project, and perform an analysis of the past and current literature along the way. Of particular

note, we cover Gaussian processes, approximate inference, Gaussian process classification, Pólya-Gamma data augmentation and inducing point selection. In Chapters 3 and 4 we explore heteroscedastic Gaussian process regression and classification respectively and establish the majority of our theoretical contributions.

Chapter 5 extends this theory with a thorough empirical investigation, evaluating our contributions against the current state-of-the-art and exploring both qualitative and quantitative aspects of our methods. Chapter 6 provides a brief discussion of ethical considerations and societal issues relevant to the project. Finally, Chapter 7 concludes the report, briefly summarising our findings as well as providing recommendations for promising directions of future research.

We also provide a significant proportion of our theoretical contributions in the appendices, which are referenced accordingly throughout the body of the report.

Chapter 2

Background

In this chapter, we summarise the prerequisite knowledge for the development of this project. As part of this, we provide a review of the past and current literature aiming to compare and contrast the relevant strengths and weaknesses of current methodologies. Our main focus in this review is to draw meaningful insight which can guide the direction of work in later chapters.

2.1 Gaussian Processes

Gaussian processes (GPs) [Rasmussen and Williams (5)] are probability distributions over functions which can be used to represent priors and posteriors in Bayesian inference. The direct placement of a distribution over functions yields many interesting observations and useful properties for inference:

- GPs represent a class of flexible non-parametric models. More specifically, with a squared exponential covariance function they can be shown to correspond to Bayesian linear regression with an infinite number of basis functions. They can also be seen as neural network layers with infinitely-many neurons [Neal (6)].

- We can specify a GP as a finite collection of random variables, any finite subset of which are jointly Gaussian distributed. The Kolmogorov extension theorem tells us that this collection implies a stochastic process[1], and that we are able to fully specify the process by considering only a finite number of points.

- The distribution over function values is Gaussian and is therefore analytically tractable. This is particularly beneficial as we can find solutions in closed-form.

One of the most prevalent benefits of GPs is that they are able to accurately represent their own uncertainty and provide confidence intervals on their predictions. Due to their non-parametricity, these uncertainty estimates are able to be maintained even in regions unconstrained by data. On the other hand, one of their largest limitations is that computation of exact inference scales cubically as $\mathcal{O}(N^3)$, with respect to the size of the training dataset, N. Recent developments in approximate methods have reduced the cost of inference to $\mathcal{O}(NM^2)$ by selecting a set of M 'inducing points' used for inference, where typically $M << N$ for significant computational reduction.

[1] For a comprehensive overview, see [Matthews (11)].

2.1.1 Notation

In the literature, Gaussian processes are often discussed as distributions over scalar-valued functions[2], i.e., $f : \mathcal{X} \to \mathbb{R}$. GPs are fully specified by a mean function[3], $m(\cdot)$, and a covariance function, $k(\cdot, \cdot')$. The mean function determines the means of the marginal Gaussian distributions, and the covariance function specifies a measure of similarity between the function values at two inputs. We denote a GP distribution as

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot')) \tag{2.1}$$

As mentioned previously, we can in fact specify a GP by a collection of random variables in which any finite subset are jointly Gaussian distributed. These random variables represent the scalar function values, $f(\boldsymbol{x})$ (or f), at input locations, $\boldsymbol{x} \in \mathbb{R}^D$. The Kolmogorov extension theorem tells us that under the assumption of marginally consistent random variables[4], this collection defines a stochastic process. We can thus completely define the GP by considering only a finite set of points of interest.

Before discussing inference procedures we must first introduce the Gaussian process prior. The prior is a multivariate Gaussian distribution over function values, whose characteristics are determined by properties of the covariance kernel. These random variables are represented by vectors $\boldsymbol{f}$ and $\boldsymbol{f}^*$ which are the function values evaluated at training and test inputs $\boldsymbol{X}$ and $\boldsymbol{X}^*$ respectively, and $[\boldsymbol{K}_{f*}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j^*)$

$$\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{ff} & \boldsymbol{K}_{f*} \\ \boldsymbol{K}_{*f} & \boldsymbol{K}_{**} \end{bmatrix}\right) \tag{2.2}$$

2.2 Exact Inference

Assume we have a dataset of inputs, $\boldsymbol{X}$, and corresponding noisy observations, $\boldsymbol{y} = f(\boldsymbol{X}) + \boldsymbol{\epsilon}$, where $y_n = f(\boldsymbol{x}_n) + \epsilon_n$ and $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$. We also define some arbitrary set of test points, $\boldsymbol{X}^*$, which we wish to make predictions at. Defining the joint distribution over observations, $\boldsymbol{y}$, and test outputs, $\boldsymbol{f}^*$, under the prior we have

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f}^* \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{K}_{ff} + \sigma^2 \boldsymbol{I} & \boldsymbol{K}_{f*} \\ \boldsymbol{K}_{*f} & \boldsymbol{K}_{**} \end{bmatrix}\right) \tag{2.3}$$

By conditioning the joint distribution on the observations using the Gaussian conditioning rule, we can compute the form of the predictive posterior distribution

$$p(\boldsymbol{f}^* \mid \boldsymbol{y}) = \mathcal{GP}(\boldsymbol{K}_{*f}[\boldsymbol{K}_{ff} + \sigma^2 \boldsymbol{I}]^{-1}\boldsymbol{y}, \boldsymbol{K}_{**} - \boldsymbol{K}_{*f}[\boldsymbol{K}_{ff} + \sigma^2 \boldsymbol{I}]^{-1}\boldsymbol{K}_{f*}) \tag{2.4}$$

and we can compute the marginal likelihood by marginalisation of the observations

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{0}, \boldsymbol{K}_{ff} + \sigma^2 \boldsymbol{I}) \tag{2.5}$$

Under a Gaussian likelihood $p(y_n | f_n) = \mathcal{N}(y_n; f_n, \sigma^2)$, both the predictive posterior distribution and the marginal likelihood have closed-form solutions. However, due to the calculation of the inverse and determinant of $\boldsymbol{K}_{ff}$ (which scale as $\mathcal{O}(N^3)$), the equations quickly become computationally intractable. Additionally, for non-Gaussian likelihoods no closed-form solutions exist for these quantities due to the lack of conjugacy - we discuss this in detail in Section 2.5.

[2] For a framework for interdomain and multioutput GPs, see [van der Wilk et al. (12)].

[3] Without loss of generality, the prior mean function is usually taken to be zero for convenience.

[4] This assumption holds for any positive-definite covariance function.

2.3 Approximate Inference

In order to ameliorate the above issue of computational intractability, a large number of approximate methods have been proposed in the recent literature. As mentioned earlier, many of these methods involve performing inference on some smaller set of M inducing points. Early methods were predominantly proposed as approximations to the model wherein exact inference is performed on an approximate prior, however some of these methods undesirably forfeit the non-parametricity of the model[5].

As previously discussed, one of the core benefits of Gaussian processes is their ability to maintain uncertainty estimates in regions unconstrained by data. [van der Wilk (13)] discussed desiderata for approximate methods and in particular placed emphasis on the importance of maintaining uncertainty estimates through the use of non-parametric models. For this reason, we favour methods which instead approximate the posterior distribution and retain the benefits of a non-parametric model.

2.3.1 Approximate Posteriors

We saw earlier that the posterior of the exact GP is computationally intractable, however by searching a restricted class of tractable posteriors we can hope to find a distribution that accurately approximates it. There are two prominent classes of methods for these approximations: expectation propagation (EP) [Minka (14), Bui et al. (15)] and variational inference (VI) [Blei et al. (16), Matthews (11)]. Both of these frameworks vary in how they select the approximate posterior and there is currently no consensus on which is definitively better than the other. [van der Wilk et al. (12)] briefly discuss their differences, stating that whilst each have their own merits, VI tends to be the more predictable of the two whereas EP can sometimes provide an approximation that significantly differs from the exact posterior. For the purposes of this report, we focus on the variational inference framework.

2.3.2 Class of Tractable Posteriors

Before introducing variational inference, it is important to discuss the class of tractable posterior distributions previously mentioned. This class of tractable posteriors should contain distributions that are mathematically convenient to manipulate and should be broad enough to sufficiently approximate the exact posterior. In particular, we choose the class of tractable posteriors as introduced by [Titsias (8)].

In order to form a sparse approximation, we first choose M inducing inputs, $\boldsymbol{Z}$, with corresponding outputs, $f(\boldsymbol{Z}) = \boldsymbol{u}$, to approximate the full posterior Gaussian process. We need not restrict the distribution over inducing points to be a posterior, but can instead define it to be any Gaussian by specifying a free mean and variance

$$q(\boldsymbol{u}) \;=\; \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}, \boldsymbol{S}) \tag{2.6}$$

We then define a distribution over a set of finite-dimensional marginals by using the Gaussian process prior conditioned on $\boldsymbol{u}$ and the approximate distribution over $\boldsymbol{u}$. Finally, by the Kolmogorov extension theorem we know that this finite collection implies the full posterior Gaussian process

[5] For a review of model-based approximations, see [Quiñonero-Candela and Rasmussen (7)].

$$
\begin{aligned}
q(\boldsymbol{f},\boldsymbol{u}) &= p(\boldsymbol{f}\,|\,\boldsymbol{u})q(\boldsymbol{u}) \\
&= \mathcal{N}(\boldsymbol{f};\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{u},\ \boldsymbol{K}_{\boldsymbol{ff}}-\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}})\mathcal{N}(\boldsymbol{u};\boldsymbol{m},\boldsymbol{S}) \qquad (2.7)\\
q(\boldsymbol{f}) &= \mathcal{N}(\boldsymbol{f};\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m},\ \boldsymbol{K}_{\boldsymbol{ff}}-\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{K}_{\boldsymbol{uu}}-\boldsymbol{S})\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}) \qquad (2.8)\\
\implies f(\cdot) &\sim \mathcal{GP}(\boldsymbol{k}_{\cdot\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m},\ \boldsymbol{k}(\cdot,\cdot)-\boldsymbol{k}_{\cdot\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}(\boldsymbol{K}_{\boldsymbol{uu}}-\boldsymbol{S})\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{k}_{\boldsymbol{u}\cdot}) \qquad (2.9)
\end{aligned}
$$

where $[\boldsymbol{k}_{\boldsymbol{u}\cdot}]_i = k(\boldsymbol{z}_i,\cdot)$, $[\boldsymbol{k}_{\cdot\boldsymbol{u}}]_j = k(\cdot,\boldsymbol{z}_j)$ and $[\boldsymbol{K}_{\boldsymbol{uu}}]_{ij} = k(\boldsymbol{z}_i,\boldsymbol{z}_j)$. This class of approximate posteriors is computationally tractable for $M << N$ as the inversion of $\boldsymbol{K}_{\boldsymbol{uu}}$ only costs $\mathcal{O}(M^3)$, and we can traverse the class by adjusting $\boldsymbol{m}$ and $\boldsymbol{S}$. [van der Wilk et al. (12)] provide an alternative view of the class, by considering it in a regression setting to be 'the collection of all the posteriors that we can get from observing M function values through an arbitrary Gaussian likelihood'.

From these formulations we see that if a Gaussian likelihood is used, as $M \to N$ the exact posterior can be obtained to within an arbitrarily high precision; however in order for the posterior to remain tractable, we wish for $M << N$. [Burt et al. (10)] show that as $N \to \infty$, the rate at which M should grow to maintain an accurate approximation of the exact GP depends on the expected eigenvalue decay of $\boldsymbol{K}_{\boldsymbol{ff}}$.

With this class of tractable posteriors defined, we are now able to approximate the exact GP whilst maintaining the benefits of a non-parametric model. This affords us flexible models with accurate uncertainty estimates, whilst significantly reducing computational costs. As previously discussed, this would not be achievable by direct approximation of the model.

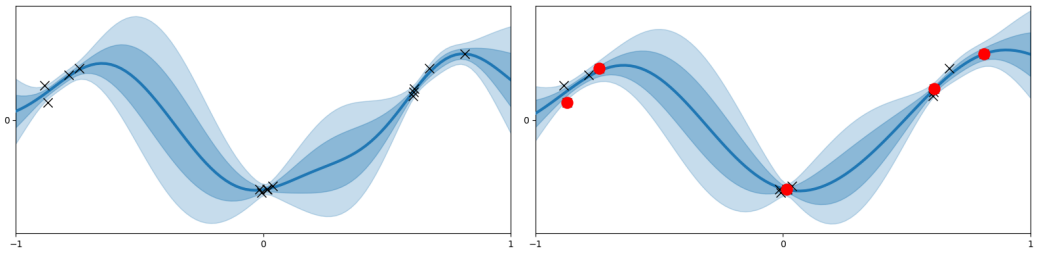

Figure 2.1: Sparse variational Gaussian process approximation. Full posterior (left) vs. sparse approximation with inducing points taken as a subset of the training data (right). Mean function shown in blue, with σ and 2σ predictive uncertainty bounds surrounding.

2.4 Variational Inference

Whilst for many years Markov chain Monte Carlo (MCMC) sampling methods [Hastings (17)] have dominated the field of Bayesian inference, variation inference (VI) has gained recent attention due to its superior efficiency and scalability. Recent research in this area has exposed the merits and shortcomings of VI with the aim to further catalyse development of the field [Blei et al. (16), Matthews (11)]. The main benefit of VI over MCMC is that it turns the inference problem into an optimisation problem which can be more efficiently solved; whereas MCMC must approximate the posterior with many samples from a Markov chain, VI directly optimises over the class of tractable posteriors which we previously defined.

Variational inference works by minimising the Kullback-Leibler (KL) divergence between an approximate posterior, $q(f(\cdot))$, and the exact posterior, $p(f(\cdot)|\boldsymbol{y})$. The KL

divergence represents a measure of difference between two probability distributions, and in the case of our two posterior distributions, is defined as

$$\mathcal{KL}[q(f(\cdot)) \,\|\, p(f(\cdot)|\boldsymbol{y})] \;=\; \mathbb{E}_{q(f(\cdot))}\left[\log \frac{q(f(\cdot))}{p(f(\cdot)|\boldsymbol{y})}\right] \tag{2.10}$$

Clearly we cannot directly evaluate this divergence as it contains exactly the quantity that we cannot compute - the exact posterior. Instead, we can implicitly minimise the KL divergence by maximising a lower bound to the log marginal likelihood[6], $\mathcal{L}$, which has the KL as its gap

$$\log p(\boldsymbol{y}) \;=\; \mathcal{L} + \mathcal{KL}[q(f(\cdot)) \,\|\, p(f(\cdot)|\boldsymbol{y})] \tag{2.11}$$

As the KL divergence is non-negative and the log marginal likelihood is constant, by maximising $\mathcal{L}$ we can minimise the KL divergence; in doing so, we find an approximate posterior that more closely represents the exact posterior.

2.4.1 The Lower Bound

To derive this bound, we must first observe that the KL divergence between the two posterior GPs is equivalent to the KL divergence between a finite marginal posterior distribution on the function values [Matthews et al. (18), van der Wilk et al. (12)]

$$\mathcal{KL}[q(f(\cdot)) \,\|\, p(f(\cdot)|\boldsymbol{y})] = \mathcal{KL}[q(f(\boldsymbol{X}), \boldsymbol{u}) \,\|\, p(f(\boldsymbol{X}), \boldsymbol{u}|\boldsymbol{y})] \tag{2.12}$$

By substituting (2.12) into (2.11) and rearranging for $\mathcal{L}$, we have

$$\begin{aligned}
\mathcal{L} &= \log p(\boldsymbol{y}) - \mathcal{KL}[q(\boldsymbol{f}, \boldsymbol{u}) \,\|\, p(\boldsymbol{f}, \boldsymbol{u}|\boldsymbol{y})] \\
&= \log p(\boldsymbol{y}) - \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\log \frac{q(\boldsymbol{f}, \boldsymbol{u})}{p(\boldsymbol{f}, \boldsymbol{u}|\boldsymbol{y})}\right] \\
&= \log p(\boldsymbol{y}) - \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\log \frac{p(\boldsymbol{f}|\boldsymbol{u})\, q(\boldsymbol{u})}{p(\boldsymbol{f}, \boldsymbol{u}|\boldsymbol{y})}\right] \\
&= \log p(\boldsymbol{y}) - \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\log \frac{p(\boldsymbol{f}|\boldsymbol{u})\, q(\boldsymbol{u})\, p(\boldsymbol{y})}{p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f}|\boldsymbol{u})\, p(\boldsymbol{u})}\right] \\
&= \log p(\boldsymbol{y}) - \mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\log \frac{q(\boldsymbol{u})\, p(\boldsymbol{y})}{p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{u})}\right] \\
&= -\mathbb{E}_{q(\boldsymbol{f},\boldsymbol{u})}\left[\log \frac{q(\boldsymbol{u})}{p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{u})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{f})}[\log p(\boldsymbol{y}|\boldsymbol{f})] - \mathbb{E}_{q(\boldsymbol{u})}\left[\log \frac{q(\boldsymbol{u})}{p(\boldsymbol{u})}\right] \\
\mathcal{L} &= \Sigma_{n=1}^{N} \mathbb{E}_{q(f_n)}[\log p(y_n|f_n)] - \mathcal{KL}[q(\boldsymbol{u}) \,\|\, p(\boldsymbol{u})]
\end{aligned} \tag{2.13}$$

[6]This lower bound is known as the 'Evidence Lower Bound' or 'ELBO'.

for any factorising likelihood that depends only on the Gaussian process at locations $\boldsymbol{X}$. It is important to note that this bound is analytically tractable for Gaussian likelihoods, however it is intractable for non-conjugate likelihoods as we shall later see.

The bound comprises two terms, the expected log-likelihood and the KL divergence between the variational distribution and prior over inducing points. The expected log-likelihood is effectively a measure of data fit, and the KL divergence is a regulariser which penalises the variational distribution for moving too far from the prior. The KL divergence is available in closed-form as $q(\boldsymbol{u})$ and $p(\boldsymbol{u})$ are Gaussian, however the log-likelihood summation can be computationally costly and hinder scalability.

2.4.2 Optimisation

In order to compute the optimal variational distribution over inducing points, gradient-based optimisation over the parameters $\boldsymbol{m}$ and $\boldsymbol{S}$ is often used. When M grows large, this optimisation can be expensive and unreliable - methods either take a prohibitively large amount of time to converge or fail to converge altogether.

In order to address the scalability issues of optimising $q(\boldsymbol{u})$ and computing the log-likelihood summation, [Hensman et al. (19)] derived a method which enables stochastic optimisation of the bound through mini-batch subsampling, however this can be noisy and slow to converge. Alternatively, [Titsias (8)] derived a collapsed bound in which the variational distribution is analytically optimised

$$\mathcal{L} = \log \mathcal{N}(\boldsymbol{y}; \boldsymbol{0}, \sigma^2 \boldsymbol{I} + \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}) - \frac{1}{2\sigma^2} Tr(\boldsymbol{K}_{ff} - \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}) \tag{2.14}$$

In this collapsed bound, the regularising trace term penalises the difference between the exact covariance matrix and the Nyström approximation, and is normalised by the Gaussian likelihood variance, σ^2. When this trace term is zero, the Nyström approximation is exact and we can say that the inducing points are sufficient statistics.

Crucially, in this formulation $q(\boldsymbol{u})$ is analytically optimised and there is no need for gradient-based optimisation over $\boldsymbol{m}$ and $\boldsymbol{S}$; the optimal variational distribution is

$$q^*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}^*, \boldsymbol{S}^*) = \mathcal{N}(\boldsymbol{u}; \sigma^{-2}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}, \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}) \tag{2.15}$$

where $\Sigma = \boldsymbol{K}_{uu} + \sigma^{-2}\boldsymbol{K}_{uf}\boldsymbol{K}_{fu}$. In the current literature the collapsed bound generally outperforms [Hensman et al. (19)] due to the instability of stochastic optimisation, however unfortunately it is only available for the conjugate GP regression framework. GP classification is forced to use stochastic optimisation due to its non-conjugate likelihood - we aim to ameliorate this issue in later chapters.

2.5 Classification

Classification problems involve assigning inputs to one (or multiple) of C classes, $\mathcal{C}_1, \ldots, \mathcal{C}_C$. The simplest example of a classification task is binary classification, where each input is assigned either the value 0 or 1; we place sole focus on the binary case in this work and do not cover the multi-class case. Whereas in previous sections we have discussed Gaussian processes for regression, they can also be used to solve classification tasks as function approximation problems [Rasmussen and Williams (5)].

2.5.1 The Model

In order to set up GP classification, we first place a GP prior on the latent function values, $\boldsymbol{f}$. We then 'squash' the Gaussian process through a sigmoidal inverse-link function to clamp its range between 0 and 1, and finally use a Bernoulli likelihood to condition the data on the values of the modified Gaussian process. In this work we place sole focus on the logit link function as the inverse-link of choice, $\sigma(z) = (1 + \exp(-z))^{-1}$. We define the logit GP classification model as

$$\begin{aligned} p(\boldsymbol{y}, \boldsymbol{f}) &= p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f}) \\ p(\boldsymbol{y}|\boldsymbol{f}) &= \prod_{n=1}^{N} \sigma(y_n f(\boldsymbol{x}_n)) \end{aligned}$$

where $p(\boldsymbol{y}|\boldsymbol{f})$ is the likelihood and $p(\boldsymbol{f})$ is the GP prior.

Whereas for regression we used a conjugate Gaussian likelihood, integration over a non-conjugate classification likelihood is unfortunately analytically intractable. We therefore cannot naively use our variational approximation of the ELBO as the expected log-likelihood terms are analytically intractable.

2.5.2 Practical Inference

Many solutions for approximating posteriors and marginal likelihoods with non-conjugate likelihoods have been proposed in the literature. [Hensman et al. (20)] provide a comprehensive review and show that the framework of variational inference can provide recourse. In fact, it turns out that we can use a slightly modified evaluation of the *exact same* variational lower bound which permits the use of stochastic optimisation. Revisiting the evidence lower bound we have

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(f)}[\log p(\boldsymbol{y}\,|\,\boldsymbol{f})] - \mathcal{KL}[q(\boldsymbol{u}) \,\|\, p(\boldsymbol{u})] \\ &= \Sigma_{n=1}^{N} \mathbb{E}_{q(f_n)}[\log p(y_n\,|\,f_n)] - \mathcal{KL}[q(\boldsymbol{u}) \,\|\, p(\boldsymbol{u})] \end{aligned}$$

where the expected log-likelihood is now analytically intractable. In the case of classification, we know that this likelihood is factorising and so we can convert the N-dimensional integral from the joint expectation, into a sum over N 1-dimensional integrals. Whilst these 1-dimensional integrals are still analytically intractable, we are able to employ Gauss-Hermite quadrature or Monte Carlo methods to approximate them and recover the bound!

It is somewhat surprising that after so many years of complex solutions (as reviewed by [Hensman et al. (20)]), it turns out that we only need this single variational lower bound to solve both regression and classification tasks - the only difference is in the computation of the expected log-likelihood. Finally we have a variational inference framework for both conjugate and non-conjugate factorising likelihoods, however there are still two prominent issues for the case of GP classification:

- We are beholden to the accuracy and computational complexity of the expected log-likelihood approximations (Gauss-Hermite quadrature or Monte Carlo).

- We know that gradient-based optimisation can be very expensive and unreliable, especially in the presence of large datasets. This problem was addressed for regression by the analytical solution of [Titsias (8)] given in equation 2.14, however it still remains an open problem for GP classification.

2.6 Pólya-Gamma

Due to the non-conjugate form of the logit GP classification model, Bayesian inference is a difficult problem to solve. In order to address this, [Polson et al. (21)] introduced the Pólya-Gamma distribution of random variables and proposed a corresponding data augmentation strategy for inference in models with a binomial likelihood. This scheme presents the possibility of an augmented classification model which is conditionally conjugate and thus permits efficient closed-form computations and analysis. In the following section we detail only the properties which hold particular relevance to our work - for a full exposition see [Polson et al. (21)].

2.6.1 The Distribution

The Pólya-Gamma family of distributions, $PG(b,c)$, is a subset of the class of infinite convolutions of gamma distributions. Of particular interest to us is the subset of random variables, $\omega \sim PG(b,0),\ b>0$, defined by the moment generating function

$$\mathbb{E}_{PG(\omega|b,0)}[\exp(-\omega t)] = \cosh^{-b}(\sqrt{t/2}) \tag{2.16}$$

The reason this is of interest to us is because we are able to write the logit link in a form which contains the *cosh* function[7]

$$\sigma(z) = \frac{\exp(\frac{1}{2}z)}{2\cosh(\frac{1}{2}z)} \tag{2.17}$$

Looking at the probability density function of the more general $PG(b,c)$ class, we see that it is simply an exponential tilting of the $PG(b,0)$ density

$$PG(\omega|b,c) \ \propto\ \exp\left(-\frac{c^2}{2}\omega\right) PG(\omega|b,0) \tag{2.18}$$

An appealing property of this Pólya-Gamma distribution is that all finite moments are available in closed-form. This allows us to compute the first moment, which will be extremely useful in later sections

$$\mathbb{E}_{PG(\omega|b,c)}[\omega] = \frac{b}{2c}\tanh\left(\frac{c}{2}\right) \tag{2.19}$$

2.6.2 Data Augmentation

In the original proposition, [Polson et al. (21)] present a data augmentation strategy for an efficient Gibbs sampler. [Wenzel et al. (22)] make further connections to the GP classification framework by noting that the logit link can further be written as

$$\sigma(z_n) = \frac{\exp(\frac{1}{2}z_n)}{2\cosh(\frac{1}{2}z_n)} = \frac{1}{2}\int \exp\left(\frac{z_n}{2} - \frac{z_n^2}{2}\omega_n\right) p(\omega_n)\, d\omega_n \tag{2.20}$$

[7] See Appendix A.1 for a derivation.

with the prior $p(\omega_n) = PG(\omega_n|1,0)$. By taking $z_n = y_n f(\boldsymbol{x}_n)$, they present the augmented joint density which we notice to be conditionally conjugate

$$p(\boldsymbol{y},\boldsymbol{\omega},\boldsymbol{f}) \propto \exp\left(\frac{1}{2}\boldsymbol{y}^\top \boldsymbol{f} - \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{\Omega} \boldsymbol{f}\right) p(\boldsymbol{f})\, p(\boldsymbol{\omega}) \tag{2.21}$$

where $\boldsymbol{\Omega} = \text{diag}(\boldsymbol{\omega})$ is the diagonal matrix of Pólya-Gamma random variables, ω_n. They further augment with the inducing variables, $\boldsymbol{u}$, to give the final model

$$p(\boldsymbol{y},\boldsymbol{\omega},\boldsymbol{f},\boldsymbol{u}) = p(\boldsymbol{y}|\boldsymbol{\omega},\boldsymbol{f})\, p(\boldsymbol{\omega})\, p(\boldsymbol{f}|\boldsymbol{u})\, p(\boldsymbol{u}) \tag{2.22}$$

2.6.3 Variational Approximation

With the formation of this augmented model, [Wenzel et al. (22)] go on to propose a stochastic variational inference algorithm. In this algorithm we aim to find a variational distribution over both $\boldsymbol{u}$ and $\boldsymbol{\omega}$. It turns out that we do not have to use the full Pólya-Gamma class $PG(\boldsymbol{\omega}|\boldsymbol{b},\boldsymbol{c})$ for the variational distribution over $\boldsymbol{\omega}$. Instead we can consider only the restricted class $q(\boldsymbol{\omega}) = PG(\boldsymbol{\omega}|\boldsymbol{1},\boldsymbol{c})$, which contains the optimal distribution and gives us a special form of the first moment from equation 2.19.

Of particular interest is the form of the variational lower bound which they derive

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})q(\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{\omega},\boldsymbol{f})] - \mathcal{KL}[q(\boldsymbol{u},\boldsymbol{\omega}) \,\|\, p(\boldsymbol{u},\boldsymbol{\omega})] \\
&= \frac{1}{2}(\log|\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}| - Tr(\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}) - \boldsymbol{m}^\top \boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m} + M - 2N\log 2 + \\
&\boldsymbol{y}^\top \boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m} - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) - Tr(\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}) - \\
&\boldsymbol{m}^\top \boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m} + \Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})])
\end{aligned} \tag{2.23}$$

where $\theta_n = \mathbb{E}_{q(\omega_n)}[\omega_n] = \frac{1}{2c_n}\tanh(\frac{c_n}{2})$, $\boldsymbol{\Theta} = \text{diag}(\boldsymbol{\theta})$ and $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u};\boldsymbol{m},\boldsymbol{S})$.

Note that due to the conditionally conjugate augmented model, this bound does not require an approximation to the expected log-likelihood which previously hindered scalability in Section 2.5.2. However, this method does still require gradient-based optimisation of the variational parameters which is the most expensive component.

The stochastic variational inference algorithm involves alternately updating the local variational parameters, $\boldsymbol{c}$, and the global variational parameters, $\boldsymbol{m}$ and $\boldsymbol{S}$. To update the local parameters we employ coordinate ascent and find an analytical solution for the unique maximum with the global parameters held fixed. We then use gradient-based optimisation to update the global parameters along with the kernel hyperparameters. In order to stabilise and speed up convergence, a mini-batched natural gradient scheme is used to approximate block-coordinate ascent updates. For a full derivation of the bound and inference algorithm, see [Wenzel et al. (22)].

The main issue remains that gradient-based optimisation of the variational parameters can be very expensive and unreliable, even with natural gradients. In later sections, we devise a novel method in which $q(\boldsymbol{u})$ is analytically optimised as in equation 2.14, removing the need for gradient-based optimisation of $q(\boldsymbol{u})$ entirely.

2.7 Inducing Point Selection

2.7.1 Gradient-Based Methods

Many methods in the early literature propose a gradient-based joint optimisation of the kernel hyperparameters[8] and inducing inputs [Snelson and Ghahramani (9), Titsias (8)]. In this scheme, the inducing inputs are often initialised by either uniform subsampling or k-means clustering, before being optimised through gradient-based methods. There are three main problems with this approach:

- There are few guarantees that can be made about the quality of the initialised inducing point locations. We often expect the initialisation to be suboptimal, and gradient-based optimisation consequently suffers from poor local optima.
- As M grows large, the convergence of gradient-based methods grows prohibitively slow. Paired with the optimisation of the variational parameters, the problem quickly becomes computationally intractable.
- Without an exhaustive search, we have little knowledge about how large M should be in order to guarantee a suitable quality of approximation - we can easily end up using far too few or too many inducing points.

2.7.2 Greedy Variance Method

The unreliability and computational cost of gradient-based methods has led to recent work which aims to remove gradient-based optimisation of the inducing points. It turns out that if we use the collapsed bound of [Titsias (8)] for regression, we are able to remove gradient-based optimisation of not only the variational parameters, but also the inducing points. This bound permits mathematical analysis which allows us to devise better inducing point initialisation schemes that guarantee an arbitrarily good approximation without the need for gradient-based optimisation. We are left only with optimising the kernel hyperparameters which is comparatively cheap.

[Burt et al. (10)] compare and contrast prominent selection methods, and devise a novel inducing point selection method known as 'greedy variance' selection. This method works by greedily selecting each inducing point in turn with the highest marginal variance in the conditioned prior, $p(\boldsymbol{f}|\boldsymbol{u})$, i.e. $argmax\ diag[\boldsymbol{K}_{ff} - \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}]$. Intuitively, a larger diagonal error of the Nyström approximation from the exact covariance matrix signals that we should place an inducing input at this location to improve the approximation. After greedy selection of the inducing points, the method then finds a new set of optimal hyperparameters[9] and alternates between these two phases in a pseudo variational Expectation-Maximisation (EM) algorithm[10]. This algorithm runs until the ELBO reaches convergence, at which point we have optimised the lower bound and thus our approximate posterior.

[8] These hyperparameters are such as the lengthscale and output variance of the squared exponential kernel. Their purpose is to determine the characteristics of the prior.

[9] The optimal hyperparameters can be found through gradient-based optimisation of the ELBO as discussed in [Burt et al. (10)].

[10] For this reason, the procedure is referred to as 'greedy variance reinitialisation'.

The authors discuss the connection of this selection scheme to the approximate sampling of a k-DPP [Kulesza and Taskar (23)], and show that this encourages the dispersal of inducing points. As discussed earlier, this behaviour is beneficial for regression tasks as the dispersion helps to capture a broad range of the function well.

Not only does this method provide guarantees on the suitability of the inducing points, but it also provides us with a means for automatically selecting how large M should be. The authors give results showing that the KL divergence between the approximate and exact posterior is upper-bounded by $Tr[\boldsymbol{K}_{ff} - \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}]$. Therefore, once this trace term grows sufficiently small we know that our approximation is close to exact, and we can simply stop accepting new inducing points. This automatic selection process provides great benefit as we no longer need to manually decide how large M should be - our model tells us when it is sufficiently accurate!

With this method, we have addressed all three issues of the previous gradient-based methods for regression through the introduction of this sophisticated selection algorithm. However, this is still an open problem for GP classification as there is no such collapsed bound. We later present a collapsed bound for GP classification, and propose an augmented inducing point initialisation scheme based on greedy variance.

Chapter 3

Heteroscedastic Regression

In the previous chapter we discussed GP regression with a homoscedastic likelihood, $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{f}, \sigma^2\boldsymbol{I})$. In this chapter we discuss GP regression with a heteroscedastic likelihood, $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{f}, \boldsymbol{\Lambda})$, where $\boldsymbol{\Lambda}$ is a diagonal matrix containing the marginal variances. We place particular focus on extending the literature by providing derivations and augmentations of the homoscedastic counterparts discussed in the previous chapter. This chapter is an essential stepping stone to later explore heteroscedastic classification, which is our core focus.

We stress that one of the main benefits of a collapsed model is the fact that $q(\boldsymbol{u})$ is analytically optimised, which entirely avoids the need for gradient-based optimisation of the variational distribution. The existence of an analytical solution also permits theoretical analysis of inducing point initialisation schemes which further remove the need for gradient-based optimisation of the inducing inputs. We begin this chapter with a brief exposition of a collapsed model for heteroscedastic GP regression, and then move on to justify a suitable inducing point initialisation scheme.

3.1 Variational Inference

3.1.1 The Lower Bound

The variational lower bound for heteroscedastic GP regression was presented in [Titsias (8)], from which we provide the collapsed bound for reference[1]

$$\mathcal{L} \;=\; \log \mathcal{N}(\boldsymbol{y};\, \boldsymbol{0}, \boldsymbol{\Lambda} + \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}) - \frac{1}{2} Tr(\boldsymbol{\Lambda}^{-1}(\boldsymbol{K}_{ff} - \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf})) \tag{3.1}$$

where we write $\boldsymbol{Q}_{ff} = \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}$ and $\tilde{\boldsymbol{K}}_{ff} = \boldsymbol{K}_{ff} - \boldsymbol{Q}_{ff}$ for convenience, giving

$$\mathcal{L} \;=\; \log \mathcal{N}(\boldsymbol{y};\, \boldsymbol{0}, \boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}) - \frac{1}{2} Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) \tag{3.2}$$

The corresponding optimal variational distribution which achieves this bound is

$$q^*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}\,;\, \boldsymbol{m}^*,\, \boldsymbol{S}^*) = \mathcal{N}(\boldsymbol{u}\,;\, \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y}\,,\, \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}) \tag{3.3}$$

where $\Sigma = \boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu}$. It turns out that this distribution is exactly the same optimal distribution found in [Snelson and Ghahramani (9)], but the difference between the schemes is that our ELBO is augmented by a regularising trace term.

[1] We refer the reader to the aforementioned work for a thorough derivation. This can be found in the supplementary appendices of the technical paper.

3.1.2 The Upper Bound

The variational upper bound for heteroscedastic GP regression has not been presented before, but can be found by augmenting the variational upper bound for homoscedastic GP regression as presented in [Titsias (24)][2]

$$\mathcal{U} = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| - \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff} + Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y} \tag{3.4}$$

3.1.3 Implementation

Our first software contribution is an efficient implementation of the lower bound and predictive distribution which use Cholesky decompositions and Woodbury inverses, computable in $\mathcal{O}(NM^2)$. The derivations can be found in Appendix B.1. We also provide a GPflow-based [Matthews et al. (25)] implementation, available *here*.

With this implementation in place it is easy to fit a heteroscedastic model, however there is still the problem of deciding how to compute the heteroscedastic marginal variances. The 'best' solution to this would be to fit the variance using another Gaussian process, however this obviously adds computational expense to the method and is difficult unless the training variance is known. As this task is not a core focus of this report, we simply define a polynomial likelihood with learnable parameters. We test our method on a toy dataset with quadratic noise in order to demonstrate the superiority of a heteroscedastic model in the presence of heteroscedastic data.

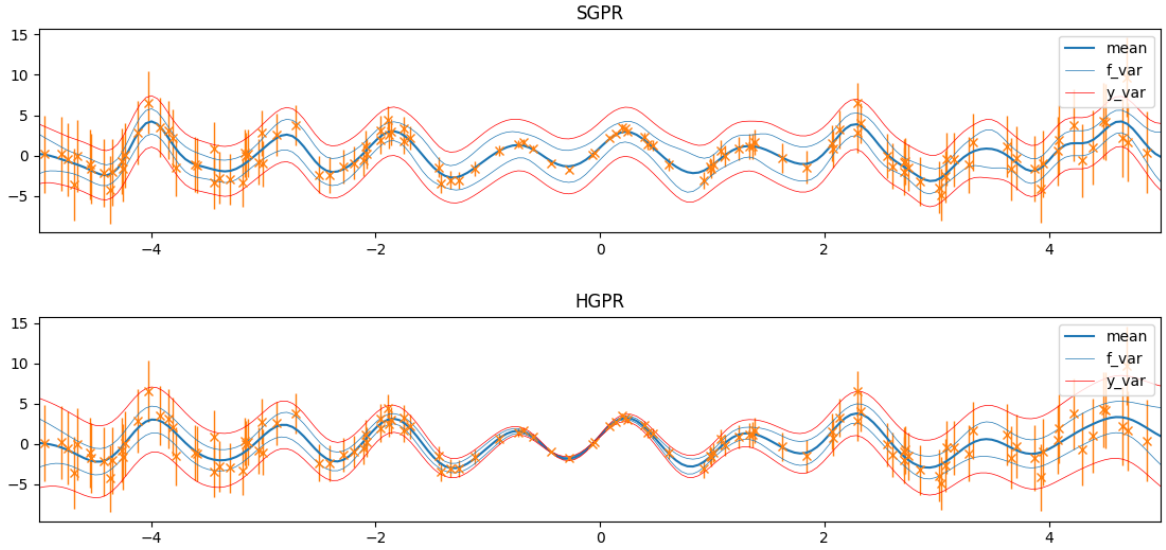

Figure 3.1: Homoscedastic (SGPR) vs heteroscedastic (HGPR) Gaussian processes with optimal $q(\boldsymbol{u})$ from the respective collapsed bounds. Data variance magnitudes shown by vertical orange bars, increasing quadratically from zero. Mean function shown in blue, with 2σ predictive uncertainty bars for both $\boldsymbol{f}$ and $\boldsymbol{y}$ surrounding.

We see from the above graphic that the heteroscedastic model fits the data extremely well, and is able to capture the heteroscedasticity in the data. The homoscedastic model however is forced to maintain wider uncertainty bounds across the domain due to its homoscedastic likelihood variance and penalisation from the ELBO.

[2] See Appendix B.2 for a full derivation.

We leave the problem of fitting the marginal variances unsolved and do not go into further empirical analysis of the regression model. Instead, we move to theoretical analysis of the heteroscedastic model and focus on the development of an augmented inducing point selection scheme. The reason we do this is because our main focus is the heteroscedastic classification model, which will later prescribe the optimal marginal variances for us in closed-form!

3.2 Bounds on the KL Divergence

We know that the difference between the true log marginal likelihood and the ELBO is equal to the KL divergence between the exact and approximate posterior, i.e.

$$\log p(\boldsymbol{y}) - \mathcal{L} = \mathcal{KL}[q(f(\boldsymbol{X}), \boldsymbol{u}) \,\|\, p(f(\boldsymbol{X}), \boldsymbol{u} \,|\, \boldsymbol{y})] \tag{3.5}$$

Therefore if we decrease the difference between the true log marginal likelihood and the ELBO, naturally we decrease the KL divergence and improve our approximation. However, due to intractability of the marginal likelihood we cannot directly compute this difference. In the following section we consider bounds on the KL divergence and use these to incentivise an inducing point selection strategy that aims to jointly minimise the bounds and thus minimise the KL divergence.

3.2.1 A-Posteriori Upper Bound

We begin by considering bounds on the KL divergence which can be computed for a specific dataset and approximation, e.g. a-posteriori bounds. The simplest a-posteriori upper bound to the KL divergence that we might consider is the difference between our upper and lower bounds to the log marginal likelihood

$$\begin{aligned}
\log p(\boldsymbol{y}) - \mathcal{L} &\leq \mathcal{U} - \mathcal{L} \qquad (3.6)\\
&= -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| - \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff} + Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y} + \\
&\frac{N}{2}\log 2\pi + \frac{1}{2}\log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| + \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y} + \frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) \\
&= \frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) + \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y} - \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff} + Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y}
\end{aligned}$$

From this bound we see that the KL divergence decreases as the two trace terms jointly decrease - both the first term goes to zero and the inverse terms cancel as $Tr(\tilde{\boldsymbol{K}}_{ff}) \to 0$. Clearly one suitable inducing point selection scheme would therefore be to use the standard greedy variance method which minimises $Tr(\tilde{\boldsymbol{K}}_{ff})$. However, we conjecture that we are able to use the heteroscedasticity to better select points.

Our goal is to choose the inducing point which decreases the above bound by the largest amount at each step. If we directly used greedy variance then there is the possibility that our selected point has a large likelihood variance, in which case the reduction in the bound from $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$ would be negligible. We could instead consider an augmented version of the greedy variance criterion, where we additionally account for $\boldsymbol{\Lambda}^{-1}$. By taking this heteroscedastic information into account, we avoid choosing a point with large variance and therefore guarantee a substantial reduction in the bound at every step. In other words, the heteroscedastic knowledge prevents us from selecting poor inducing points with high variance. With this in mind, we proceed to explore additional justifications that are able reinforce this conjecture.

3.2.2 Average Case Prior Bounds

[Burt et al. (10)] provide bounds for the average case prior model, where $\boldsymbol{X}$ is known a-posteriori and assumptions are made about $\boldsymbol{y}$ a-priori. In this section we follow their work and make adaptations for the heteroscedastic framework.

We take the training labels, training inputs and inducing inputs as $\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{Z}$ respectively and y, X, Z the corresponding random variables. In the average case it is natural to make distributional assumptions on $y|X$, and the natural candidate is the prior distribution, so we take $y|X \sim \mathcal{N}(y; \boldsymbol{0}, \boldsymbol{K}_{ff} + \boldsymbol{\Lambda})$. Under the assumption that the distributions of $y|X$ and $Z|X$ are independent then we additionally have that $y|X, Z \sim \mathcal{N}(y; \boldsymbol{0}, \boldsymbol{K}_{ff} + \boldsymbol{\Lambda})$. We can then derive bounds on the conditional expectation of the KL divergence conditioned on both the training inputs and inducing inputs[3]

$$\frac{t}{2} \leq \mathbb{E}[\mathcal{KL}[q(f(\boldsymbol{X}), \boldsymbol{u}) \,\|\, p(f(\boldsymbol{X}), \boldsymbol{u}|\boldsymbol{y})]|Z = \boldsymbol{Z}, X = \boldsymbol{X}] \leq t \tag{3.7}$$

with $t = Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$. In the average case, this implies that the KL divergence is entirely bound by the augmented trace term we have so often seen. These bounds can be shown to be tighter than our previous result, and they specifically bound from both above and below giving a more robust result. [Burt et al. (10)] state that when optimising the location of the inducing points using gradient-based methods, we may in fact expect the KL divergence to be somewhat smaller in practice than the average case lower bound. The important point to note however is that both bounds tell us that we can decrease the KL divergence by minimizing $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$.

3.3 Inducing Point Selection

In the previous section we considered two different derivations to bound the KL divergence between the exact and approximate posterior distributions. Both of these results showed us that by explicitly minimizing the term $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$, we can implicitly minimize the KL divergence. This now gives us a strong justification to develop a heteroscedastic inducing point selection scheme.

3.3.1 Monotonic Bound Increase

Before we devise an inducing point selection scheme, it is critically important to discuss one key aspect. In order for any chosen scheme to be both reliable and interpretable, we wish for the inclusion of additional inducing points to never decrease the ELBO, and thus to never worsen the approximation. To be explicit, we want the ELBO to be monotonically increasing with the addition of new inducing points[4].

The homoscedastic collapsed model of [Titsias (8)] has been shown to display this property, however other models such as [Snelson and Ghahramani (9)] do not. In Appendix B.4 we show that the heteroscedastic collapsed lower bound does indeed increase monotonically with the inclusion of additional inducing points.

[3] See Appendix B.3 for a full derivation.

[4] This additionally implies that the KL divergence will be monotonically decreasing.

3.3.2 Heteroscedastic Greedy Variance

With the knowledge that minimizing $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$ leads to the direct minimization of the KL divergence, we propose an augmented inducing point initialisation scheme for heteroscedastic GP regression - 'heteroscedastic greedy variance'. This scheme is similar to the standard greedy variance method, however the selection criterion is augmented by the heteroscedastic likelihood variance. One key justification for the use of this method over the standard greedy variance is that the augmented criterion encourages the selection of inducing points with a low marginal variance. This is an attractive trait as the placement of a point in a low variance region will naturally be more informative than the placement of a point in a noisy region.

Our method works by iteratively minimizing $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$ through the selection of an incremental inducing set. For each index $1...M$ we greedily select the inducing point corresponding to $argmax\ diag[\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}]$ and add this to our inducing set. In practice we implement this as a rank-1 update cycle to avoid costly repetitive computations. The entire optimisation cycle then consists of alternately optimizing the kernel hyperparameters and then reinitialising the inducing points using heteroscedastic greedy variance until convergence of the ELBO - this can be seen as a pseudo variational Expectation-Maximisation (EM) algorithm.

As discussed in Section 2.7.2, due to the iterative nature of this algorithm we are able to automatically select how large M should be. As our bounds showed that we want to minimize $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$, we can simply set a threshold and stop accepting new inducing points once the trace term is sufficiently small.

We do not go into detail on an empirical analysis of heteroscedastic regression as there is still the problem of deciding how to fit the likelihood variance. We instead defer empirical analysis until we reach the heteroscedastic classification case, where the optimal marginal variances will be explicitly computed for us in closed-form.

3.4 Summary

At the end of this chapter, we have extended the literature with derivations and findings for the heteroscedastic Gaussian process regression model. Here we summarise the main contributions:

- We gave an upper and lower variational bound for heteroscedastic GP regression in which the variational distribution was analytically optimised.
- We derived efficient closed-form implementations of the variational lower bound and predictive distribution, and provided a GPflow-based implementation in a public GitHub repository, available *here*.
- We derived bounds on the KL divergence between the exact and approximate posterior distributions. We further showed that these bounds jointly justified the minimization of $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$ as a means to minimize the KL divergence.
- We discussed requirements for a reliable inducing point selection scheme and proposed an augmented selection scheme called 'heteroscedastic greedy variance', which directly minimizes $Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$. Additionally, we found that this method allows the model to automatically select how large M should be.

Chapter 4

Heteroscedastic Classification

In Chapter 3 we presented a model for heteroscedastic GP regression in which the variational distribution over inducing points was analytically optimised. This model provided many benefits, such as removing the need for gradient-based optimisation of $q(\boldsymbol{u})$, and permitting mathematical analysis of suitable inducing point initialisation schemes. Unfortunately, due to the non-conjugate form of the logit GP classification model no such analytical solution is readily available for classification.

In this chapter we propose a novel model for GP classification in which Pólya-Gamma data augmentation [Polson et al. (21)] is used to form a conditionally conjugate lower bound to the logit likelihood. Crucially, this conjugate model allows the variational distribution over inducing points to be analytically optimised, entirely removing the need for gradient-based optimisation of $q(\boldsymbol{u})$.

This analytical solution further permits strong relations to Chapter 3 as the classification model's effective likelihood is an unnormalised Gaussian distribution with heteroscedastic variance. Finally, we propose the use of a novel inducing point reinitialisation scheme for heteroscedastic GP classification and discuss its behaviour.

4.1 Variational Inference

4.1.1 The Lower Bound

As discussed in Section 2.6.3, [Wenzel et al. (22)] derive an augmented GP classification model based on Pólya-Gamma data augmentation and propose the use of a stochastic variational inference algorithm with the variational lower bound

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{p(\boldsymbol{f}|\boldsymbol{u})q(\boldsymbol{u})q(\boldsymbol{\omega})}[\log p(\boldsymbol{y}|\boldsymbol{\omega},\boldsymbol{f})] - \mathcal{KL}[q(\boldsymbol{u},\boldsymbol{\omega}) \,\|\, p(\boldsymbol{u},\boldsymbol{\omega})] \\ &= \frac{1}{2}(\log|\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}| - Tr(\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}) - \boldsymbol{m}^\top\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m} + M - 2N\log 2 + \\ &\boldsymbol{y}^\top\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m} - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) - Tr(\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}) - \\ &\boldsymbol{m}^\top\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m} + \Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})])\end{aligned}$$

where $\theta_n = \mathbb{E}_{q(\omega_n)}[\omega_n] = \frac{1}{2c_n}\tanh(\frac{c_n}{2})$, $\boldsymbol{\Theta} = \mathrm{diag}(\boldsymbol{\theta})$ and $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u};\boldsymbol{m},\boldsymbol{S})$.

In their algorithm, the form of $q(\boldsymbol{u})$ is computed through gradient-based optimisation and the local variational parameters $\boldsymbol{c}$ have a closed-form solution. Instead, we wish

to motivate the use of a collapsed model in which $q(\boldsymbol{u})$ is analytically optimised. As the conjugate model presents a closed-form variational lower bound, we can derive the form of the collapsed model by directly solving for the optimal $\boldsymbol{m}$ and $\boldsymbol{S}$.

We begin by finding the derivative with respect to $\boldsymbol{m}$

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{m}}\mathcal{L} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{m}}\left(-\boldsymbol{m}^\top \boldsymbol{K}_{uu}^{-1}\boldsymbol{m} + \boldsymbol{y}^\top \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m} - \boldsymbol{m}^\top \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}\right) \\ &= \frac{1}{2}\left(-2\boldsymbol{K}_{uu}^{-1}\boldsymbol{m} + \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} - 2\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}\right) \\ &= -(\boldsymbol{K}_{uu}^{-1} + \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1})\boldsymbol{m} + \frac{1}{2}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \end{aligned} \tag{4.1}$$

Setting this to zero, we solve for the optimal $\boldsymbol{m}^*$

$$\begin{aligned}(\boldsymbol{K}_{uu}^{-1} + \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1})\boldsymbol{m}^* &= \frac{1}{2}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\ (\boldsymbol{K}_{uu}^{-1}\boldsymbol{\Sigma}\boldsymbol{K}_{uu}^{-1})\boldsymbol{m}^* &= \frac{1}{2}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\ \boldsymbol{m}^* &= \frac{1}{2}(\boldsymbol{K}_{uu}^{-1}\boldsymbol{\Sigma}\boldsymbol{K}_{uu}^{-1})^{-1}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\ &= \frac{1}{2}\boldsymbol{K}_{uu}\boldsymbol{\Sigma}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \end{aligned} \tag{4.2}$$

with $\boldsymbol{\Sigma} = \boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}$. We can similarly find the derivative with respect to $\boldsymbol{S}$

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{S}}\mathcal{L} &= \frac{1}{2}\frac{\partial}{\partial \boldsymbol{S}}\left(\log|\boldsymbol{S}| - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}) - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{S})\right) \\ &= \frac{1}{2}\left(\boldsymbol{S}^{-1} - \boldsymbol{K}_{uu}^{-1} - \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\right) \end{aligned} \tag{4.3}$$

Setting this to zero, we solve for the optimal $\boldsymbol{S}^*$

$$\begin{aligned}\boldsymbol{S}^{*-1} &= \boldsymbol{K}_{uu}^{-1} + \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1} \\ &= \boldsymbol{K}_{uu}^{-1}\boldsymbol{\Sigma}\boldsymbol{K}_{uu}^{-1} \\ \implies \boldsymbol{S}^* &= \boldsymbol{K}_{uu}\boldsymbol{\Sigma}^{-1}\boldsymbol{K}_{uu} \end{aligned} \tag{4.4}$$

With these optimal parameters, we attain the final form of our analytical solution

$$q^*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}^*, \boldsymbol{S}^*) = \mathcal{N}(\boldsymbol{u}; \frac{1}{2}\boldsymbol{K}_{uu}\boldsymbol{\Sigma}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}, \boldsymbol{K}_{uu}\boldsymbol{\Sigma}^{-1}\boldsymbol{K}_{uu}) \tag{4.5}$$

which is noticeably similar to our heteroscedastic solution for regression

$$q^*_{\mathcal{R}}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}^*_{\mathcal{R}}, \boldsymbol{S}^*_{\mathcal{R}}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{K}_{uu}\boldsymbol{\Sigma}_{\mathcal{R}}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y}, \boldsymbol{K}_{uu}\boldsymbol{\Sigma}_{\mathcal{R}}^{-1}\boldsymbol{K}_{uu}) \tag{4.6}$$

with $\boldsymbol{\Sigma}_{\mathcal{R}} = \boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu}$. Crucially, we note that the variance of these two solutions is identical if we set $\boldsymbol{\Lambda} = \boldsymbol{\Theta}^{-1}$. This result tells us that the Pólya-Gamma random variables used to form $\boldsymbol{\Theta}$ give us the optimal heteroscedastic marginal variances in closed-form. The troubles that we encountered in Section 3.1.3 with fitting the heteroscedastic variance, do not exist for this classification model - it is automatically computed for us by the data augmentation! This is one of the fundamental results that we will later use to motivate the use of a heteroscedastic selection scheme.

Finally, we plug the parameters of $q^*(\boldsymbol{u})$ back into the original Pólya-Gamma bound to form the collapsed variational lower bound[1]

$$\begin{aligned}\mathcal{L}^* = &-\frac{1}{2}\log|\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\Sigma| - \frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}) + \frac{1}{8}\boldsymbol{y}^\top \boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} - \\ &N\log 2 + \frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})]\end{aligned} \tag{4.7}$$

where as a reminder of terms we have

$$\begin{aligned}\Sigma &= \boldsymbol{K}_{\boldsymbol{uu}} + \boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}} \\ \tilde{\boldsymbol{K}}_{ff} &= \boldsymbol{K}_{ff} - \boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}} = \boldsymbol{K}_{ff} - \boldsymbol{Q}_{ff} \\ q(\omega_n) &= PG(\omega_n; 1, c_n) \\ \theta_n &= \mathbb{E}_{q(\omega_n)}[\omega_n] = \frac{1}{2c_n}\tanh\left(\frac{c_n}{2}\right) \\ \boldsymbol{\Theta} &= \mathrm{diag}(\boldsymbol{\theta})\end{aligned}$$

4.1.2 The Upper Bound

[Wenzel et al. (22)] briefly detail the relationship between their work and the work of [Gibbs and MacKay (26)], and find that in the non-sparse case ($M = N$) their very different approaches lead to the same lower bound for GP classification. This is interesting as the former employ Pólya-Gamma data augmentation, whereas the latter compute a direct lower bound to the logit link without data augmentation. The key relationship is that the variational parameters of the logit link lower bound can be interpreted as the Pólya-Gamma parameters.

In the same work, an upper bound to the logit link function is proposed which initially looks very promising as a route to compute a variational upper bound for our classification model. However, this upper bound uses a different set of variational parameters which are not the Pólya-Gamma parameters.

Unfortunately, we find no alternative route to an upper bound as the marginal likelihood is not available in closed-form as a starting point. For this reason we do not propose an upper bound for the Pólya-Gamma model, but believe that it is worthwhile to note that [Gibbs and MacKay (26)] may present a promising avenue for future research in this area.

4.1.3 Optimisation

Now that we have the variational lower bound established with $q(\boldsymbol{u})$ analytically optimised, it remains that we only need to compute the kernel hyperparameters and the local variational parameters $\boldsymbol{c}$ (assuming fixed inducing points). The kernel hyperparameters must be computed through gradient-based optimisation, however it turns out that the optimal local variational parameters can also be computed in closed-form. [Wenzel et al. (22)] compute the closed-form solution to be

$$c_n^2 = \left[\tilde{\boldsymbol{K}}_{ff} + \boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}} + \boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m}\boldsymbol{m}^\top\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\right]_{nn} \tag{4.8}$$

[1] See Appendix C.1 for a full derivation.

which with our analytical solution gives

$$c_n^2 = \left[\tilde{\boldsymbol{K}}_{ff} + \boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf} + \frac{1}{4}\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}\boldsymbol{y}^\top\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf})\right]_{nn} \tag{4.9}$$

However, this can be more succinctly denoted by plugging in our analytical solution to the form of the approximate posterior distribution given in equation 2.8

$$\begin{aligned} q^*(\boldsymbol{f}) &= \mathcal{N}(\boldsymbol{f}\,;\,\boldsymbol{\mu}^*,\,\Sigma_q^*) \\ &= \mathcal{N}(\boldsymbol{f}\,;\,\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^*,\,\boldsymbol{K}_{ff} - \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}(\boldsymbol{K}_{uu} - \boldsymbol{S}^*)\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}) \\ &= \mathcal{N}(\boldsymbol{f}\,;\,\frac{1}{2}\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y},\,\tilde{\boldsymbol{K}}_{ff} + \boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}) \end{aligned} \tag{4.10}$$

This then allows us to succinctly define the closed-form solution for $\boldsymbol{c}$ as

$$c_n^2 = \left[\Sigma_q^* + \boldsymbol{\mu}^*\boldsymbol{\mu}^{*\top}\right]_{nn} \tag{4.11}$$

Optimisation then consists of a coordinate ascent algorithm where we alternately update $\boldsymbol{c}$ in closed-form, and then run gradient-based optimisation of the kernel hyperparameters. We no longer have to optimise $\boldsymbol{m}$ or $\boldsymbol{S}$ as they are optimally ingrained into the ELBO. One important nuance to note is that as $\boldsymbol{c}$ depends on $q^*(\boldsymbol{f})$, and $q^*(\boldsymbol{f})$ depends on $\boldsymbol{c}$, we have to update $\boldsymbol{c}$ multiple times in a loop until convergence of the ELBO to ensure a tight lower bound.

4.1.4 Implementation

Our second software contribution is an efficient implementation of the collapsed Pólya-Gamma lower bound and predictive distribution which use Cholesky decompositions and Woodbury inverses, computable in $\mathcal{O}(NM^2)$. The derivations can be found in Appendix C.2. We also provide a GPflow-based [Matthews et al. (25)] implementation in the same repository as earlier, available *here*.

To contrast this implementation with that of the heteroscedastic regression model, the main benefit is that the heteroscedastic marginal variances are directly computed for us in closed-form. This means that we are able to perform a thorough evaluation, and both qualitatively and quantitatively assess the model in Chapter 5. Before doing so however, we discuss the introduction of an inducing point selection scheme.

4.2 Motivations for a Selection Strategy

In Section 4.1.2 we discussed the matter of an upper variational bound for our heteroscedastic classification model. We noted that as the marginal likelihood is not available in closed-form, we cannot easily pursue an upper bound in the same manner as we did for heteroscedastic regression. Now we further note that the lack of a closed-form marginal likelihood prevents us from justifying the use of an inducing point selection strategy through bounds on the KL divergence as in Section 3.2.

In this section, we instead investigate the similarities between our novel classification model and the heteroscedastic regression model discussed in Chapter 3. We make note of key relationships between the two models and use these to motivate the use of the very same heteroscedastic selection method discussed in Section 3.3.2. We later reinforce this reasoning with a thorough empirical evaluation in Chapter 5.

4.2.1 Pólya-Gamma Variance

In Section 4.1.1, we discussed the relationship between the form of the optimal variational distribution over inducing points in the heteroscedastic classification and regression frameworks. We showed that our analytical solution for classification

$$q^*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}^*, \boldsymbol{S}^*) = \mathcal{N}(\boldsymbol{u}; \frac{1}{2}\boldsymbol{K}_{\boldsymbol{uu}}\Sigma^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{y}, \boldsymbol{K}_{\boldsymbol{uu}}\Sigma^{-1}\boldsymbol{K}_{\boldsymbol{uu}}) \tag{4.12}$$

with $\Sigma = \boldsymbol{K}_{\boldsymbol{uu}} + \boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}$, is similar to the solution for heteroscedastic regression

$$q^*_{\mathcal{R}}(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}^*_{\mathcal{R}}, \boldsymbol{S}^*_{\mathcal{R}}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{K}_{\boldsymbol{uu}}\Sigma^{-1}_{\mathcal{R}}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Lambda}^{-1}\boldsymbol{y}, \boldsymbol{K}_{\boldsymbol{uu}}\Sigma^{-1}_{\mathcal{R}}\boldsymbol{K}_{\boldsymbol{uu}}) \tag{4.13}$$

with $\Sigma_{\mathcal{R}} = \boldsymbol{K}_{\boldsymbol{uu}} + \boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{\boldsymbol{fu}}$. Namely, we noticed that the variance of the two solutions is identical if we set $\boldsymbol{\Lambda} = \boldsymbol{\Theta}^{-1}$. This tells us that the Pólya-Gamma data augmentation directly computes the heteroscedastic variance for us in closed-form, $\theta_n = \frac{1}{2c_n}\tanh(\frac{c_n}{2})$. This relationship can be more formally found by inspecting the form of the unnormalised Gaussian attained by the Pólya-Gamma lower bound to the likelihood - by completing the square we find that the covariance matrix is $\boldsymbol{\Theta}^{-1}$.

Whilst the optimal mean functions are slightly different, this sheds light on a key relationship between the two models which reinforces later findings.

4.2.2 Augmented Lower Bound

In this subsection, we inspect the relationship between the collapsed variational lower bounds in the heteroscedastic classification and regression frameworks. We start from our collapsed Pólya-Gamma bound for classification

$$\begin{aligned}\mathcal{L}^* = &-\frac{1}{2}\log|\boldsymbol{K}^{-1}_{\boldsymbol{uu}}\Sigma| - \frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{\boldsymbol{ff}}) + \frac{1}{8}\boldsymbol{y}^\top\boldsymbol{K}_{\boldsymbol{fu}}\Sigma^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{y} - \\ &N\log 2 + \frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})]\end{aligned}$$

which can be alternatively written as[2]

$$\begin{aligned}\mathcal{L}^* = &\log\mathcal{N}(\boldsymbol{y}; \boldsymbol{0}, 4\boldsymbol{\Theta}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{\boldsymbol{ff}})\boldsymbol{\Theta}) - \frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{\boldsymbol{ff}}) + \\ &\frac{1}{2}\log|\boldsymbol{\Theta}| + \frac{1}{8}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{N}{2}\log 2\pi + \frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})]\end{aligned} \tag{4.14}$$

In this formulation we see that the first two terms are exactly the same as the collapsed heteroscedastic regression bound with $\boldsymbol{\Lambda} = \boldsymbol{\Theta}^{-1}$, but with additional augmentation of $2\boldsymbol{\Theta}$ on either side of the log-normal covariance. The other four terms are constant, assuming that the local variational parameters $\boldsymbol{c}$ are held fixed (and thus $\boldsymbol{\Theta}$ is fixed). This tells us that the two models are almost identical up to a constant, but with the classification model being augmented by Pólya-Gamma data augmentation. This formulation reinforces the relationship between the two models, and suggests that we may be able to follow similar reasonings as in the regression case to motivate the use of an inducing point selection scheme for classification.

[2] See Appendix C.3 for a full derivation.

4.3 Inducing Point Selection

The use of an inducing point reinitialisation scheme is a wholly new and exciting concept for Gaussian process classification, made possible by our novel collapsed model. In this section we build upon the recent success of inducing point reinitialisation schemes for GP regression, and consider the use of inducing point reinitialisation for heteroscedastic GP classification.

4.3.1 Monotonic Bound Increase

As in the case of regression, we believe that it is first and foremost critical to verify that during inducing point selection our heteroscedastic classification model exhibits monotonicity in the variational lower bound. Specifically, this means that our chosen scheme should never decrease the ELBO or worsen the approximation by adding additional inducing points. We provide a derivation in Appendix C.4 which shows that our heteroscedastic collapsed model does indeed exhibit this property.

4.3.2 Heteroscedastic Greedy Variance

Whilst intractability of the marginal likelihood made analysis of the GP classification framework difficult, we believe that the relationships discussed in Section 4.2 between heteroscedastic regression and classification, suggest grounds for an inducing point selection strategy. In particular, we saw that $\mathbf{\Theta}^{-1}$ corresponds to the heteroscedastic likelihood variance $\mathbf{\Lambda}$, and that we can yield a classification bound that represents an augmented regression bound differing by only a constant. For these reasons, we propose the use of the same augmented inducing point reinitialisation scheme, 'heteroscedastic greedy variance', as was detailed in Section 3.3.2.

In the case of GP classification, this method iteratively minimizes $Tr(\mathbf{\Theta}\tilde{\boldsymbol{K}}_{ff})$ by greedily selecting candidate inducing points from the criterion $argmax\, diag[\mathbf{\Theta}\tilde{\boldsymbol{K}}_{ff}]$. Again, this is implemented in practice as a rank-1 update cycle which avoids the need for costly repetitive computations.

As in the case of regression, this method can either run until a fixed number of points are chosen, or until a threshold on the trace term is achieved. We stress the substantial benefit of this contribution, as our model is now able to automatically select how large M should be in order to produce a sufficiently accurate approximation. This is in contrast to the current state-of-the-art which requires developers to run an exhaustive search or provide a possibly suboptimal point estimate of M. We therefore believe that this method could prove very beneficial for future research.

In Section 4.1.3, we discussed the inner optimisation loop of the local variational parameters $\boldsymbol{c}$ and kernel hyperparameters, with fixed inducing points. With the introduction of our inducing point selection scheme, the entire optimisation cycle consists of alternately running the inner optimisation loop and then reinitialising the inducing points using heteroscedastic greedy variance until convergence of the ELBO. Due to this doubly-looped behaviour, this can now be seen as a double-layered variational EM algorithm. In practice, this double-layered algorithm is much more efficient than gradient-based optimisation of $q(\boldsymbol{u})$ and the inducing points.

In Section 3.3.2, we also discussed that this augmented criterion encourages the selection of inducing points that have a low marginal variance. This fits nicely with our hypothesis from the introduction of this report, where we stated that we believe clustering the inducing points around decision boundaries may lead to increased predictive performance for Gaussian process classification. In Section 5.2, we visually demonstrate that the Pólya-Gamma data augmentation produces low variance regions around the predictive decision boundaries, and so our selection criterion which is augmented by the reciprocal variance, $\boldsymbol{\Theta}$, will favour points near the boundaries. This trait allows us to evaluate our hypothesis in Chapter 5 and further reinforce our beliefs that heteroscedastic greedy variance is a favourable selection strategy.

4.4 Summary

In this chapter, we extended the foundations of Chapter 3 to the case of heteroscedastic GP classification. Here we summarise the main contributions:

- We derived a novel model for heteroscedastic GP classification in which the variational distribution over inducing points was analytically optimised. This provided the novel benefit of entirely removing the need for gradient-based optimisation of $q(\boldsymbol{u})$ in the GP classification framework.
- We derived efficient closed-form implementations of the variational lower bound and predictive distribution, and provided a GPflow-based implementation in a public GitHub repository, available *here*.
- We investigated relationships between our heteroscedastic classification model and the heteroscedastic regression model discussed in Chapter 3. In particular, we showed that the heteroscedastic variance can be automatically determined by the Pólya-Gamma random variables, and that the collapsed heteroscedastic classification bound can be written as an augmented case of the collapsed heteroscedastic regression bound.
- We reviewed requirements for a reliable inducing point selection method and used the aforementioned relationships to motivate the use of the same augmented inducing point selection scheme proposed in Chapter 3. We made clear that one of the most significant benefits of this method is that we can automatically select how large M should be, which is novel for GP classification.
- Finally, we revisited our hypothesis that clustering inducing points around decision boundaries may lead to increased predictive performance for GP classification. We briefly justified that our heteroscedastic inducing point selection method inherently encourages the selection of points nearby decision boundaries, which will allow us to later evaluate our hypothesis in Chapter 5.

Chapter 5

Evaluation

In this chapter we evaluate the performance of our novel classification model and inducing point selection procedure. We compare and contrast both qualitative and quantitative aspects against the current state-of-the-art, with the intent to thoroughly address a number of crucial questions:

- How accurately does our Pólya-Gamma classification model approximate the exact Bernoulli classification model?
- How does our inducing point selection procedure qualitatively behave in comparison to state-of-the-art alternatives?
- How sparse can we go with different inducing point selection procedures?
- Does gradient-based optimisation of the inducing points help?
- Can the combination of our model and our inducing point selection procedure quantitatively outperform the current state-of-the-art?

5.1 Preliminaries

Before proceeding with our evaluation, we wish to cover preliminaries such as introducing experimental terminology and detailing the procedures used to ensure fair, consistent and accurate analysis throughout the rest of this chapter.

5.1.1 Terminology

Our evaluation is performed through the use of the GPflow [Matthews et al. (25)] library, and so our terminology in this chapter aligns with that used in GPflow.

- **SVGP**: The state-of-the-art model for Gaussian process classification. In this model $q(\boldsymbol{u})$ is optimised through gradient-based optimisation, with M parameters for $\boldsymbol{m}$, and $M(M+1)/2$ parameters for the Cholesky factorisation $\boldsymbol{S} = \boldsymbol{L}\boldsymbol{L}^\top$.
- **PGPR**: Our novel heteroscedastic Gaussian process classification model using Pólya-Gamma data augmentation, in which $q(\boldsymbol{u})$ is analytically optimised.
- **GV**: Greedy variance inducing point selection, as discussed in Section 2.7.2.
- **HGV**: Our novel heteroscedastic greedy variance inducing point selection procedure, as discussed in Sections 3.3.2 and 4.3.2.

5.1.2 Experimental Procedures

We briefly detail some key aspects of our evaluation and training procedures below:

- SVGP is set up with a Bernoulli likelihood using a sigmoidal inverse link function so that it is comparable against our PGPR model based on the logit link.
- All models use a squared exponential kernel with consistent initial hyperparameters. These kernels have a single shared lengthscale for all dimensions.
- All models are optimised using the 'L-BFGS-B' algorithm. This is a fast-converging quasi-Newton method which approximates the traditional BFGS algorithm by using a limited amount of computer memory and box constraints.
- All experiments are given a sufficient number of optimisation iterations for convergence, unless this is computationally infeasible and upper limits are used.
- All quantitative experiments are run ten times in order to reduce the effect of random noise, and we inspect the distribution over results.
- Experiments on smaller datasets ($N < 1,000$) are run on a single CPU, and experiments on larger datasets ($N \geq 1,000$) are run on a single GPU.

5.1.3 Robust Optimisation

Gaussian processes are notoriously difficult to train due to the fragility of operations required for their optimisation. Namely, the largest issues are caused by matrix inversions or Cholesky decompositions which can be numerically unstable and cause software exceptions in the presence of ill-conditioned matrices.

In order to improve matrix conditioning, a small jitter term, $\epsilon \approx 10^{-6}$, is often added to the matrix $\boldsymbol{K_{uu}}$. This effectively increases the relative size of the smallest eigenvalue of the kernel matrix and improves the condition number. Whilst this jitter term changes the optimisation objective, the effect on the final result is typically small and the jitter is often enough to avoid software exceptions. It can be shown however that the variational lower bound is monotonically decreasing in ϵ, and so we wish to avoid adding any more jitter than is absolutely necessary[1].

One problem with this fixed procedure is that 10^{-6} is a rather arbitrary number. As the size of the matrix grows, conditioning typically worsens and 10^{-6} may not be enough to avoid exceptions. On the other hand, the matrix may not even need any jitter, and by adding it we only worsen our bound and thus our approximation.

To address this, we implement a dynamic jitter scheme in which the amount of jitter is automatically adjusted to reduce the bias introduced into our model. We do so by first attempting to decompose the matrix without any jitter, and then adding small incremental amounts of jitter in the case of an exception. If a maximum amount of jitter still fails, then we finally throw an exception. We find that often jitter is not even required, or that $\epsilon = 10^{-9}$ will suffice. This adaptive scheme allows more accurate and robust inference, and typically improves the performance of our models.

[1] The quantitative effect of jitter is discussed in detail in [Burt et al. (10)].

5.1.4 Datasets

We perform our evaluations on ten datasets chosen from the OpenML [Vanschoren et al. (27)] and UCI [Dua and Graff (28)] repositories, along with an additional toy dataset. The datasets are briefly detailed in the table below with their size, dimensionality and description. All datasets are available in our GitHub repository, with download links and further information.

Dataset	N	D	Brief Description
Platform	50	1	Toy dataset
Crabs	200	6	Detecting sex of Leptograpsus crabs
Heart Statlog	270	13	Detecting heart disease
Ionosphere	351	33	Radar returns from the ionosphere
Banana	400	2	Artificial clustering example
Breast Cancer	569	30	Detecting breast cancer
Pima Diabetes	768	8	Detecting diabetes in Pima Indians
Twonorm	7,400	20	Leo Breiman's twonorm example
Ringnorm	7,400	20	Leo Breiman's ringnorm example
MAGIC Telescope	19,020	10	Particle registration in a gamma telescope
Electricity	45,312	8	'NSW Electricity Market' price change

Table 5.1: Brief summary of datasets used for evaluation.

5.2 Pólya-Gamma Model

In this section we perform a brief qualitative assessment of our classification model, PGPR, in the non-sparse case. We use the Banana dataset to visually explore how the Pólya-Gamma data augmentation can be used to incentivise the use of our inducing point selection scheme, HGV, in the sparse case.

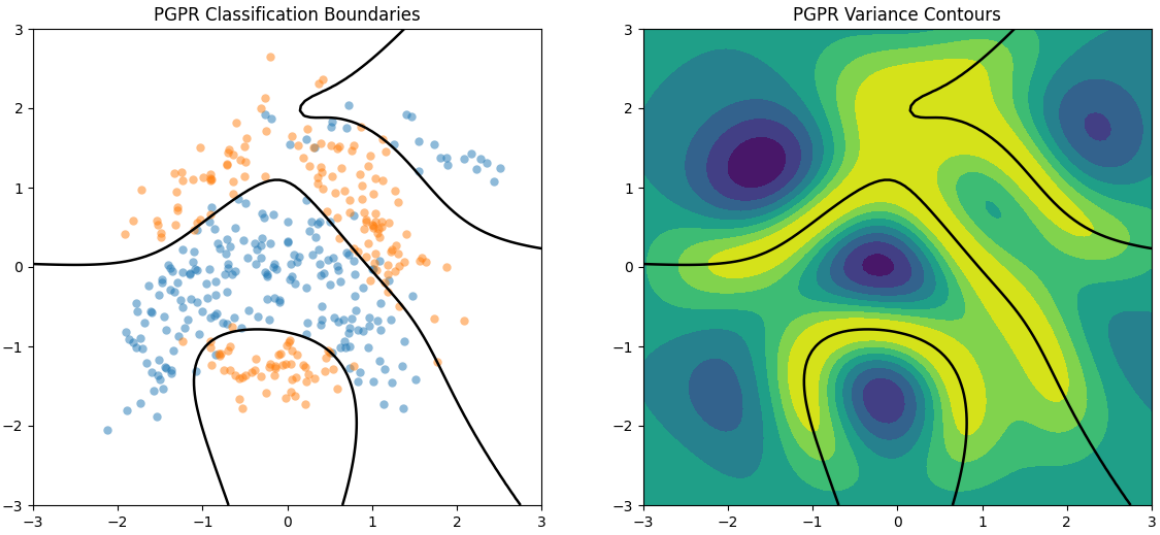

Figure 5.1: PGPR classification on the Banana dataset. The left image shows the data and PGPR predictive decision boundaries. The right image shows the contoured Pólya-Gamma variance, $\mathbf{\Theta}^{-1}$, where yellow and blue signify low and high variance respectively.

From the left image in Figure 5.1, we see that PGPR separates the two classes well and does not display any undesirable behaviour. However, the most important piece

of information that we find is contained in the right image. This contour plot is created by computing the value $\theta_n = \frac{1}{2c_n}\tanh\left(\frac{c_n}{2}\right)$ for a set of test points in the 2D grid, where c_n is defined in equation 4.11. We then plot contours of the Pólya-Gamma variance, θ_n^{-1}, as discussed in Section 4.2.1. This plot crucially tells us that the Pólya-Gamma variance is lowest at the predictive boundaries of the model. For a more thorough intuition as to why PGPR exhibits such behaviour, see Section 5.5.

Turning to our heteroscedastic inducing point selection scheme, HGV, we know that the selection criterion is $argmax\ diag[\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}]$. Therefore any point where θ_n is large (or θ_n^{-1} is small) is more likely to be chosen as an inducing point. This means that points at the boundaries will be chosen more often as they have been shown to have low variance. This entirely relates to our hypothesis proposed at the beginning of this report - we stated that it would be desirable to place inducing points at the predictive boundaries, and HGV will do just that! We are now able to evaluate our hypothesis directly by evaluating the performance of HGV.

5.3 Model Comparison

In this section we perform a brief qualitative analysis of our model, PGPR, against the exact Bernoulli model, SVGP, in the non-sparse case. We first inspect the performance of both models on our one-dimensional toy dataset, plotting the inputs on the horizontal axis and the predictive probabilities $p(y = 1|x)$ on the vertical axis.

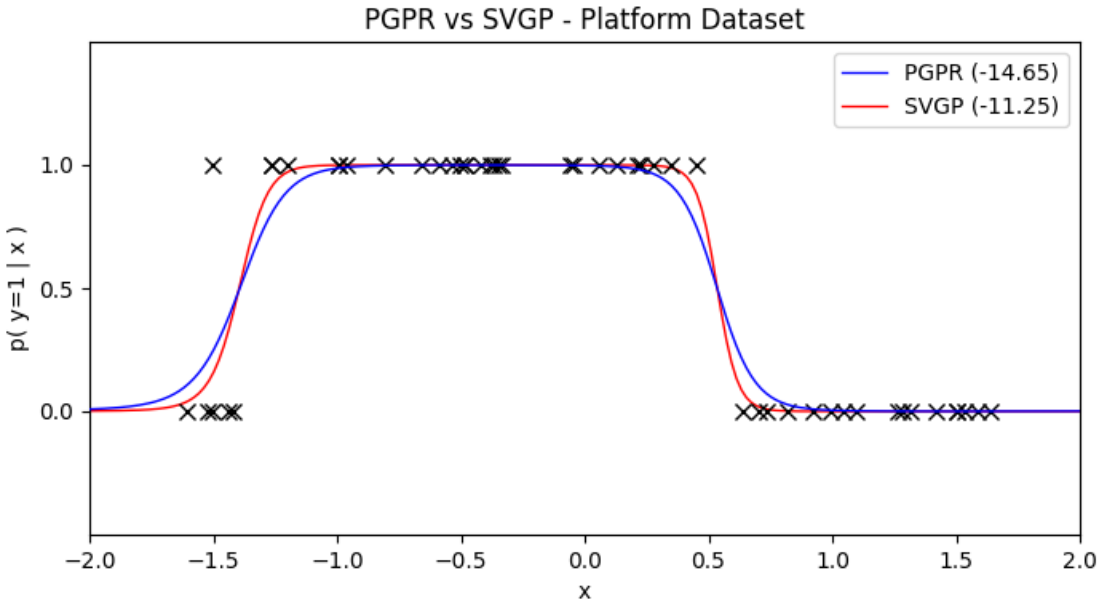

Figure 5.2: PGPR vs SVGP on the toy dataset, Platform. The predictive probabilities of PGPR and SVGP are plotted in blue and red respectively, with the ELBOs in the legend.

From this image we see that the two models perform similarly, but PGPR holds a softer gradient at the decision boundary. Both models achieve the same accuracy with one outlier around $x = -1.5$. The ELBO achieved by SVGP is higher than that of PGPR, which is to be expected as PGPR uses a lower bound to the Bernoulli likelihood and so the log-likelihood term in the ELBO is smaller. As for the two-dimensional Banana dataset, we plot the predictive boundaries for $p(y = 1|x) = 0.5$.

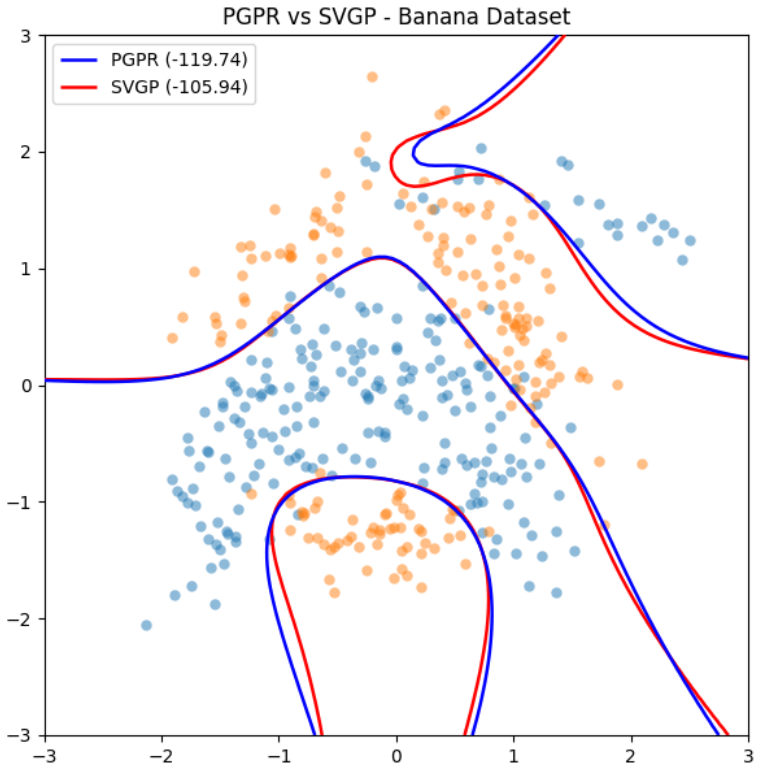

Figure 5.3: PGPR vs SVGP on the Banana dataset. The predictive decision boundaries of PGPR and SVGP are plotted in blue and red respectively, with the ELBOs in the legend.

Again we see that the two models perform similarly, with the ELBO achieved by SVGP being higher than that of PGPR. Whilst it may appear that SVGP simply outperforms PGPR, there are two important details to note. Firstly, due to the analytical nature of PGPR, it is much quicker and more robust to train than SVGP which finds $q(\boldsymbol{u})$ through gradient-based optimisation. Secondly, again due to its analytical nature, PGPR is able to use our inducing point selection method which may allow it to outperform SVGP in the sparse case - this is what we are most interested about.

Finally, we show that SVGP with a Pólya-Gamma likelihood is equivalent to PGPR upon convergence of $q(\boldsymbol{u})$. This ensures that there are no deficiencies in our model, and that the only difference is the lower bound to the Bernoulli likelihood. We assess both models on the Banana dataset with an increasing number of inducing points initialised by uniform subsampling, and see that both models perform identically.

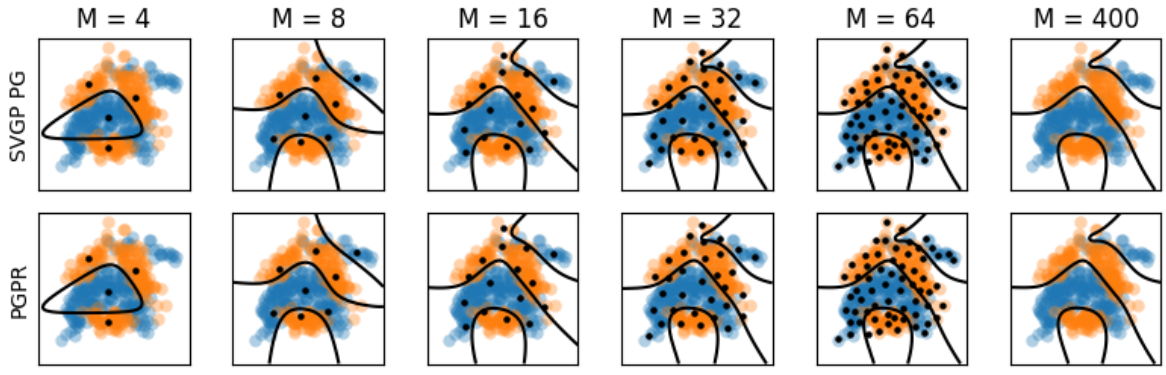

Figure 5.4: PGPR vs SVGP with a Pólya-Gamma likelihood on the Banana dataset with increasing M. The predictive boundaries and inducing inputs are highlighted in black.

5.4 Inducing Point Selection

We now perform a brief qualitative analysis of our model, PGPR, against the exact Bernoulli model, SVGP, in the sparse case. To do so we visually compare the behaviour of different inducing point selection methods on the Banana dataset. In particular, we plot the predictive boundaries and inducing inputs for four methods:

- **SVGP GO**: $\boldsymbol{Z}$ initialised by k-means clustering, and then gradient-optimised.
- **PGPR GO**: $\boldsymbol{Z}$ initialised by k-means clustering, and then gradient-optimised.
- **PGPR GV**: $\boldsymbol{Z}$ chosen through greedy variance reinitialisation.
- **PGPR HGV**: $\boldsymbol{Z}$ chosen through heteroscedastic greedy variance reinitialisation.

For k-means, we choose M clusters using the default implementation of k-means++ from the SciPy Python package. We run gradient-based optimisation of the inducing points until convergence of the ELBO. For GV and HGV we reinitialise the inducing points until convergence of the ELBO, or until a maximum number of ten iterations.

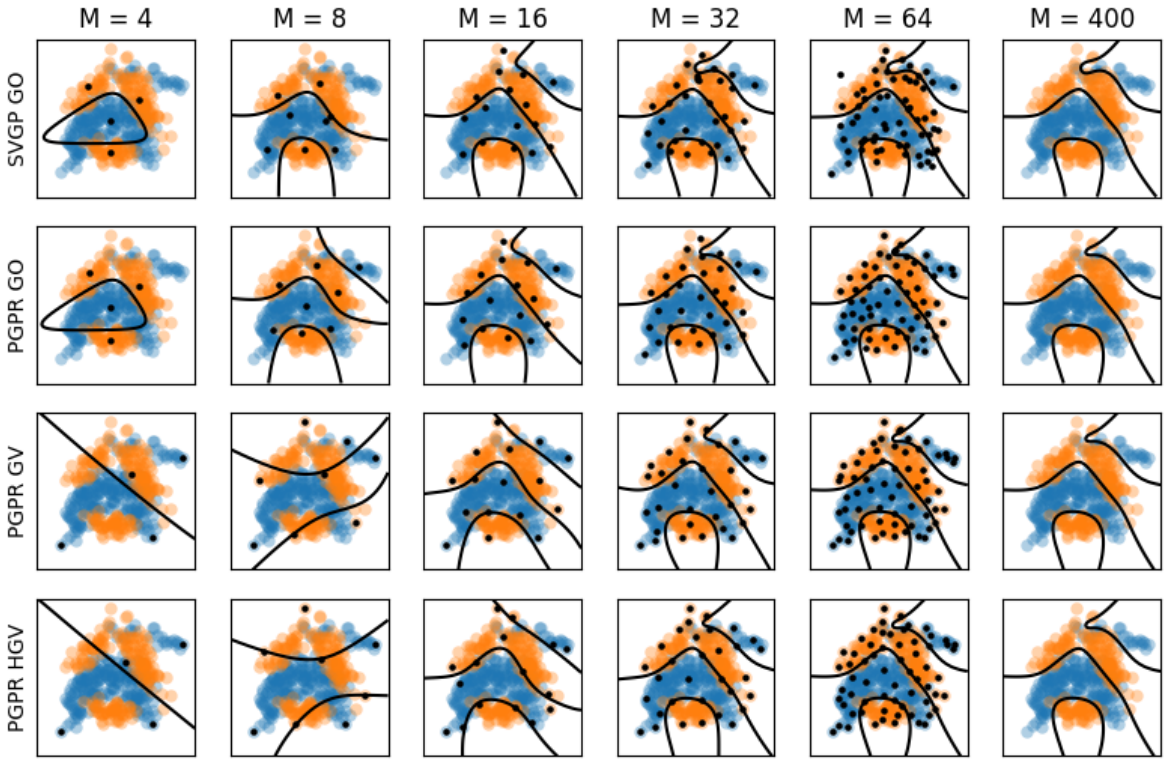

Figure 5.5: PGPR vs SVGP with different inducing point selection methods on the Banana dataset. The predictive boundaries and inducing inputs are highlighted in black.

From Figure 5.5, we notice straight away that SVGP GO and PGPR GO lead to almost identical predictive boundaries, except for $M = 8$ where PGPR finds a better boundary; this is intriguing as the optimised points are in very different locations. SVGP GO appears to draw most of the inducing points close to the boundaries, which matches our hypothesis. PGPR GO on the other hand appears to disperse the inducing points like a regression model, and then place a few points at the boundaries.

This link between the placement of $\boldsymbol{Z}$ for a regression model and for PGPR may be caused by the fact that PGPR's effective likelihood is an unnormalised Gaussian.

However, in the exact Bernoulli case we indeed see that SVGP prioritises points at the boundaries and so believe that these placements may still be beneficial for PGPR.

Turning to PGPR GV, we see that the points are well dispersed which matches the links made between greedy variance and the approximate sampling of a k-DPP in Section 2.7.2. Again, this is a characteristic of the placement of $\boldsymbol{Z}$ for a regression model and does not fit our hypothesis. We see that PGPR GV performs suboptimally for small M, and this is because the selection criterion is not suitable for severely underparameterised models where each element in the trace is almost uniformly large.

Finally, we look at PGPR HGV and see that our hypothesis is much more closely followed than in the alternative PGPR methods. Whilst there is still some dispersion of inducing points, many of the inducing points are placed exactly on the decision boundaries. It is clear that HGV initially chooses points to cover the domain, and then prioritises points along the boundaries where the Pólya-Gamma variance is low.

As with PGPR GV, we see that PGPR HGV performs suboptimally for small M. This is again influenced by the fact that for an underparameterised model, the Nyström difference in the selection criterion is almost uniformly large for all data points. Additionally, as the Pólya-Gamma variance is computed by the model itself, if the model is underparameterised then the variance term in the selection criterion will be inaccurate. This is not a problem in practice, as we will always choose M large enough to ensure that the model is well calibrated.

5.5 Behaviour at the Boundaries

In Section 5.2, we saw that the Pólya-Gamma variance was the lowest at the boundaries. Subsequently in Section 5.4, we saw that gradient-optimised SVGP pulls its inducing points towards the predictive boundaries, and that HGV also places its inducing points close to the boundaries. Whilst we have already discussed this in part, we believe it important to provide intuition as to why we see such behaviour in both cases - we take a brief interlude in this section to discuss these matters. We also investigate the benefit of possessing such knowledge during inducing point selection.

5.5.1 Bernoulli Model

We begin by discussing why SVGP pulls its inducing points towards the predictive boundaries. In order to do so we recall that the variational lower bound in equation 2.13 contains two terms: an expected log-likelihood data fit term, and a KL prior regularisation term. We focus on the data fit term, which uses a Bernoulli likelihood for classification to penalize points with a low predictive probability for the correct label.

Before we discuss the data fit penalty, we first remember that the predictive boundaries represent the plane $\sigma(\mu_n^*) = 0.5$, where the predictive mean is passed through the logit inverse-link (sigmoid) function. By recalling the shape of the sigmoid, we remember that $\sigma(\mu_n^*) = 0.5$ is attained when $\mu_n^* = 0$, and that the transition phase lies roughly in $[-5, 5]$, outside of which the sigmoid saturates to 0 and 1 respectively.

With this shape in mind, we consider two scenarios. Firstly, we consider a point where the sigmoid output is highly saturated. In this case, a high predictive variance will not significantly affect the output - even if the input deviates largely from its mean, the output will still be near 0 or 1. Secondly, we consider a point close to the boundary $\sigma(0) = 0.5$. In this case, a high predictive variance can significantly affect the sigmoid output, which in turn induces a large data fit penalty. We therefore understand that high predictive variance far from the boundaries is inconsequential, whereas high predictive variance close to the boundaries largely penalizes the ELBO.

As SVGP optimises the location of the inducing points using the ELBO as its objective function, it therefore aims to minimize the aforementioned data fit penalty. To achieve this, the model will place the inducing points near the predictive boundaries which reduces the predictive uncertainty in these regions, and consequently reduces the data fit penalty. This is why gradient-based optimisation of the inducing points in SVGP pulls the inducing points towards the predictive decision boundaries.

5.5.2 Pólya-Gamma Model

We now discuss why PGPR exhibits low variance at the boundaries, which is much simpler than in the Bernoulli case. As previously discussed, the effective Pólya-Gamma variance for a specific data point is θ_n^{-1}, where $\theta_n = \frac{1}{2c_n}\tanh\left(\frac{c_n}{2}\right)$ and $c_n \geq 0$. From this we see that when c_n is small, θ_n is large and thus the variance θ_n^{-1} is small.

In Section 4.1.3 we saw that the optimal c_n was computable as $c_n^2 = \left[\Sigma_q^* + \boldsymbol{\mu}^*\boldsymbol{\mu}^{*\top}\right]_{nn}$, where the predictive distribution is $q^*(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}\ ;\ \boldsymbol{\mu}^*,\ \Sigma_q^*)$. For a well calibrated model, Σ_q^* is more uniform than $\boldsymbol{\mu}^*$ across the training set, and we remember that the predictive boundaries represent the regions with mean $\mu_n^* = 0$ and so $\sigma(0) = 0.5$. Therefore our closed-form solution for c_n, which involves the squared mean, will be smaller at the boundaries than far from the boundaries where the squared mean is large. This consequently means that θ_n^{-1} will be smaller in these regions as displayed in Figure 5.1. As previously discussed, HGV favours inducing points in low variance regions and so will tend to select these points close to the predictive boundaries.

5.5.3 Inducing Point Optimality

Now we have gained intuition as to why SVGP pulls its inducing points towards the boundaries, and as to why PGPR HGV behaves similarly, we wish to more accurately quantify the benefit of using such knowledge during inducing point initialisation. We propose to do so by comparing the convergence of SVGP with gradient-optimised inducing points, with the inducing points initialised through the state-of-the-art k-means, and through HGV. We inspect the learning curves for both regimes, with the postulation that whichever regime converges in the fewest number of iterations is initialising the inducing points closest to the optimal placement.

For this experiment we plot the variational lower bound against the number of optimisation iterations. To corroborate these results, we repeat the experiment using both the 'L-BFGS-B' optimiser and the 'Adam' optimiser. For the Adam optimiser we set $(\beta_1, \beta_2) = (0.5, 0.5)$, initialise the learning rate to 0.5, and use cosine annealing with a minimum learning rate of 0.025 to encourage steady convergence.

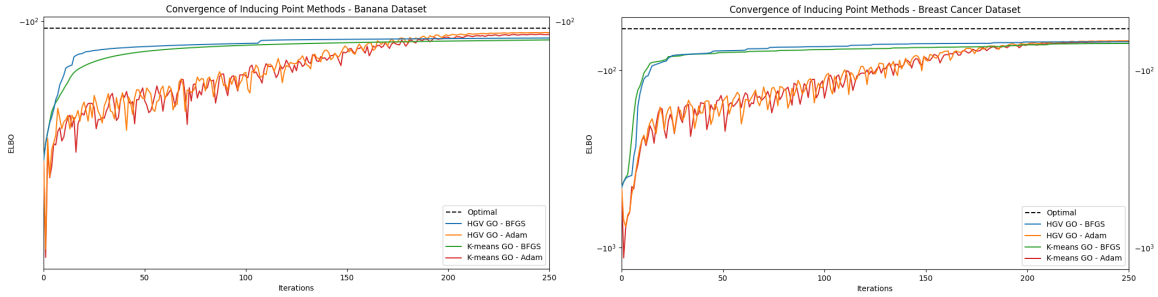

Figure 5.6: SVGP with optimised inducing points initialised by HGV and k-means on the Banana (N=400) and Breast Cancer (N=569) datasets. We repeat the experiment using both the 'L-BFGS-B' and 'Adam' optimisers. The value of the variational lower bound and the number of optimisation iterations are plotted on the vertical and horizontal axes respectively. The optimal non-sparse results are denoted by dashed black lines.

The results for the other datasets can be found in Appendix D.1. From these results we see that in all datasets HGV GO converges faster to its optimal value, and in some cases achieves a superior optimum to K-means GO; we see this behaviour for both optimisers. To compare the two optimisers, we see that L-BFGS-B always learns faster in the early iterations, but Adam is able to overtake in the later stages for some datasets such as Banana. Generally however, we see than L-BFGS-B is much more reliable, providing a monotonic guarantee and removing fragile hyperparameters.

The main constant between all experiments is that HGV GO always achieves an equal or superior ELBO to K-means GO at convergence. This confirms our beliefs that HGV selects inducing points that are closer to the predictive boundaries, and are thus closer to the optimal placements for the SVGP model.

The number of inducing points used in this experiment were chosen to be the smallest feasible M at which SVGP could attain a near-optimal ELBO - specific values can be found in Appendix D.1. Importantly, we note that the number of inducing points required for SVGP to achieve its optimal ELBO is *much* larger than the number required for PGPR to achieve its optimum. For example, in the above SVGP Banana experiment we use $M = 80$ which struggles to attain its optimum, whereas we find in the next section that $M = 35$ suffices for PGPR; this is another key benefit of PGPR.

5.6 Sparsity

With this knowledge, we move on to answer the question: 'How sparse can we go?'. We attempt to answer this question by assessing the performance of PGPR paired with a number of inducing point selection methods, and inspect the value of the variational lower bound as we increase M. This sheds light onto the effectiveness of each method, and allows us to discover which method permits the sparsest models.

5.6.1 Fixed Inducing Points

We begin by looking at selection methods where the inducing points are fixed. By 'fixed' we mean to say that the inducing points are not optimised through gradient-based optimisation, although the inducing points of GV and HGV are not literally fixed in the case of reinitialisation. The four methods that we investigate are:

- **Uniform**: $\boldsymbol{Z}$ chosen through uniform subsampling.
- **K-means**: $\boldsymbol{Z}$ chosen through k-means clustering.
- **GV**: $\boldsymbol{Z}$ chosen through greedy variance reinitialisation.
- **HGV**: $\boldsymbol{Z}$ chosen through heteroscedastic greedy variance reinitialisation.

For Uniform, we sample the inducing points with uniform probability from the training set, without replacement. For K-means, we select the inducing points as in Section 5.4, where k-means++ is used to initialise the clusters. For GV and HGV, we also select the inducing points as in Section 5.4. We run the experiment multiple times for each value of M, plotting the mean value of the variational lower bound on the vertical axis and the number of inducing points on the horizontal axis.

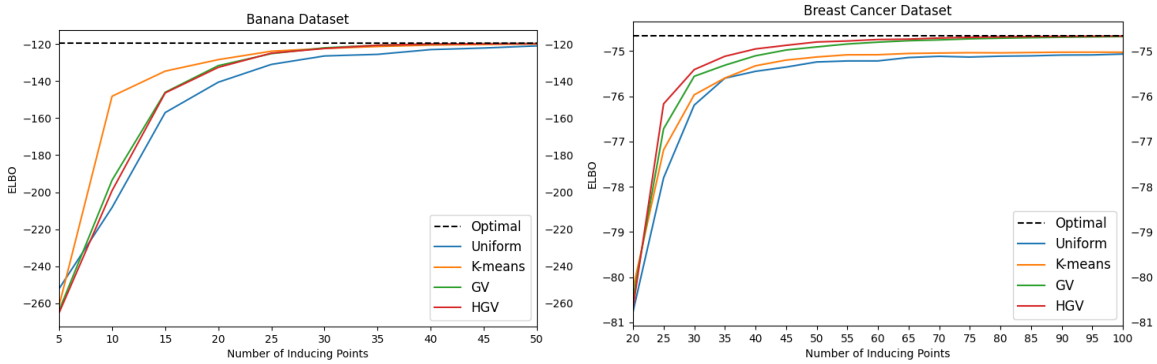

Figure 5.7: PGPR with four inducing point selection methods on the Banana (N=400) and Breast Cancer (N=569) datasets. The value of the variational lower bound and the number of inducing points are plotted on the vertical and horizontal axes respectively, and the optimal non-sparse result is denoted by the dashed black line.

The results for the other eight datasets can be found in Appendix D.2.1. We see from the majority of cases that K-means often outperforms all other methods for very small M. This is because both GV and HGV are based on properties of the model, and their selection criterions are poorly calibrated when the model is severely underparameterised, as discussed in Section 5.4.

We see however that both GV and HGV consistently outperform Uniform and K-means as M grows, and approach the optimal ELBO with a much smaller M. Take the Breast Cancer results for example: GV and HGV both converge to the optimal value around $M = 80$, whereas Uniform and K-means are still significantly suboptimal. This behaviour is what we really care about - which method is able to converge to the optimal ELBO with the fewest number of inducing points. It is not particularly useful that K-means outperforms when M is small, as we would never use such an M where the ELBO is not close to optimal. Out of all the datasets, there is not a single case where Uniform or K-means outperforms GV or HGV at convergence.

Finally, to compare GV and HGV we see that the two methods perform fairly similarly, with two noticeable differences. Firstly, GV tends to perform slightly better with very small M, this is likely because the Pólya-Gamma variances in HGV are poorly calibrated. Secondly, HGV consistently achieves a better ELBO than GV when M grows large enough. Again, the latter case is what we care about - it seems that out of all methods at convergence, HGV is able to achieve the best ELBO with the

fewest number of inducing points. In light of these findings, we believe that HGV is the best fixed inducing point selection method investigated for the PGPR model.

Aside from directly inspecting these graphs and seeing which M is sufficient, it is important to remember that a key innovation of GV and HGV is that the model is able to automatically select the number of inducing points. By inspecting the trace term during greedy selection, both methods are able to tell when they have converged and can stop accepting new points. As a crude metric, we often see that HGV permits models that use less than $5-10\%$ of the total training set, which is a huge computational saving. We do note however that some datasets simply do not permit a sparse approximation; for example, the 'kin40k' UCI dataset as thoroughly investigated in [van der Wilk (13)]. We do not propose a remedy for such cases.

5.6.2 Gradient-Based Optimisation

We now move on to answer the question: 'Does gradient-based optimisation help?'. To do so we take the best performing method from the previous section (HGV), and run the same experiment with and without gradient-based optimisation of the inducing points. For the optimised case, we optimise the inducing points throughout the entire reinitialisation procedure. We also compare these methods to the current state-of-the-art, gradient-optimised k-means. Again, we run the experiment multiple times for each value of M, and plot the mean value of the variational lower bound and the number of inducing points on the vertical and horizontal axes respectively.

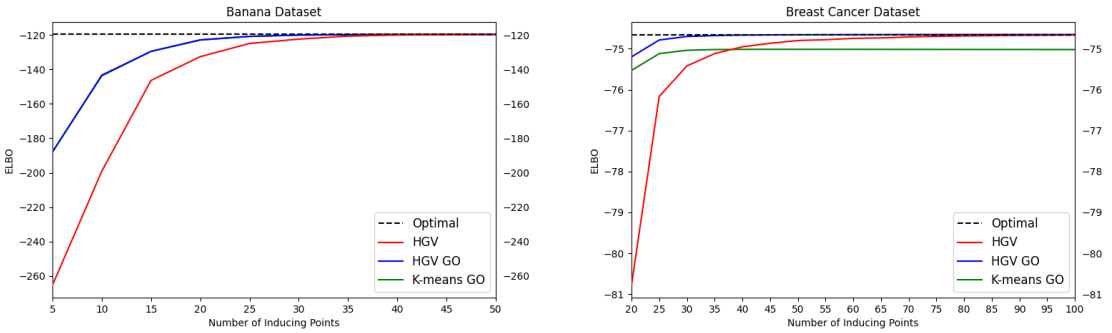

Figure 5.8: PGPR with $\boldsymbol{Z}$ chosen by HGV with and without gradient-based optimisation, and with $\boldsymbol{Z}$ chosen by gradient-optimised k-means, on the Banana (N=400) and Breast Cancer (N=569) datasets. The value of the variational lower bound and the number of inducing points are plotted on the vertical and horizontal axes respectively, and the optimal non-sparse result is denoted by the dashed black line.

The results for the other datasets can be found in Appendix D.2.2. Clearly we see that gradient-based optimisation does help and allows sparser models than HGV. We do notice however that gradient-based optimisation provides little benefit when the number of inducing points is sufficiently large, provided that the inducing points are well placed. In the previous sections, we showed that HGV does indeed place the inducing points in good locations and closely approximates the exact Gaussian process with only a very small number of inducing points. On the other hand, we saw that k-means does not reliably place the inducing points in good locations.

We note that K-means GO often achieves an inferior ELBO to that of HGV GO, due to the suboptimality of its initialisations. Whilst K-means GO is sometimes able to

perform competitively against HGV GO, it often gets stuck in a local optimum in datasets such as Breast Cancer or Crabs. Therefore it is clear that HGV GO outperforms the state-of-the-art K-means GO, and that even the non-optimised HGV often outperforms the optimised state-of-the-art, provided M is large enough.

It is therefore questionable whether the extra computational expense is worth it; for large datasets even a small M can be very expensive and slow to optimise. We also note that even though gradient-based optimisation is able to achieve sparser models, there is no way to automatically select the number of inducing points in this setting - this was a key benefit of HGV. Whilst we recognise the performance benefits of gradient-based optimisation, we believe that at this point HGV strikes the best balance between computational efficiency, accuracy and simplicity. We will however perform a more thorough quantitative analysis in subsequent sections.

5.7 Quantitative Results

In this section we analyse the quantitative differences between our methods and the current state-of-the-art. Namely, we investigate whether PGPR can outperform SVGP in the sparse case, with the supposition that the sparsity and stability afforded by PGPR and our initialisation scheme may be able to outperform SVGP in practice.

To conduct this investigation we train various models on ten datasets and compare their performance on a number of metrics. We randomly sample 10% of each dataset as a held-out test set and compute the ELBO, predictive accuracy and negative test log-likelihood. Each experiment is run ten times and we compute the maximum, minimum, median, mean and standard deviation as a distribution over each metric - this means that we have a total of fifteen criteria. For each dataset, we award a point to the model that achieves the best score in each criterion, and aggregate the total number of points for each model as a measure of general performance. We also assess the rankings of each model across the datasets as a measure of reliability.

The number of inducing points for each dataset is chosen by inspecting convergence of the sparsity graphs in the previous section. In general, we choose extremely sparse models and find that PGPR can often perform close to optimally if given a suitable initialisation scheme, whereas SVGP often struggles. We run all experiments on a single CPU or GPU and limit the number of inducing points to $M = 300$ due to computational constraints. We repeat this experiment with and without gradient-based optimisation of the inducing points in order to assess any differences in performance.

We do not investigate the time-based performance of each model as our experimental setup is not sufficiently accurate for this, however we notice that our methods converge significantly faster than SVGP when provided with suitable convergence limits. [Wenzel et al. (22)] present a time-based analysis of their Pólya-Gamma model and show that the closed-form computations provide significant speedups, and [Burt et al. (10)] show that a reinitialisation scheme often converges significantly faster than gradient-based optimisation of the inducing points.

5.7.1 Fixed Inducing Points

We first evaluate fixed inducing point selection methods, where by 'fixed' we mean to say that the inducing points are not optimised through gradient-based optimisation. We compare the state-of-the-art SVGP K-means to our model PGPR paired with four different initialisation procedures. The five models that we evaluate are:

- **SVGP K-means**: $\boldsymbol{Z}$ chosen through k-means clustering.
- **PGPR Uniform**: $\boldsymbol{Z}$ chosen through uniform subsampling.
- **PGPR K-means**: $\boldsymbol{Z}$ chosen through k-means clustering.
- **PGPR GV**: $\boldsymbol{Z}$ chosen through greedy variance reinitialisation.
- **PGPR HGV**: $\boldsymbol{Z}$ chosen through heteroscedastic greedy variance reinitialisation.

We begin by assessing the performance of each model across three metrics: the ELBO, the predictive accuracy and the negative test log-likelihood. As discussed earlier, we compute a distribution over each metric so that there are a total number of fifteen criteria for each model on each dataset, and the model that achieves the best performance in each criterion is awarded a point. For example, the model with the highest median accuracy on the Banana dataset wins one point, and the model with the lowest ELBO standard deviation on the Crabs dataset wins one point. In the table below we denote the aggregate number of points awarded to each model across the ten datasets. The complete set of results can be found in Appendix D.3.1.

Method	ELBO	Acc.	NLL	Total
SVGP K-means	**23**	12	13	48
PGPR Uniform	1	14	3	18
PGPR K-means	5	12	1	22
PGPR GV	6	22	8	36
PGPR HGV	15	**33**	**24**	**72**

Table 5.2: Points awarded to each model for three metrics across ten datasets. We assess the SVGP benchmark against PGPR paired with four different fixed initialisation methods. The model with the most points in each category is emboldened.

As is no surprise, we see that the SVGP model wins the majority of the points in the ELBO category; this is because the PGPR model has a lower theoretically optimal ELBO. The only reason that a PGPR model might attain a better ELBO would be if SVGP had failed to converge to the global optimum, or if the inducing point placement of the PGPR model was superior. The fact that PGPR HGV attains a significant amount of points in this category is impressive, and suggests that its placement of inducing points if often superior.

In the other categories we see that PGPR HGV achieves the most points in both the accuracy and NLL categories by a large margin. In terms of aggregate performance, we see that PGPR HGV achieves 50% more points than its next competitor and is clearly the best performing model in general. This is very promising and highlights the superior placement of inducing points and stability afforded by PGPR HGV.

In order to more accurately quantify the reliability of each method, we assign model rankings for each dataset and inspect the consistency of each model's rank. The rankings are decided by the number of points attained by each model on each dataset, and we denote the rankings for each model over the ten datasets below. To compute an aggregate score for each model we award one to five points, where a model that wins all ten datasets would score $10 * 5 = 50$ points.

Method	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	Score
SVGP K-means	4	2	1	0	3	34/50
PGPR Uniform	0	0	3	5	2	21/50
PGPR K-means	0	0	5	5	0	25/50
PGPR GV	0	6	1	2	1	32/50
PGPR HGV	6	4	0	0	0	**46/50**

Table 5.3: Rankings attained for each model across ten datasets. We assess the SVGP benchmark against PGPR paired with four different fixed initialisation methods. The model with the highest aggregate ranking is emboldened.

Again, we straight away see that PGPR HGV wins by a large margin. Not only does it have the largest number of points, but it is consistently placed in the top two models for every single dataset, winning the majority of datasets. Whilst SVGP wins the other datasets, we see that it is extremely unreliable and frequently performs the worst out of all models! It appears that the gradient-based optimisation of $q(\boldsymbol{u})$ in SVGP can be incredibly unstable which often leads to catastrophic failures. The complete set of results in Appendix D.3.1 provide more detail, from which we can see that even though SVGP has the potential to outperform PGPR, it is very inconsistent. Consistency is a prevalent issue in the optimisation of Gaussian processes, so the fact that PGPR HGV performs so well *and* so consistently is very desirable.

We conclude from this quantitative assessment that PGPR paired with heteroscedastic greedy variance reinitialisation is the most performant and reliable fixed method, beating the state-of-the-art SVGP K-means across a wide variety of metrics. Unless we are able to improve the stability and inducing point selection of SVGP models, PGPR appears to be a promising alternative.

5.7.2 Gradient-Based Optimisation

We again move on to answer the question: 'Does gradient-based optimisation help?'. In order to answer this, we compare the best performing fixed method (PGPR HGV) against two methods where the inducing points are optimised. In Section 5.6.2, we showed that gradient-based optimisation does not provide significant benefits over HGV, provided that M is large enough. In this section, we wish to more quantitatively evaluate whether gradient-optimised SVGP K-means or PGPR K-means will be able to outperform PGPR HGV across our distribution of metrics. We therefore evaluate the following models:

- **SVGP GO**: $\boldsymbol{Z}$ initialised by k-means clustering, and then gradient-optimised.
- **PGPR GO**: $\boldsymbol{Z}$ initialised by k-means clustering, and then gradient-optimised.
- **PGPR HGV**: $\boldsymbol{Z}$ chosen through heteroscedastic greedy variance reinitialisation.

We repeat the analysis from the previous section, and begin by assessing the performance of each model across three metrics. We inspect the aggregate points awarded to each model across all ten datasets, with the complete results in Appendix D.3.2.

Method	ELBO	Acc.	NLL	Total
SVGP GO	**29**	16	13	58
PGPR GO	4	23	13	40
PGPR HGV	17	**29**	**24**	**70**

Table 5.4: Points awarded to each model for three metrics across ten datasets. We assess the SVGP gradient-optimised benchmark against two variants of PGPR. The model with the most points in each category is emboldened.

For the same reasons as in the previous section, we are not surprised that the SVGP model wins the majority of the points in the ELBO category. It is impressive however that PGPR HGV is still able to win a significant proportion of these points against two gradient-optimised models - this is generally due to the instability of gradient-based methods which causes the distribution over results to have large variance, as well as due to the poor inducing point initialisations from k-means.

In the accuracy and NLL categories, PGPR HGV again significantly outperforms its competitors. We see that it achieves 21% more points than its closest competitor, and is the best performing model in general. This shows that our heteroscedastic reinitialisation scheme is indeed able to outperform even directly optimised models!

In order to quantify the reliability of each method, we again assign model rankings for each dataset as in the previous section. As we now only have three models, each model is awarded one to three points per dataset depending on its ranking - the maximum aggregate score is therefore $10 * 3 = 30$ points. We denote the rankings for each model over the ten datasets below.

Method	1^{st}	2^{nd}	3^{rd}	Score
SVGP GO	3	4	3	20/30
PGPR GO	2	4	4	18/30
PGPR HGV	5	3	2	**23/30**

Table 5.5: Rankings attained for each model across ten datasets. We assess the SVGP gradient-optimised benchmark against two variants of PGPR. The model with the highest aggregate ranking is emboldened.

Again we see that PGPR HGV comes out on top, but the results are definitely much closer than in the previous section. Having reviewed these results, we see two main reasons that allow PGPR HGV to outperform the gradient-based models:

- PGPR models in general are much more stable than SVGP models. The analytical optimisation of $q(\boldsymbol{u})$ removes a layer of gradient-optimisation which often causes SVGP to catastophically fail due to slow convergence or local optima. For example, on the Twonorm and Ringnorm datasets we see that SVGP GO has real trouble in achieving a satisfactory result.
- Gradient-based optimisation of the inducing points complicates the optimisation surface of a model. We often see that the GO models have trouble converging to the global optimum, or that they take an infeasible amount of time

to converge. For example, on the Crabs dataset both PGPR GO and SVGP GO seriously underperform; even increasing the number of training iterations by a magnitude of ten times is insufficient for either model to reach convergence.

These findings reinforce the results of [Burt et al. (10)], and highlight the poor convergence properties of gradient-based models. Whilst we might expect such models to outperform fixed methods, it is clear that their slow convergence and instability is a large issue. We therefore conclude that there is no clear benefit to gradient-based optimisation of the inducing points, provided that M is large enough and that the inducing points are placed in good locations. We believe that out of all evaluated methods, our heteroscedastic greedy variance reinitialisation scheme indeed finds the best inducing points through coordinate ascent. We also believe that pairing this scheme with our efficient PGPR model leads to the best known performance for general Gaussian process classification, outperforming the current state-of-the-art.

5.8 Summary

In this chapter we performed a thorough qualitative and quantitative analysis of our model, PGPR, as well as our inducing point selection method, HGV. We ran experiments to compare these to the state-of-the-art, and provided recommendations for robust Gaussian process classification. Here we summarise the main contributions:

- We briefly detailed a dynamic jitter scheme for accurate and robust Gaussian process optimisation, and discussed the benefits of such a method.
- We investigated the qualitative behaviour of our model, and visually demonstrated that the Pólya-Gamma variance is lowest at the predictive boundaries. We showed that HGV would therefore place inducing points close to the boundaries, as per our hypothesis.
- We investigated the qualitative differences between our model and the state-of-the-art in the non-sparse case, and found that our model is able to closely approximate the true Bernoulli model.
- We visually investigated the qualitative differences between various inducing point selection methods on the Banana dataset. We showed that the inducing points chosen by SVGP GO and PGPR HGV closely align with our hypothesis.
- We showed that out of all fixed inducing point selection methods for PGPR, HGV permits the sparsest models. We also showed that further optimisation of the inducing points is not necessary, provided that M is large enough.
- We performed a quantitative analysis of various models and found that PGPR HGV outperformed all fixed inducing point models, including the current state-of-the-art. We further showed that PGPR HGV can even outperform models in which the inducing points are optimised through gradient-based optimisation.
- Finally, we concluded that out of all evaluated models, PGPR HGV performs the best for general Gaussian process classification. We therefore have strong evidence that our hypothesis is correct, and that our analytical solution for GP classification is able to outperform the unreliable state-of-the-art.

Chapter 6

Ethical Considerations

Before presenting our conclusions, we take a brief detour in this chapter to discuss a number of ethical and societal issues surrounding our project area.

Measurable uncertainty is held to be a key factor in the growing adoption of automated decision-making systems in society. As mentioned in the introduction, many machine learning methods operate as black box systems with little ability to accurately convey confidence in their decisions (often being overconfident). As a result of ethical concerns and strict regulations, this has proved to be a barrier to the introduction of such systems in many industries. For example, imagine a self-driving car that is overly confident in its path across a busy intersection, or an automated clinician that is overly confident in its diagnoses of patients. If we wish for the integration of these systems into society, it is essential that we have the ability to interpret their results and thus reason about their decisions.

Recent work has shed light on these issues and many researchers are either working on solutions to improve machine learning explainability, or have thought it pertinent to discuss the ethical considerations concerning these systems. [Lo Piano (29)] surveys the current state of ethics in machine learning, detailing ethical dimensions affected by algorithm-driven decision-making, and providing thoughts on possible ways forward. [Grote and Berens (30)] provide discussion on ethical considerations in healthcare with an aim to 'lay the grounds for further ethical reflection of the opportunities and pitfalls of machine learning for enhancing decision-making'. They provide compelling arguments on issues surrounding the attribution of accountability for clinicians working with machine learning algorithms. For example, if a clinician defers their decision to that of an automated counterpart and this decision subsequently causes harm to a patient - is the clinician to blame?

We believe that the above issues can be ameliorated by uncertainty-aware methods such as Gaussian processes. Firstly, due to their accurate uncertainty bounds, we are able to hold higher confidence in rejecting or accepting provided decisions. Whilst of course we cannot guarantee that these decisions are correct, they at least grant us the ability to reason about them and make informed actions. Secondly, by their inherent Bayesian nature, we have less concern for dangerously overconfident or overfit solutions[1] - under sensible priors, GPs naturally find simpler solutions [Rasmussen and Ghahramani (3)]. We believe that these benefits provide a path forward and may provide recourse in the search for interpretable machine learning systems.

[1] We 'average' over *all* possible model parameters, instead of relying on uncertain point estimates.

Chapter 7

Conclusion

Having arrived at the end of our report, we conclude with a summary of our contributions and findings. Throughout this work we presented two main contributions:

1. We derived a novel heteroscedastic model for Gaussian process classification, in which the variational distribution over inducing points is analytically optimised. This is in contrast to the current state-of-the-art where the variational distribution is optimised through unreliable gradient-based methods. We showed that our model permits accurate inference, whilst also being both significantly faster and more robust than the current state-of-the-art.

2. We proposed a novel heteroscedastic inducing point selection scheme for use in both Gaussian process regression and classification. We performed a thorough quantitative analysis for the case of classification, and showed that pairing this scheme with our model allows us to consistently outperform the current state-of-the-art across a wide variety of metrics and datasets.

We believe that these results present a very promising step forward for the field of Gaussian process classification. Not only is our model significantly faster and more reliable, but our selection method also allows us to automatically select the number of inducing points required for a sufficiently accurate approximation. We showed that our scheme is able to find extremely sparse models with large computational savings, and that the placement of the inducing points consistently produces superior models to those produced through alternative methods. Finally, we showed that gradient-based optimisation of the inducing points is not required, provided that we have a sufficiently large number of inducing points; we stated our preference in this case for the use of our efficient coordinate ascent reinitialisation scheme.

We also provide an extensive suite of mathematical derivations and results in the appendices, which theoretically motivate the aforementioned contributions. Additionally, we provide detailed implementations in a GitHub repository, available *here*.

7.1 Future Work

With these results and contributions in place, we hold a positive outlook on the direction of future work in this area. In this final section, we present our thoughts on some promising avenues for subsequent research.

7.1.1 Multi-Class Classification

In this work, we focused solely on the task of binary classification. Having seen the success of our model, an extension to the multi-class and/or multi-label case would be valuable in order to widen the applicability of the Pólya-Gamma framework.

7.1.2 Merged Models

Throughout this work we discussed that the use of the true Bernoulli model is theoretically preferable to the use of our Pólya-Gamma approximation. However, in practice we encountered the issues of unreliable optimisation and poor inducing point selection in the Bernoulli model; we therefore wonder if it might be possible to merge the models in order to benefit from the best of both.

Such a scheme might involve selecting the inducing points using PGPR, and then passing these to an SVGP model which is then optimised. Theoretically if the SVGP model were to converge to its global optimum, it would attain a superior ELBO to that of PGPR. This is promising, but there are two main obstacles:

1. This would involve training two models, and might require multiple reinitialisations to find appropriate inducing points. In order to avoid this computational expense we could use an interleaved training scheme where we: reinitialise the inducing points using PGPR, pass these to SVGP, optimise SVGP, pass the optimised hyperparameters back to PGPR, and then repeat until convergence. This means that we only have to directly optimise one model, under the assumption that the optimal hyperparameters are similar for both models.

2. We are still susceptible to the unreliable gradient-based optimisation of SVGP. We might be able to ameliorate this to some degree by initialising $q(\boldsymbol{u})$ to the analytical solution of PGPR, however this is not guaranteed to fix all issues.

Having briefly implemented this we find minor improvements over PGPR HGV, but do not provide a detailed analysis. Whether the additional computational expense and complication is worth it depends on the use case, however we do think that this might yet prove beneficial with a better-tuned optimisation scheme.

7.1.3 Inducing Point Limits

We mentioned that our inducing point selection methods allow us to automatically select the number of inducing points by inspecting the trace term in the selection criterion. We stated that when the trace term is sufficiently small, we can stopping accepting new inducing points as we know that the approximation is close to exact.

We do not however quantify how small 'sufficiently small' is. It turns out that this is a difficult question to answer - in most cases we can set a small threshold such as 10^{-3}, and this is small enough to guarantee an accurate approximation. However, as the size of the dataset increases, we often see that the threshold needs to increase as well. By only using a fixed threshold, we often take too many or too few inducing points. In order to improve the generalisation of these selection schemes, future work might find a way to automatically select this threshold for any dataset.

In the case of Gaussian process regression, the trace criterion is an upper bound on the KL divergence between the approximate and exact posterior. [Burt et al. (10)] discuss bounds on the error between the approximate and exact posteriors' mean and variance, and relate this to the KL divergence. We conjecture that this could be extended and related to the trace term such that we are able to say: 'Set the trace threshold such that we allow no more than 1% error in the posterior mean'. This would provide a theoretically and practically motivated criterion for selecting the threshold for any dataset, agnostic of the size or complexity of the dataset.

7.1.4 Optimisation Hyperparameters

Additional complexity of PGPR comes from the fact that we have a double optimisation loop. Not only do we have an outer reinitialisation loop, but we also have the inner optimisation loop discussed in Section 4.1.3. Determining the number of iterations required in both loops for convergence is difficult - setting low limits makes PGPR very fast to train but it may not converge, whereas setting high limits better ensures convergence but may make PGPR put in more work than it needs to.

Currently we set simple convergence thresholds on the ELBO, and use an upper limit in case the model has not converged by some predefined number of iterations. We believe that a way to automatically determine the optimal number of iterations, or a way to better check for convergence in each loop, would be valuable in improving the efficiency and reliability of PGPR.

7.1.5 Deep Gaussian Processes

Deep Gaussian processes (DGPs) are multi-layered extensions of the single-layered Gaussian processes that we discuss in this report. Similarly to deep neural networks, DGPs present the opportunity for richer models that are able to represent an extremely flexible class of functions. However, the issues surrounding the instability of gradient-based optimisation of the variational distribution are only elevated when we extend our models to multiple layers - optimisation can be very expensive and unstable. Presenting a DGP model for classification in which the variational distribution is analytically optimised might require a substantial amount of work, but would likely be a valuable addition to the Gaussian process classification framework.

References

[1] Ghahramani Z. Bayesian non-parametrics and the probabilistic approach to modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 2013;371(1984):20110553. Available from: https://dx.doi.org/10.1098/rsta.2011.0553. 8

[2] Ghahramani Z. Probabilistic machine learning and artificial intelligence. *Nature*. 2015;521(7553):452–459. Available from: https://doi.org/10.1038/nature14541. 8

[3] Rasmussen CE, Ghahramani Z. Occam's Razor. In: Leen TK, Dietterich TG, Tresp V, editors. *Proceedings of the 13th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: The MIT Press; 2001. p. 276–282. Available from: https://papers.nips.cc/paper/2000/file/0950ca92a4dcf426067cfd2246bb5ff3-Paper.pdf. 8, 51

[4] MacKay DJC. *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press; 2003. Available from: https://www.inference.org.uk/itprnn/book.pdf. 8

[5] Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: The MIT Press; 2006. Available from: http://www.gaussianprocess.org/gpml/chapters/RW.pdf. 8, 12, 17

[6] Neal RM. *Bayesian Learning for Neural Networks (PhD thesis)*. Department of Computer Science, University of Toronto. 1994. Available from: https://www.cs.toronto.edu/~radford/ftp/thesis.pdf. 9, 12

[7] Quiñonero-Candela J, Rasmussen CE. A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research*. 2005;6:1939–1959. Available from: https://dl.acm.org/doi/pdf/10.5555/1046920.1194909. 9, 14

[8] Titsias M. Variational Learning of Inducing Variables in Sparse Gaussian Processes. In: van Dyk D, Welling M, editors. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*. Clearwater Beach, Florida, USA: PMLR; 2009. p. 567–574. Available from: https://proceedings.mlr.press/v5/titsias09a/titsias09a.pdf. 9, 10, 14, 17, 19, 21, 23, 26, 69

[9] Snelson E, Ghahramani Z. Sparse Gaussian Processes using Pseudo-inputs. In: Weiss Y, Schölkopf B, Platt JC, editors. *Proceedings of the 18th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: The MIT Press; 2006. p. 1257–1264. Available from: https://proceedings.neurips.cc/paper/2005/file/4491777b1aa8b5b32c2e8666dbe1a495-Paper.pdf. 9, 21, 23, 26

[10] Burt DR, Rasmussen CE, van der Wilk M. Convergence of Sparse Variational Inference in Gaussian Processes Regression. *Journal of Machine Learning Research*. 2020;21(131):1–63. Available from: https://jmlr.org/papers/volume21/19-1015/19-1015.pdf. 9, 15, 21, 26, 36, 46, 50, 54, 62

[11] Matthews AGDG. *Scalable Gaussian process inference using variational methods (PhD thesis)*. Department of Engineering, University of Cambridge. 2016. Available from: http://mlg.eng.cam.ac.uk/matthews/thesis.pdf. 12, 14, 15

[12] van der Wilk M, Dutordoir V, John ST, Artemev A, Adam V, Hensman J. A Framework for Interdomain and Multioutput Gaussian Processes. *arXiv preprint*. 2020. Available from: https://arxiv.org/pdf/2003.01115.pdf. 13, 14, 15, 16

[13] van der Wilk M. *Sparse Gaussian Process Approximations and Applications (PhD thesis)*. Department of Engineering, University of Cambridge. 2018. Available from: https://doi.org/10.17863/CAM.35660. 14, 45

[14] Minka TP. Expectation Propagation for Approximate Bayesian Inference. In: Breese J, Koller D, editors. *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2001. p. 362–369. Available from: https://dl.acm.org/doi/pdf/10.5555/2074022.2074067. 14

[15] Bui TD, Yan J, Turner RE. A Unifying Framework for Gaussian Process Pseudo-Point Approximations using Power Expectation Propagation. *Journal of Machine Learning Research*. 2017;18(1):3649–3720. Available from: https://www.jmlr.org/papers/volume18/16-603/16-603.pdf. 14

[16] Blei DM, Kucukelbir A, McAuliffe JD. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*. 2017;112(518):859–877. Available from: https://doi.org/10.1080/01621459.2017.1285773. 14, 15

[17] Hastings WK. Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*. 1970;57(1):97–109. Available from: https://doi.org/10.1093/biomet/57.1.97. 15

[18] Matthews AGDG, Hensman J, Turner RE, Ghahramani Z. On Sparse Variational Methods and the Kullback-Leibler Divergence between Stochastic Processes. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*; 2016. p. 231–239. Available from: http://proceedings.mlr.press/v51/matthews16.pdf. 16

[19] Hensman J, Fusi N, Lawrence ND. Gaussian Processes for Big Data. In: Nicholson A, Smyth P, editors. *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence*. Arlington, Virginia, USA: AUAI Press; 2013. p. 282–290. Available from: https://auai.org/uai2013/prints/papers/244.pdf. 17

[20] Hensman J, Matthews AGDG, Ghahramani Z. Scalable Variational Gaussian Process Classification. In: Lebanon G, Vishwanathan SVN, editors. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*.

San Diego, California, USA: PMLR; 2015. p. 351–360. Available from: https://proceedings.mlr.press/v38/hensman15.pdf. 18

[21] Polson NG, Scott JG, Windle J. Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables. *Journal of the American Statistical Association*. 2013;108(504):1339–1349. Available from: https://doi.org/10.1080/01621459.2013.829001. 19, 28

[22] Wenzel F, Galy-Fajou T, Donner C, Kloft M, Opper M. Efficient Gaussian Process Classification Using Pólya-Gamma Data Augmentation. *Proceedings of the AAAI Conference on AI*. 2019;33(01):5417–5424. Available from: https://ojs.aaai.org/index.php/AAAI/article/view/4481. 19, 20, 28, 30, 46, 65

[23] Kulesza A, Taskar B. k-DPPs: Fixed-Size Determinantal Point Processes. In: *Proceedings of the 28th International Conference on International Conference on Machine Learning*. Madison, WI, USA: Omnipress; 2011. p. 1193–1200. Available from: https://icml.cc/2011/papers/611_icmlpaper.pdf. 22

[24] Titsias MK. Variational Inference for Gaussian and Determinantal Point Processes. *Advances in Variational Inference (NIPS)*. 2014. Available from: http://www2.aueb.gr/users/mtitsias/papers/titsiasNipsVar14.pdf. 24, 61

[25] de G Matthews AG, 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;18(40):1–6. Available from: http://jmlr.org/papers/v18/16-537.html. 24, 31, 35, 59

[26] Gibbs MN, Mackay DJC. Variational Gaussian Process Classifiers. *IEEE Transactions on Neural Networks*. 2000;11(6):1458–1464. Available from: http://www.inference.org.uk/mackay/vgc.pdf. 30

[27] Vanschoren J, van Rijn JN, Bischl B, Torgo L. OpenML: Networked Science in Machine Learning. *SIGKDD Explorations*. 2013;15(2):49–60. Available from: http://doi.acm.org/10.1145/2641190.2641198. 37

[28] Dua D, Graff C. *UCI Machine Learning Repository*. University of California, Irvine, School of Information and Computer Sciences. 2017. Available from: http://archive.ics.uci.edu/ml. 37

[29] Lo Piano S. Ethical principles in machine learning and artificial intelligence: cases from the field and possible ways forward. *Humanities and Social Sciences Communications*. 2020;7(9). Available from: https://doi.org/10.1057/s41599-020-0501-9. 51

[30] Grote T, Berens P. On the ethics of algorithmic decision-making in healthcare. *Journal of Medical Ethics*. 2020;46(3):205–211. Available from: https://jme.bmj.com/content/medethics/46/3/205.full.pdf. 51

[31] Bauer M, van der Wilk M, Rasmussen CE. Understanding Probabilistic Sparse Gaussian Process Approximations. In: Lee DD, von Luxburg U, Garnett R, Sugiyama M, Guyon I, editors. *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc.; 2016. p. 1533–1541. Available from: https://proceedings.neurips.cc/paper/2016/file/7250eb93b3c18cc9daa29cf58af7a004-Paper.pdf. 63

Appendix A

Pólya-Gamma

A.1 Logit link as a function of cosh

The logit link function used for Gaussian process classification can be written in a form which contains the *cosh* function. This allows us to investigate connections to the Pólya-Gamma random variables, $\omega \sim PG(b, 0),\ b > 0$, whose moment generating function involves the *cosh* function.

$$\begin{aligned}
\sigma(z) &= \frac{1}{1+\exp(-z)} \\
&= \frac{\exp(0)}{\exp(0)+\exp(-z)} \\
&= \frac{\exp(-\frac{1}{2}z)\exp(\frac{1}{2}z)}{\exp(-\frac{1}{2}z)(\exp(\frac{1}{2}z)+\exp(-\frac{1}{2}z))} \\
&= \frac{\exp(\frac{1}{2}z)}{\exp(\frac{1}{2}z)+\exp(-\frac{1}{2}z)} \\
&= \frac{\exp(\frac{1}{2}z)}{2\cosh(\frac{1}{2}z)}
\end{aligned}$$

Appendix B

Heteroscedastic Regression

B.1 Efficient Variational Lower Bound and Predictions

Here we derive an efficient implementation of the variational lower bound and predictive distribution for heteroscedastic GP regression. We follow a similar derivation as is given for the homoscedastic case in GPflow [Matthews et al. (25)], and also provide a GPflow-based software implementation, available *here*.

B.1.1 Variational Lower Bound

For the variational lower bound, we start from the ELBO

$$\begin{aligned}\mathcal{L} &= \log \mathcal{N}(\boldsymbol{y}; \boldsymbol{0}, \boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}) - \frac{1}{2} Tr(\boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{K}}_{ff}) \\ &= -\frac{N}{2} \log 2\pi - \frac{1}{2} \log |\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| - \frac{1}{2} \boldsymbol{y}^\top (\boldsymbol{Q}_{ff} + \boldsymbol{\Lambda})^{-1} \boldsymbol{y} - \frac{1}{2} Tr(\boldsymbol{\Lambda}^{-1} \tilde{\boldsymbol{K}}_{ff})\end{aligned}$$

with $\boldsymbol{Q}_{ff} = \boldsymbol{K}_{fu} \boldsymbol{K}_{uu}^{-1} \boldsymbol{K}_{uf}$ and $\tilde{\boldsymbol{K}}_{ff} = \boldsymbol{K}_{ff} - \boldsymbol{Q}_{ff}$. In order to avoid the computational cost of the $\mathcal{O}(N^3)$ matrix inversion and determinant, we investigate cheaper computations. We first use the Woodbury matrix identity to form a cheaper inversion

$$(\boldsymbol{Q}_{ff} + \boldsymbol{\Lambda})^{-1} = \boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu} (\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu})^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1}$$

We then perform a Cholesky decomposition $\boldsymbol{K}_{uu} = \boldsymbol{L}\boldsymbol{L}^\top$ and rotate the above expression by $\boldsymbol{L}$ in order to obtain a better conditioned matrix

$$\begin{aligned}&\boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu} (\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu})^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \\ &= \boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu} \boldsymbol{L}^{-\top} \boldsymbol{L}^\top (\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu})^{-1} \boldsymbol{L} \boldsymbol{L}^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \\ &= \boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu} \boldsymbol{L}^{-\top} [\boldsymbol{L}^{-1} (\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu}) \boldsymbol{L}^{-\top}]^{-1} \boldsymbol{L}^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \\ &= \boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu} \boldsymbol{L}^{-\top} [\boldsymbol{I} + \boldsymbol{L}^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1} \boldsymbol{K}_{fu} \boldsymbol{L}^{-\top}]^{-1} \boldsymbol{L}^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-1}\end{aligned}$$

We define $\boldsymbol{A} \triangleq \boldsymbol{L}^{-1} \boldsymbol{K}_{uf} \boldsymbol{\Lambda}^{-\frac{1}{2}}$ and $\boldsymbol{B} \triangleq \boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^\top$ to further yield

$$\boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{A}^\top [\boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^\top]^{-1} \boldsymbol{A} \boldsymbol{\Lambda}^{-\frac{1}{2}} = \boldsymbol{\Lambda}^{-1} - \boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{A}^\top \boldsymbol{B}^{-1} \boldsymbol{A} \boldsymbol{\Lambda}^{-\frac{1}{2}}$$

which gives us the final form of our cheap inversion. Note that whilst we still have to invert the $N \times N$ matrix $\boldsymbol{\Lambda}$, this is a diagonal matrix and is therefore cheap to invert.

Next we apply the Matrix Determinant Lemma to form a cheaper determinant

$$\begin{aligned}|\boldsymbol{Q}_{ff}+\boldsymbol{\Lambda}| &= |\boldsymbol{K}_{uu}+\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu}||\boldsymbol{K}_{uu}^{-1}||\boldsymbol{\Lambda}| \\ &= |\boldsymbol{L}\boldsymbol{L}^{\top}+\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu}||\boldsymbol{L}^{-\top}||\boldsymbol{L}^{-1}||\boldsymbol{\Lambda}| \\ &= |\boldsymbol{I}+\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}||\boldsymbol{\Lambda}| \\ &= |\boldsymbol{I}+\boldsymbol{A}\boldsymbol{A}^{\top}||\boldsymbol{\Lambda}| \\ &= |\boldsymbol{B}||\boldsymbol{\Lambda}|\end{aligned}$$

With these two definitions in place, we can simplify the original bound

$$\begin{aligned}\mathcal{L} &= -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff}| - \frac{1}{2}\boldsymbol{y}^{\top}(\boldsymbol{Q}_{ff}+\boldsymbol{\Lambda})^{-1}\boldsymbol{y} - \frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) \\ &= -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{B}||\boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{y}^{\top}(\boldsymbol{\Lambda}^{-1}-\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{A}^{\top}\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{\Lambda}^{-\frac{1}{2}})\boldsymbol{y} - \frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})\end{aligned}$$

We then perform a Cholesky decomposition $\boldsymbol{B}=\boldsymbol{L}_B\boldsymbol{L}_B^{\top}$ and define $\boldsymbol{c} \triangleq \boldsymbol{L}_B^{-1}\boldsymbol{A}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{y}$

$$\mathcal{L} = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{B}||\boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{y}^{\top}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} + \frac{1}{2}\boldsymbol{c}^{\top}\boldsymbol{c} - \frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$$

Finally we note that we can simplify the trace term as

$$\begin{aligned}\frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) &= \frac{1}{2}[Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{ff}) - Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{Q}_{ff})] \\ &= \frac{1}{2}[Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{ff}) - Tr(\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-\frac{1}{2}})] \\ &= \frac{1}{2}[Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{ff}) - Tr(\boldsymbol{A}^{\top}\boldsymbol{A})] \\ &= \frac{1}{2}[Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{ff}) - Tr(\boldsymbol{A}\boldsymbol{A}^{\top})]\end{aligned}$$

With all of these simplifications, we present our final efficient and stable lower bound

$$\mathcal{L} = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{B}||\boldsymbol{\Lambda}| - \frac{1}{2}\boldsymbol{y}^{\top}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} + \frac{1}{2}\boldsymbol{c}^{\top}\boldsymbol{c} - \frac{1}{2}Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{ff}) + \frac{1}{2}Tr(\boldsymbol{A}\boldsymbol{A}^{\top}) \quad \text{(B.1)}$$

B.1.2 Predictive Distribution

Similar to equation 2.8, we start from the predictive posterior distribution

$$\begin{aligned}q(\boldsymbol{f}^*) &= \mathcal{N}(\boldsymbol{f}^*\,;\,\boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m},\ \boldsymbol{K}_{**}-\boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}(\boldsymbol{K}_{uu}-\boldsymbol{S})\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*}) \\ &= \mathcal{N}(\boldsymbol{f}^*\,;\,\boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m},\ \boldsymbol{K}_{**}-\boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*}+\boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*}))\end{aligned}$$

In order to compute this distribution we need to insert the optimal values of $q^*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u};\boldsymbol{m}^*,\boldsymbol{S}^*)$, which with $\Sigma = \boldsymbol{K}_{uu}+\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu}$ we previously found to be

$$\begin{aligned}\boldsymbol{m}^* &= \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} \\ \boldsymbol{S}^* &= \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}\end{aligned}$$

We first simplify some of the distribution using definitions from the above bound

$$\begin{aligned}\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*\boldsymbol{K}_{uu}^{-1} &= \boldsymbol{K}_{uu}^{-1}(\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu})\boldsymbol{K}_{uu}^{-1} \\ &= \Sigma^{-1}\end{aligned}$$

$$
\begin{aligned}
&= (\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{K}_{fu})^{-1} \\
&= (\boldsymbol{L}\boldsymbol{L}^\top + \boldsymbol{L}\boldsymbol{L}^{-1}(\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{K}_{fu})\boldsymbol{L}^{-\top}\boldsymbol{L}^\top)^{-1} \\
&= (\boldsymbol{L}\boldsymbol{L}^\top + \boldsymbol{L}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{L}^\top)^{-1} \\
&= (\boldsymbol{L}[\boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^\top]\boldsymbol{L}^\top)^{-1} \\
&= (\boldsymbol{L}\boldsymbol{B}\boldsymbol{L}^\top)^{-1} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{L}^{-1}
\end{aligned}
$$

and we also simplify

$$
\begin{aligned}
\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* &= \boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} \\
&= \Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} \\
&= \boldsymbol{K}_{uu}^{-1}\boldsymbol{S}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Lambda}^{-1}\boldsymbol{y} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{y} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{L}_B^{-\top}\boldsymbol{L}_B^{-1}\boldsymbol{A}\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{y} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{L}_B^{-\top}\boldsymbol{c}
\end{aligned}
$$

which gives us the final efficient and stable predictive distribution

$$
\begin{aligned}
q(\boldsymbol{f}^*) &= \mathcal{N}(\boldsymbol{f}^*\,;\, \boldsymbol{K}_{*u}\boldsymbol{L}^{-\top}\boldsymbol{L}_B^{-\top}\boldsymbol{c},\, \boldsymbol{K}_{**} - \boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*} + \boldsymbol{K}_{*u}(\boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{L}^{-1})\boldsymbol{K}_{u*}) \\
&= \mathcal{N}(\boldsymbol{f}^*\,;\, \boldsymbol{K}_{*u}\boldsymbol{L}^{-\top}\boldsymbol{L}_B^{-\top}\boldsymbol{c},\, \boldsymbol{K}_{**} - \boldsymbol{K}_{*u}\boldsymbol{L}^{-\top}[\boldsymbol{I} - \boldsymbol{B}^{-1}]\boldsymbol{L}^{-1}\boldsymbol{K}_{u*})
\end{aligned} \tag{B.2}
$$

B.2 Variational Upper Bound

Here we extend the derivation of the upper bound on the log marginal likelihood given in [Titsias (24)] to the heteroscedastic case.

We know that the true log marginal likelihood has the form

$$
\log p(\boldsymbol{y}) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Lambda} + \boldsymbol{K}_{ff}| - \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{K}_{ff} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y}
$$

and we want to prove the bound

$$
\log p(\boldsymbol{y}) \leq \mathcal{U} = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| - \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff} + Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I} + \boldsymbol{\Lambda})^{-1}\boldsymbol{y}
$$

We start by stating two important properties. Suppose that two positive semi-definite matrices $\boldsymbol{A}, \boldsymbol{B}$ are such that $\boldsymbol{A} \succcurlyeq \boldsymbol{B}$, i.e., $\boldsymbol{A} - \boldsymbol{B}$ is positive semi-definite. It holds that

1. $|\boldsymbol{A}| \geq |\boldsymbol{B}|$
2. If $\boldsymbol{A}^{-1}, \boldsymbol{B}^{-1}$ exist, then $\boldsymbol{B}^{-1} \succcurlyeq \boldsymbol{A}^{-1}$

We also know that $\boldsymbol{K}_{ff} \succcurlyeq \boldsymbol{Q}_{ff}$ by the properties of Schur complements of SPSD matrices. This further implies $\boldsymbol{K}_{ff} + \boldsymbol{\Lambda} \succcurlyeq \boldsymbol{Q}_{ff} + \boldsymbol{\Lambda}$, from which we conclude

$$
-\frac{N}{2}\log 2\pi - \frac{1}{2}|\boldsymbol{K}_{ff} + \boldsymbol{\Lambda}| \leq -\frac{N}{2}\log 2\pi - \frac{1}{2}|\boldsymbol{Q}_{ff} + \boldsymbol{\Lambda}|
$$

Therefore to prove the upper bound it remains to show that $\forall \boldsymbol{y}$

$$
\begin{aligned}
&-\frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{K}_{ff}+\boldsymbol{\Lambda})^{-1}\boldsymbol{y} \leq -\frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff}+Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I}+\boldsymbol{\Lambda})^{-1}\boldsymbol{y} \\
&\implies \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{K}_{ff}+\boldsymbol{\Lambda})^{-1}\boldsymbol{y} \geq \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff}+Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I}+\boldsymbol{\Lambda})^{-1}\boldsymbol{y} \\
&\implies \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{K}_{ff}+\boldsymbol{\Lambda})\boldsymbol{y} \leq \frac{1}{2}\boldsymbol{y}^\top(\boldsymbol{Q}_{ff}+Tr(\tilde{\boldsymbol{K}}_{ff})\boldsymbol{I}+\boldsymbol{\Lambda})\boldsymbol{y}
\end{aligned}
$$

By rearranging the inequality we get

$$
\frac{1}{2}\boldsymbol{y}^\top\tilde{\boldsymbol{K}}_{ff}\boldsymbol{y} \leq \frac{1}{2}Tr(\tilde{\boldsymbol{K}}_{ff})\|\boldsymbol{y}\|_2^2
$$

We take the eigen-decomposition $\tilde{\boldsymbol{K}}_{ff} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top$ and $\boldsymbol{z} = \boldsymbol{U}^\top\boldsymbol{y}$ with $\|\boldsymbol{z}\|_2 = \|\boldsymbol{y}\|_2$

$$
\begin{aligned}
\boldsymbol{y}^\top\tilde{\boldsymbol{K}}_{ff}\boldsymbol{y} &= \boldsymbol{y}^\top\boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^\top\boldsymbol{y} \\
&= \boldsymbol{z}^\top\boldsymbol{D}\boldsymbol{z} \\
&= \Sigma_{n=1}^N \lambda_n z_n^2 \\
&\leq \lambda_{max}\Sigma_{n=1}^N z_n^2 \\
&= \lambda_{max}\|\boldsymbol{y}\|_2^2 \\
&\leq Tr(\tilde{\boldsymbol{K}}_{ff})\|\boldsymbol{y}\|_2^2
\end{aligned}
$$

where λ_n are the eigenvalues of $\tilde{\boldsymbol{K}}_{ff}$ and the final inequality holds as $Tr(\tilde{\boldsymbol{K}}_{ff}) = \Sigma_{n=1}^N \lambda_n$ and $\tilde{\boldsymbol{K}}_{ff}$ is SPSD. This completes the proof of the variational upper bound.

We do not provide an efficient implementation of this bound as in Appendix B.1, as we only use this bound for theoretical analysis.

B.3 Average Case Bounds on the KL Divergence

Using the same notation used to form equation 3.7, we follow the derivations of [Burt et al. (10)] and aim to prove the average case bounds on the KL divergence

$$
\frac{t}{2} \leq \mathbb{E}[\mathcal{KL}[q(f(\boldsymbol{X}),\boldsymbol{u}) \,\|\, p(f(\boldsymbol{X}),\boldsymbol{u}|\boldsymbol{y})]|Z=\boldsymbol{Z},X=\boldsymbol{X}] \leq t
$$

with $t = Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff})$. We denote the density of a Gaussian random variable with mean $\boldsymbol{\mu}$ and covariance Σ, evaluated at $\boldsymbol{y}$ as $n(\boldsymbol{y};\boldsymbol{\mu},\Sigma)$ and start by showing

$$
\begin{aligned}
&\mathbb{E}[\mathcal{KL}[q(f(\boldsymbol{X}),\boldsymbol{u}) \,\|\, p(f(\boldsymbol{X}),\boldsymbol{u}|\boldsymbol{y})]|Z=\boldsymbol{Z},X=\boldsymbol{X}] \\
&= \mathbb{E}[\log p(\boldsymbol{y}) - \mathcal{L}|Z=\boldsymbol{Z},X=\boldsymbol{X}] \\
&= \mathbb{E}[\log n(\boldsymbol{y};\boldsymbol{0},\boldsymbol{\Lambda}+\boldsymbol{K}_{ff}) - \log n(\boldsymbol{y};\boldsymbol{0},\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff}) + \frac{t}{2}|Z=\boldsymbol{Z},X=\boldsymbol{X}] \\
&= \mathbb{E}[\log\frac{n(\boldsymbol{y};\boldsymbol{0},\boldsymbol{\Lambda}+\boldsymbol{K}_{ff})}{n(\boldsymbol{y};\boldsymbol{0},\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})}|Z=\boldsymbol{Z},X=\boldsymbol{X}] + \frac{t}{2} \\
&= \mathcal{KL}[\mathcal{N}(\boldsymbol{y};\boldsymbol{0},\boldsymbol{\Lambda}+\boldsymbol{K}_{ff}) \,\|\, \mathcal{N}(\boldsymbol{y};\boldsymbol{0},\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})] + \frac{t}{2}
\end{aligned} \tag{B.3}
$$

The lower bound, $\frac{t}{2}$, trivially follows by the intrinsic non-negative nature of the KL divergence. The upper bound can be found by upper-bounding the KL term

in equation B.3. We start by using the closed-form solution to the KL divergence between two multivariate Gaussian distributions

$$
\begin{aligned}
&\mathcal{KL}[\mathcal{N}(\boldsymbol{y}; \mathbf{0}, \boldsymbol{\Lambda} + \boldsymbol{K}_{ff}) \,||\, \mathcal{N}(\boldsymbol{y}; \mathbf{0}, \boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})] \\
&= \frac{1}{2}\left(\log \frac{|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}|}{|\boldsymbol{\Lambda} + \boldsymbol{K}_{ff}|} - N + Tr((\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}(\boldsymbol{\Lambda} + \boldsymbol{K}_{ff}))\right) \\
&\leq \frac{1}{2}\left(-N + Tr((\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}(\boldsymbol{\Lambda} + \boldsymbol{K}_{ff}))\right) \\
&= \frac{1}{2}\left(-N + Tr((\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}((\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}) + (\boldsymbol{K}_{ff} - \boldsymbol{Q}_{ff})))\right) \\
&= \frac{1}{2}\left(-N + Tr(\boldsymbol{I}_N) + Tr((\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}\tilde{\boldsymbol{K}}_{ff})\right) \\
&= \frac{1}{2} Tr((\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}\tilde{\boldsymbol{K}}_{ff}) \\
&\leq \frac{1}{2} Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) = \frac{t}{2}
\end{aligned}
$$

where the first inequality holds by the properties discussed in Appendix B.2 and $log(x) \leq 0 \; \forall x \leq 1$, and the final inequality holds as $\boldsymbol{Q}_{ff}$ is SPSD.

Plugging this result back into equation B.3 we get

$$
\mathcal{KL}[\mathcal{N}(\boldsymbol{y}; \mathbf{0}, \boldsymbol{\Lambda} + \boldsymbol{K}_{ff}) \,||\, \mathcal{N}(\boldsymbol{y}; \mathbf{0}, \boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})] + \frac{t}{2} \;\leq\; \frac{t}{2} + \frac{t}{2} \;=\; t
$$

which finalises our upper bound and thus our average case prior analysis.

B.4 Monotonic Improvement of the ELBO

Here we show that the ELBO for heteroscedastic regression is monotonically increasing with respect to the addition of new inducing points.

Following the work of [Bauer et al. (31)], we know that the addition of a new inducing point corresponds to a rank-1 update of the matrix $\boldsymbol{Q}_{ff}$. We can define

$$
\boldsymbol{Q}^{+}_{ff} = \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^{\top}
$$

with $\boldsymbol{Q}^{+}_{ff}$ the approximate covariance matrix after adding a new inducing point, and $\boldsymbol{b}$ the corresponding rank-1 update vector[1]. Noting that the matrix $\boldsymbol{Q}_{ff}$ is the only quantity in the ELBO which depends on the inducing points, we can inspect the difference between the ELBO before and after adding an arbitrary inducing point

$$
\begin{aligned}
2(\mathcal{L}^{+} - \mathcal{L}) = &-\log|\boldsymbol{\Lambda} + \boldsymbol{Q}^{+}_{ff}| + \log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| - \\
&\boldsymbol{y}^{\top}(\boldsymbol{\Lambda} + \boldsymbol{Q}^{+}_{ff})^{-1}\boldsymbol{y} + \boldsymbol{y}^{\top}(\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{y} - \\
&Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}^{+}_{ff}) + Tr(\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{K}}_{ff}) \\
&= -\log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^{\top}| + \log|\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff}| - \\
&\boldsymbol{y}^{\top}(\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^{\top})^{-1}\boldsymbol{y} + \boldsymbol{y}^{\top}(\boldsymbol{\Lambda} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{y} +
\end{aligned}
$$

[1] See [Bauer et al. (31)] for exact definitions, which are extraneous to our needs.

$$Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top)$$

We can use the Matrix Determinant Lemma to simplify the determinant terms

$$\begin{aligned}2(\mathcal{L}^+ - \mathcal{L}) = &-\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) - \log|\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff}| + \log|\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff}| - \\ &\boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff}+\boldsymbol{b}\boldsymbol{b}^\top)^{-1}\boldsymbol{y} + \boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{y} + \\ &Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) \\ &= -\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) - \\ &\boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff}+\boldsymbol{b}\boldsymbol{b}^\top)^{-1}\boldsymbol{y} + \boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{y}\end{aligned}$$

We can use the Sherman-Morrison formula to simplify the inverse terms

$$\begin{aligned}2(\mathcal{L}^+ - \mathcal{L}) &= -\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) - \\ &\quad \boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{y} + \boldsymbol{y}^\top\frac{(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}}{1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}\boldsymbol{y} + \boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{y} \\ &= -\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) + \boldsymbol{y}^\top\frac{(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}}{1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}\boldsymbol{y}\end{aligned}$$

which gives us the final form of the ELBO difference. In order to show that this provides a monotonic increase, we need to show that it is non-negative. We start by looking at the first two terms and note some important properties

$$\begin{aligned}Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) &= \boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b} \\ \log(1+x) &\leq x \\ \boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b} &\leq \boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\end{aligned}$$

from which we can deduce

$$\begin{aligned}&\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) \leq \boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b} \leq \boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b} \\ &\implies -\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) \geq -\boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\end{aligned}$$

and so we can show that the sum of the first two terms is non-negative

$$\begin{aligned}-\log(1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) &\geq -\boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + Tr(\boldsymbol{\Lambda}^{-1}\boldsymbol{b}\boldsymbol{b}^\top) \\ &= -\boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b} + \boldsymbol{b}^\top\boldsymbol{\Lambda}^{-1}\boldsymbol{b} = 0\end{aligned}$$

Now to show that the final term is non-negative we simply show

$$\boldsymbol{y}^\top\frac{(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}}{1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}\boldsymbol{y} = \frac{(\boldsymbol{y}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b})^2}{1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}$$

which is trivially non-negative as the squared numerator is non-negative, and the denominator $1+\boldsymbol{b}^\top(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}\boldsymbol{b} \geq 1$ as $(\boldsymbol{\Lambda}+\boldsymbol{Q}_{ff})^{-1}$ is SPSD.

In the case that the new inducing point is at a duplicate location, the matrix $\boldsymbol{K}_{\boldsymbol{uu}}$ becomes singular and alternative reasoning is required. We omit this possibility as our inducing point selection procedures prevent such a scenario from occurring. This concludes our proof of the monotonic improvement of the ELBO.

Appendix C

Heteroscedastic Classification

C.1 Collapsed Variational Lower Bound

We started from the bound presented by [Wenzel et al. (22)], and directly solved for the optimal $\boldsymbol{m}^*$ and $\boldsymbol{S}^*$ to find the analytical solution of the optimal variational distribution over inducing points

$$q^*(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}; \boldsymbol{m}^*, \boldsymbol{S}^*) = \mathcal{N}(\boldsymbol{u}; \frac{1}{2}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}, \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu})$$

We now plug the parameters of $q^*(\boldsymbol{u})$ back into the original bound to form the collapsed variational lower bound

$$\begin{aligned}\mathcal{L}^* = \frac{1}{2}(&\log|\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*| - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*) - \boldsymbol{m}^{*\top}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* + M - 2N\log 2 + \\ &\boldsymbol{y}^\top\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*) - \\ &\boldsymbol{m}^{*\top}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* + \Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})])\end{aligned}$$

For brevity, we treat each grouping of terms separately. We start with

$$\log|\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*| = \log|\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}| = \log|\Sigma^{-1}\boldsymbol{K}_{uu}| = -\log|\boldsymbol{K}_{uu}^{-1}\Sigma|$$

and then address the trace terms

$$\begin{aligned}&- Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*) - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*) + M \\ =& - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}) - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) - Tr(\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}) + M \\ =& - Tr(\Sigma^{-1}\boldsymbol{K}_{uu}) - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) - Tr(\Sigma^{-1}\boldsymbol{K}_{uu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}) + M \\ =& - Tr(\Sigma^{-1}(\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu})) - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) + M \\ =& - Tr(\Sigma^{-1}\Sigma) - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}) + M \\ =& - Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}})\end{aligned}$$

Finally we address the quadratic terms

$$\begin{aligned}&- \boldsymbol{m}^{*\top}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* + \boldsymbol{y}^\top\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* - \boldsymbol{m}^{*\top}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m}^* \\ =& - \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} + \frac{1}{2}\boldsymbol{y}^\top\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} - \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\ =& - \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{K}_{fu}\Sigma^{-1}(\boldsymbol{K}_{uu} - 2\Sigma + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu})\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}\end{aligned}$$

$$= -\frac{1}{4}\boldsymbol{y}^\top \boldsymbol{K}_{fu}\Sigma^{-1}(\Sigma - 2\Sigma)\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}$$
$$= \frac{1}{4}\boldsymbol{y}^\top \boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y}$$

Collating all of these groupings with the constants, we form the final collapsed bound

$$\mathcal{L}^* = -\frac{1}{2}\log|\boldsymbol{K}_{uu}^{-1}\Sigma| - \frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}) + \frac{1}{8}\boldsymbol{y}^\top \boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} - N\log 2 + \frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})]$$

C.2 Efficient Variational Lower Bound and Predictions

Here we derive an efficient implementation of the variational lower bound and predictive distribution for heteroscedastic GP classification. We follow a similar derivation to the one given in Appendix B.1 for the regression case, and also provide a GPflow-based software implementation, available *here*.

As in the regression case, we define some intermediate variables

$$\begin{aligned}\boldsymbol{K}_{uu} &= \boldsymbol{L}\boldsymbol{L}^\top \\ \boldsymbol{A} &\triangleq \boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}^{\frac{1}{2}} \\ \boldsymbol{B} &\triangleq \boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{L}_B\boldsymbol{L}_B^\top \\ \boldsymbol{c} &\triangleq \frac{1}{2}\boldsymbol{L}_B^{-1}\boldsymbol{A}\boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{y}\end{aligned}$$

C.2.1 Variational Lower Bound

Starting from the ELBO, we treat each grouping of terms individually. We start from the log determinant term and employ the Matrix Determinant Lemma

$$\begin{aligned}\log|\boldsymbol{K}_{uu}^{-1}\Sigma| &= \log|\boldsymbol{L}^{-\top}\boldsymbol{L}^{-1}\Sigma| \\ &= \log|\boldsymbol{L}^{-1}(\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu})\boldsymbol{L}^{-\top}| \\ &= \log|\boldsymbol{I} + \boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}| \\ &= \log|\boldsymbol{I}| + \log|\boldsymbol{\Theta}| + \log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}| \\ &= \log(|\boldsymbol{\Theta}^{\frac{1}{2}}||\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}||\boldsymbol{\Theta}^{\frac{1}{2}}|) \\ &= \log|\boldsymbol{I} + \boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}^{\frac{1}{2}}| \\ &= \log|\boldsymbol{I} + \boldsymbol{A}^\top\boldsymbol{A}| \\ &= \log|\boldsymbol{I} + \boldsymbol{A}\boldsymbol{A}^\top| \\ &= \log|\boldsymbol{B}|\end{aligned}$$

We can then address the trace terms

$$\begin{aligned}-Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}) &= -Tr(\boldsymbol{\Theta}\boldsymbol{K}_{ff}) + Tr(\boldsymbol{\Theta}\boldsymbol{Q}_{ff}) \\ &= -Tr(\boldsymbol{\Theta}\boldsymbol{K}_{ff}) + Tr(\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}^{\frac{1}{2}}) \\ &= -Tr(\boldsymbol{\Theta}\boldsymbol{K}_{ff}) + Tr(\boldsymbol{A}^\top\boldsymbol{A})\end{aligned}$$

$$= -Tr(\boldsymbol{\Theta}\boldsymbol{K}_{ff}) + Tr(\boldsymbol{A}\boldsymbol{A}^\top)$$

The quadratic terms can be simplified as

$$\begin{aligned}
\boldsymbol{y}^\top \boldsymbol{K}_{fu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} &= \boldsymbol{y}^\top \boldsymbol{K}_{fu}(\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu})^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\
&= \boldsymbol{y}^\top \boldsymbol{K}_{fu}(\boldsymbol{L}\boldsymbol{L}^\top + \boldsymbol{L}\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\
&= \boldsymbol{y}^\top \boldsymbol{K}_{fu}(\boldsymbol{L}\boldsymbol{L}^\top + \boldsymbol{L}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\
&= \boldsymbol{y}^\top \boldsymbol{K}_{fu}(\boldsymbol{L}\boldsymbol{B}\boldsymbol{L}^\top)^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\
&= \boldsymbol{y}^\top \boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}\boldsymbol{L}_B^{-\top}\boldsymbol{L}_B^{-1}\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\
&= \boldsymbol{y}^\top \boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{K}_{fu}\boldsymbol{L}^{-\top}\boldsymbol{L}_B^{-\top}\boldsymbol{L}_B^{-1}\boldsymbol{L}^{-1}\boldsymbol{K}_{uf}\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{y} \\
&= \boldsymbol{y}^\top \boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{A}^\top\boldsymbol{L}_B^{-\top}\boldsymbol{L}_B^{-1}\boldsymbol{A}\boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{y} \\
&= 4\boldsymbol{c}^\top\boldsymbol{c}
\end{aligned}$$

Finally we note that the subtracted KL term can be written as

$$\frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})] = \Sigma_{n=1}^{N}[\frac{c_n}{4}\tanh(\frac{c_n}{2}) - \log\cosh(\frac{c_n}{2})]$$

Collating these terms, we present the final efficient and stable lower bound

$$\begin{aligned}
\mathcal{L}^* = &-\frac{1}{2}\log|\boldsymbol{B}| - \frac{1}{2}Tr(\boldsymbol{\Theta}\boldsymbol{K}_{ff}) + \frac{1}{2}Tr(\boldsymbol{A}\boldsymbol{A}^\top) + \frac{1}{2}\boldsymbol{c}^\top\boldsymbol{c} - \\
&N\log 2 + \Sigma_{n=1}^{N}[\frac{c_n}{4}\tanh(\frac{c_n}{2}) - \log\cosh(\frac{c_n}{2})]
\end{aligned} \tag{C.1}$$

C.2.2 Predictive Distribution

In a manner similar to Appendix B.1, we investigate an efficient predictive distribution and show that it is identical to the case of heteroscedastic regression, but with our new intermediate variable definitions. Again we start from equation 2.8

$$\begin{aligned}
q(\boldsymbol{f}^*) &= \mathcal{N}(\boldsymbol{f}^*\,;\, \boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m},\, \boldsymbol{K}_{**} - \boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}(\boldsymbol{K}_{uu} - \boldsymbol{S})\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*}) \\
&= \mathcal{N}(\boldsymbol{f}^*\,;\, \boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{m},\, \boldsymbol{K}_{**} - \boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*} + \boldsymbol{K}_{*u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{u*}))
\end{aligned}$$

We first revisit the optimal parameters of $q^*(\boldsymbol{u})$, which with $\Sigma = \boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu}$ are

$$\begin{aligned}
\boldsymbol{m}^* &= \frac{1}{2}\boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uf}\boldsymbol{y} \\
\boldsymbol{S}^* &= \boldsymbol{K}_{uu}\Sigma^{-1}\boldsymbol{K}_{uu}
\end{aligned}$$

and we aim to show two relations. The first of which is

$$\begin{aligned}
\boldsymbol{K}_{uu}^{-1}\boldsymbol{S}^*\boldsymbol{K}_{uu}^{-1} &= \Sigma^{-1} \\
&= (\boldsymbol{K}_{uu} + \boldsymbol{K}_{uf}\boldsymbol{\Theta}\boldsymbol{K}_{fu})^{-1} \\
&= (\boldsymbol{L}\boldsymbol{L}^\top + \boldsymbol{L}\boldsymbol{L}^{-1}(\boldsymbol{K}_{uf}\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{K}_{fu})\boldsymbol{L}^{-\top}\boldsymbol{L}^\top)^{-1} \\
&= (\boldsymbol{L}\boldsymbol{L}^\top + \boldsymbol{L}\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{L}^\top)^{-1} \\
&= (\boldsymbol{L}\boldsymbol{B}\boldsymbol{L}^\top)^{-1} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{L}^{-1}
\end{aligned}$$

Next we show

$$\begin{aligned}
\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{m}^{*} &= \frac{1}{2}\Sigma^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{y} \\
&= \frac{1}{2}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{S}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{y} \\
&= \frac{1}{2}\boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{L}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{y} \\
&= \frac{1}{2}\boldsymbol{L}^{-\top}\boldsymbol{L}_{B}^{-\top}\boldsymbol{L}_{B}^{-1}\boldsymbol{L}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}^{\frac{1}{2}}\boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{y} \\
&= \frac{1}{2}\boldsymbol{L}^{-\top}\boldsymbol{L}_{B}^{-\top}\boldsymbol{L}_{B}^{-1}\boldsymbol{A}\boldsymbol{\Theta}^{-\frac{1}{2}}\boldsymbol{y} \\
&= \boldsymbol{L}^{-\top}\boldsymbol{L}_{B}^{-\top}\boldsymbol{c}
\end{aligned}$$

This now gives us the efficient and stable predictive distribution, identical to the case of heteroscedastic regression but with our new intermediate variable definitions

$$\begin{aligned}
q(\boldsymbol{f}^{*}) &= \mathcal{N}(\boldsymbol{f}^{*}\,;\,\boldsymbol{K}_{*\boldsymbol{u}}\boldsymbol{L}^{-\top}\boldsymbol{L}_{B}^{-\top}\boldsymbol{c},\,\boldsymbol{K}_{**}-\boldsymbol{K}_{*\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{u}*}+\boldsymbol{K}_{*\boldsymbol{u}}(\boldsymbol{L}^{-\top}\boldsymbol{B}^{-1}\boldsymbol{L}^{-1})\boldsymbol{K}_{\boldsymbol{u}*}) \\
&= \mathcal{N}(\boldsymbol{f}^{*}\,;\,\boldsymbol{K}_{*\boldsymbol{u}}\boldsymbol{L}^{-\top}\boldsymbol{L}_{B}^{-\top}\boldsymbol{c},\,\boldsymbol{K}_{**}-\boldsymbol{K}_{*\boldsymbol{u}}\boldsymbol{L}^{-\top}[\boldsymbol{I}-\boldsymbol{B}^{-1}]\boldsymbol{L}^{-1}\boldsymbol{K}_{\boldsymbol{u}*}) \qquad \text{(C.2)}
\end{aligned}$$

C.3 Augmented Bound

We find the augmented bound in two stages. The first stage will help with derivations for Appendix C.4, and act as an intermediate stage to our augmented bound. We start by looking at the log determinant term of the collapsed variational lower bound and take the Cholesky decomposition $\boldsymbol{K}_{\boldsymbol{uu}} = \boldsymbol{L}\boldsymbol{L}^{\top}$

$$\begin{aligned}
\log|\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\Sigma| &= \log|\boldsymbol{L}^{-\top}\boldsymbol{L}^{-1}\Sigma| \\
&= \log|\boldsymbol{L}^{-1}(\boldsymbol{K}_{\boldsymbol{uu}}+\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}})\boldsymbol{L}^{-\top}| \\
&= \log|\boldsymbol{I}+\boldsymbol{L}^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{L}^{-\top}| \\
&= \log|\boldsymbol{I}|+\log|\boldsymbol{\Theta}|+\log|\boldsymbol{\Theta}^{-1}+\boldsymbol{Q}_{\boldsymbol{ff}}| \\
&= \log|\boldsymbol{\Theta}|+\log|\boldsymbol{\Theta}^{-1}+\boldsymbol{Q}_{\boldsymbol{ff}}|
\end{aligned}$$

where the penultimate equality is attained by the Matrix Determinant Lemma. We now look at the quadratic term

$$\begin{aligned}
\boldsymbol{y}^{\top}\boldsymbol{K}_{\boldsymbol{fu}}\Sigma^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{y} &= \boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\left[\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}(\boldsymbol{K}_{\boldsymbol{uu}}+\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}})^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\right]\boldsymbol{\Theta}^{-1}\boldsymbol{y}-\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{y}+\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{y} \\
&= \boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{y}-\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\left[\boldsymbol{\Theta}-\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}}(\boldsymbol{K}_{\boldsymbol{uu}}+\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\boldsymbol{K}_{\boldsymbol{fu}})^{-1}\boldsymbol{K}_{\boldsymbol{uf}}\boldsymbol{\Theta}\right]\boldsymbol{\Theta}^{-1}\boldsymbol{y} \\
&= \boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{y}-\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1}+\boldsymbol{K}_{\boldsymbol{fu}}\boldsymbol{K}_{\boldsymbol{uu}}^{-1}\boldsymbol{K}_{\boldsymbol{uf}})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} \\
&= \boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{y}-\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1}+\boldsymbol{Q}_{\boldsymbol{ff}})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y}
\end{aligned}$$

where the penultimate equality is attained by the Woodbury Matrix Identity. This forms the first stage of our augmented variational lower bound

$$\begin{aligned}
\mathcal{L}^{*} = &-\frac{1}{2}\log|\boldsymbol{\Theta}|-\frac{1}{2}\log|\boldsymbol{\Theta}^{-1}+\boldsymbol{Q}_{\boldsymbol{ff}}|+\frac{1}{8}\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}\boldsymbol{y}-\frac{1}{8}\boldsymbol{y}^{\top}\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1}+\boldsymbol{Q}_{\boldsymbol{ff}})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y}- \\
&\frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{\boldsymbol{ff}})-N\log 2+\frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n-2\log\cosh(\frac{c_n}{2})]
\end{aligned}$$

Now in order for our bound to satisfy the form of a log-normal distribution, we wish for the log determinant to contain the same covariance matrix as the inverted matrix in the quadratic term. We can therefore augmented the log determinants as such

$$\begin{aligned}
-\frac{1}{2}\log|\boldsymbol{\Theta}| - \frac{1}{2}\log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}| &= -\frac{1}{2}\log|\boldsymbol{\Theta}| - \frac{1}{2}\log(|\boldsymbol{\Theta}||\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}|) + \frac{1}{2}\log|\boldsymbol{\Theta}| \\
&= -\frac{1}{2}\log(|\boldsymbol{\Theta}||\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}|) \\
&= -\frac{1}{2}\log(|\boldsymbol{\Theta}||\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}||4\boldsymbol{\Theta}|) + \frac{1}{2}\log|4\boldsymbol{\Theta}| \\
&= -\frac{1}{2}\log|4\boldsymbol{\Theta}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})\boldsymbol{\Theta}| + \frac{1}{2}\log 4^N + \frac{1}{2}\log|\boldsymbol{\Theta}| \\
&= -\frac{1}{2}\log|4\boldsymbol{\Theta}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})\boldsymbol{\Theta}| + N\log 2 + \frac{1}{2}\log|\boldsymbol{\Theta}|
\end{aligned}$$

Plugging this back into our bound we get

$$\begin{aligned}
\mathcal{L}^* = &-\frac{1}{2}\log|4\boldsymbol{\Theta}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})\boldsymbol{\Theta}| - \frac{1}{8}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} - \frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}) + \\
&\frac{1}{2}\log|\boldsymbol{\Theta}| + \frac{1}{8}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})]
\end{aligned}$$

which we can finally augment to form the [Titsias (8)] style bound

$$\begin{aligned}
\mathcal{L}^* \;= &\log\mathcal{N}(\boldsymbol{y};\, \boldsymbol{0},\, 4\boldsymbol{\Theta}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})\boldsymbol{\Theta}) - \frac{1}{2}Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}) + \\
&\frac{1}{2}\log|\boldsymbol{\Theta}| + \frac{1}{8}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{N}{2}\log 2\pi + \frac{1}{2}\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})]
\end{aligned}$$

C.4 Monotonic Improvement of the ELBO

Similarly to Appendix B.4, we know that the addition of a new inducing point corresponds to a rank-1 update of the approximate covariance matrix $\boldsymbol{Q}_{ff}$. Namely, we have $\boldsymbol{Q}_{ff}^+ = \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^\top$, with $\boldsymbol{Q}_{ff}^+$ the approximate covariance matrix after adding a new inducing point, and $\boldsymbol{b}$ the corresponding rank-1 update vector.

Starting from the first stage of the augmented bound derived in Appendix C.3, we note that the matrix $\boldsymbol{Q}_{ff}$ is the only quantity which depends on the inducing points. We can therefore inspect the difference between the ELBO before and after adding an arbitrary inducing point

$$\begin{aligned}
2(\mathcal{L}^+ - \mathcal{L}) = &-\log|\boldsymbol{\Theta}| + \log|\boldsymbol{\Theta}| - \log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}^+| + \log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}| + \\
&\frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\boldsymbol{y} - \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\boldsymbol{y} - \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}^+)^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} - \\
&Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}^+) + Tr(\boldsymbol{\Theta}\tilde{\boldsymbol{K}}_{ff}) - N\log 2 + N\log 2 + \\
&\Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})] - \Sigma_{n=1}^{N}[c_n^2\theta_n - 2\log\cosh(\frac{c_n}{2})] \\
&= -\log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^\top| + \log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}| + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) - \\
&\frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^\top)^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y}
\end{aligned}$$

We can use the Matrix Determinant Lemma to simplify the determinant terms

$$\begin{aligned}2(\mathcal{L}^+ - \mathcal{L}) &= -\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) - \log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}| + \log|\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff}| + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) - \\ &\quad \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^\top)^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} \\ &= -\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) - \\ &\quad \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff} + \boldsymbol{b}\boldsymbol{b}^\top)^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y}\end{aligned}$$

We can use the Sherman-Morrison formula to simplify the inverse terms

$$\begin{aligned}2(\mathcal{L}^+ - \mathcal{L}) &= -\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) - \\ &\quad \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{\Theta}^{-1}\boldsymbol{y} + \\ &\quad \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\frac{(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}\boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}}{1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}\boldsymbol{\Theta}^{-1}\boldsymbol{y} \\ &= -\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) + \\ &\quad \frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\frac{(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}\boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}}{1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}\boldsymbol{\Theta}^{-1}\boldsymbol{y}\end{aligned}$$

which gives us the final form of the ELBO difference that is conveniently very similar to Appendix B.4. In order to show that this difference provides a monotonic increase, we again need to show that it is non-negative. We restate some important properties about the first two terms in the context of our classification model

$$\begin{aligned}Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) &= \boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b} \\ \log(1 + x) &\leq x \\ \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b} &\leq \boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b}\end{aligned}$$

From which we can deduce

$$\begin{aligned}&\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) \leq \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b} \leq \boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b} \\ \implies &-\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) \geq -\boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b}\end{aligned}$$

and so we can show that the sum of the first two terms is non-negative

$$\begin{aligned}-\log(1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}) + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) &\geq -\boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b} + Tr(\boldsymbol{\Theta}\boldsymbol{b}\boldsymbol{b}^\top) \\ &= -\boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b} + \boldsymbol{b}^\top\boldsymbol{\Theta}\boldsymbol{b} = 0\end{aligned}$$

Finally it remains to show that the quadratic term is non-negative. We have

$$\frac{1}{4}\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}\frac{(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}\boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}}{1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}\boldsymbol{\Theta}^{-1}\boldsymbol{y} = \frac{1}{4}\frac{(\boldsymbol{y}^\top\boldsymbol{\Theta}^{-1}(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b})^2}{1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b}}$$

which is trivially non-negative as the squared numerator is non-negative, and the denominator $1 + \boldsymbol{b}^\top(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}\boldsymbol{b} \geq 1$ as $(\boldsymbol{\Theta}^{-1} + \boldsymbol{Q}_{ff})^{-1}$ is SPSD.

Again, we note that if the new inducing point is a duplicate, the matrix $\boldsymbol{K}_{\boldsymbol{uu}}$ becomes singular and alternative reasoning is required. As our inducing points selection procedures prevent this duplication of inducing points from occurring, we omit this possibility. This concludes our proof of the monotonic improvement of the ELBO for our heteroscedastic classification model.

Appendix D

Evaluation

D.1 Inducing Point Optimality

In this section we provide graphical results for the convergence of SVGP with gradient-optimised inducing points initialised through the state-of-the-art k-means, and through HGV. We run each experiment using both the 'L-BFGS-B' and 'Adam' optimisers. We plot the variational lower bound on the vertical axis and the number of optimisation iterations on the horizontal axis. We denote the optimal non-sparse results by dashed black lines and do not repeat the datasets given in the body of the report.

We do not provide results for datasets with $N > 5000$, as gradient-based optimisation of the inducing points in these settings is computationally infeasible. We note that in the Crabs dataset, optimisation of the inducing points is extremely slow to converge, and even allowing $10,000$ iterations is insufficient for further visible convergence. We use $M = 80$ for Banana, $M = 150$ for Breast Cancer, $M = 20$ for Crabs, $M = 100$ for Heart Statlog, $M = 150$ for Ionosphere and $M = 100$ for Pima Diabetes.

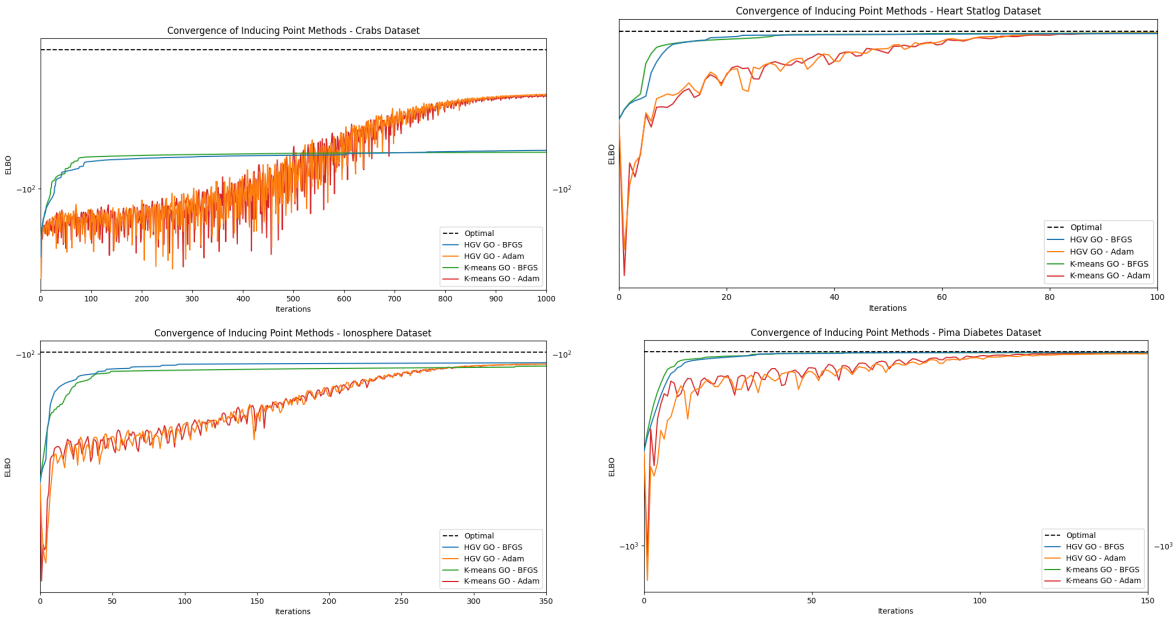

Figure D.1: Analysis of the change in variational lower bound with respect to the number of optimisation iterations for SVGP with gradient-optimised inducing points initialised with HGV and k-means. We repeat the experiment using both the 'L-BFGS-B' and 'Adam' optimisers. The optimal non-sparse results are denoted by black dashed lines.

D.2 Sparsity

In this section we provide the graphical results for the sparsity experiments which compare the performance of PGPR with a number of inducing point selection methods. We run each experiment multiple times, plotting the mean value of the variational lower bound on the vertical axis and the number of inducing points on the horizontal axis. We denote the optimal non-sparse results by dashed black lines and do not repeat the datasets given in the body of the report. The experimental results can be found on the subsequent pages.

D.2.1 Fixed Inducing Points

Here we give the sparsity results for the fixed inducing point schemes. For the MAGIC Telescope and Electricity datasets, a full sparsity experiment is not computationally feasible, so we sample 25% and 10% of the datasets respectively.

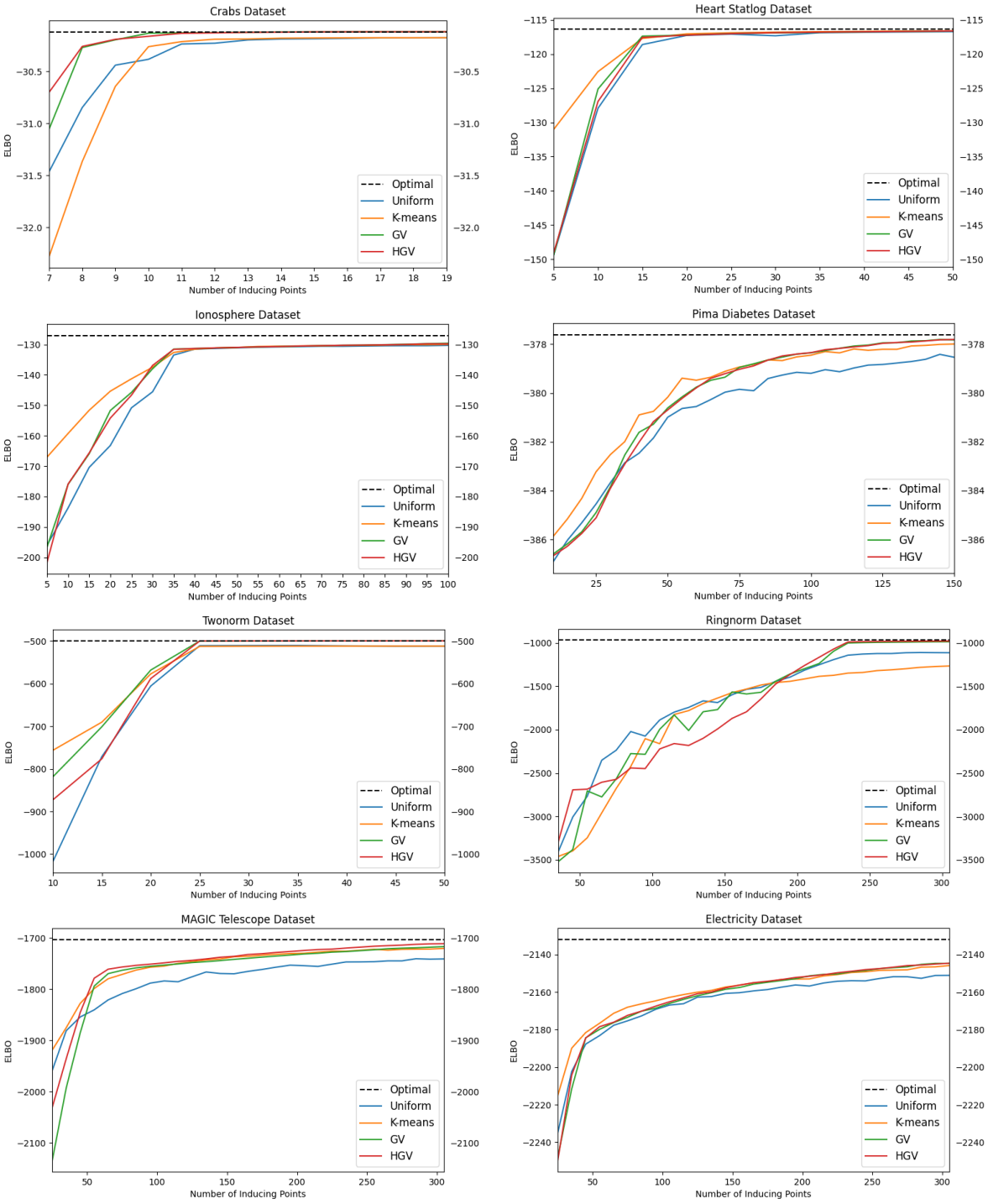

Figure D.2: Analysis of the change in variational lower bound with respect to the number of inducing points for PGPR paired with multiple inducing point selection methods. The optimal non-sparse results are denoted by black dashed lines.

D.2.2 Gradient-Based Optimisation

Here we give the sparsity results for PGPR HGV with and without gradient-based optimisation of the inducing points, and for PGPR K-means with gradient-based optimisation of the inducing points. We do not provide results for datasets with $N > 5000$, as a full sparsity experiment is computationally infeasible with gradient-based optimisation. In practice however, we observe the continuing trend that after a sufficiently large M is used, gradient-based optimisation of the inducing points does not provide significant benefits compared to its computational expense. We also observe that gradient-optimised k-means converges suboptimally for some datasets.

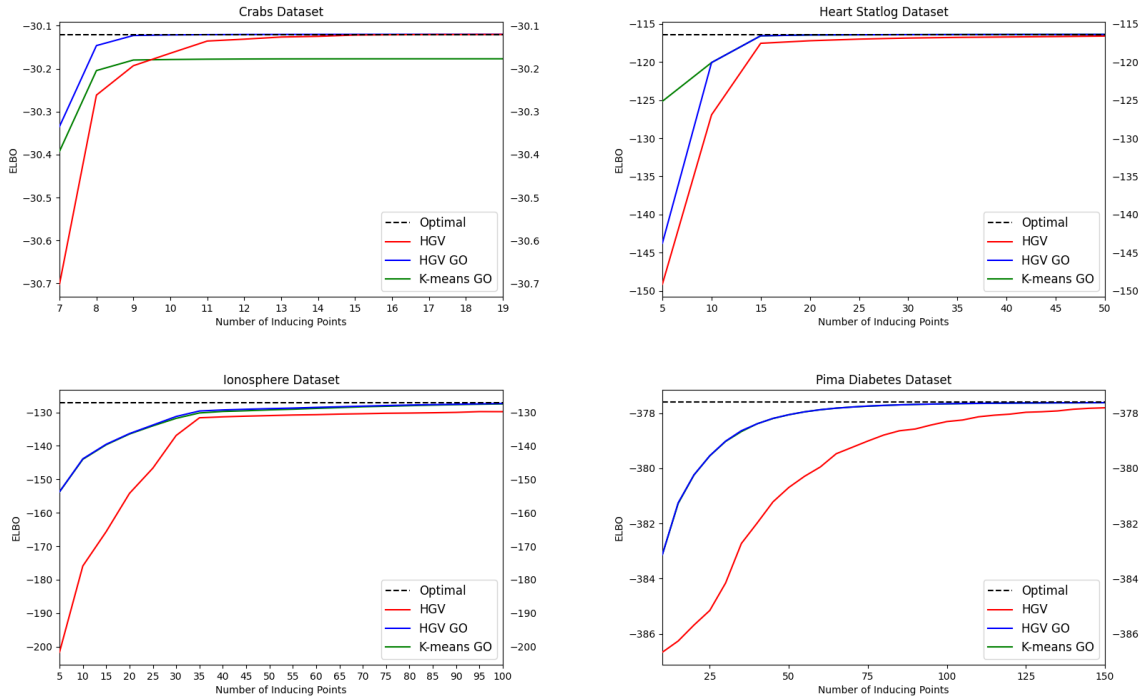

Figure D.3: Analysis of the change in variational lower bound with respect to the number of inducing points for PGPR HGV with and without gradient-based optimisation of the inducing points, and for PGPR K-means with gradient-based optimisation of the inducing points. The optimal non-sparse results are denoted by black dashed lines.

D.3 Quantitative Results

In the following sections we give the quantitative results for a variety of models on ten datasets across three metrics: the ELBO, the predictive accuracy and the negative test log-likelihood. We compute a distribution over each metric so that there are a total number of fifteen criteria for each model on each dataset.

D.3.1 Fixed Inducing Points

We first evaluate fixed inducing point selection methods, where by 'fixed' we mean to say that the inducing points are not optimised through gradient-based optimisation. We compare the state-of-the-art SVGP K-means to our model PGPR paired with four different initialisation procedures.

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Crabs N=200 M=10 D=6	SVGP K-means	ELBO	**-22.58**	-54.52	**-29.84**	-32.03	8.38
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	0.1041	**0.0017**	0.0213	0.0290	0.0289
	PGPR Uniform	ELBO	-30.10	-30.63	-30.34	-30.32	0.16
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	0.0289	0.0037	0.0089	0.0125	0.0089
	PGPR K-means	ELBO	-30.01	-30.56	-30.28	-30.29	0.17
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	0.0284	0.0033	0.0091	0.0125	0.0090
	PGPR GV	ELBO	-29.63	-30.14	-29.92	-29.92	**0.14**
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	**0.0165**	0.0026	0.0079	0.0081	**0.0044**
	PGPR HGV	ELBO	-29.46	**-30.11**	-29.96	**-29.89**	0.20
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	0.0280	**0.0025**	**0.0045**	**0.0079**	0.0089

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Heart N=270 M=35 D=13	SVGP K-means	ELBO	-102.46	-147.65	-105.17	-116.76	19.14
		Acc.	0.8889	0.7407	0.8148	0.8259	0.0372
		NLL	0.5416	0.2341	0.3946	0.3947	0.0958
	PGPR Uniform	ELBO	-102.17	-111.34	-105.27	-106.15	2.98
		Acc.	**0.9259**	0.7778	0.7963	0.8259	0.0575
		NLL	0.5465	**0.1907**	0.4353	0.3953	0.1141
	PGPR K-means	ELBO	-102.08	-111.20	**-105.03**	**-105.98**	2.97
		Acc.	**0.9259**	0.7778	0.7963	0.8259	0.0575
		NLL	0.5466	0.1909	0.4346	0.3949	0.1140
	PGPR GV	ELBO	**-100.66**	**-108.26**	-106.78	-106.21	2.12
		Acc.	**0.9259**	0.7037	**0.8704**	**0.8556**	0.0584
		NLL	0.6215	0.3112	0.3719	0.3898	0.0878
	PGPR HGV	ELBO	-105.71	-109.83	-107.19	-107.37	**1.17**
		Acc.	0.8889	**0.8148**	0.8518	0.8444	**0.0277**
		NLL	**0.4247**	0.2407	**0.3479**	**0.3472**	**0.0509**

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Ionosphere N=351 M=50 D=33	**SVGP K-means**	ELBO	**-103.56**	**-112.05**	**-107.21**	**-107.00**	2.52
		Acc.	**0.9444**	0.8333	**0.9028**	0.8917	0.0339
		NLL	0.4449	0.2047	**0.2907**	**0.2990**	0.0817
	PGPR Uniform	ELBO	-116.08	-122.25	-119.97	-119.35	2.23
		Acc.	0.8889	0.8056	0.8611	0.8667	**0.0242**
		NLL	0.5304	0.2440	0.3356	0.3590	0.0929
	PGPR K-means	ELBO	-115.89	-122.03	-119.63	-119.11	**2.19**
		Acc.	0.8889	0.8056	0.8611	0.8667	**0.0242**
		NLL	0.5304	0.2441	0.3314	0.3569	0.0934
	PGPR GV	ELBO	-114.37	-124.59	-119.59	-119.10	3.26
		Acc.	0.9167	0.8333	0.8750	0.8750	0.0334
		NLL	0.5857	**0.1519**	0.3261	0.3702	0.1450
	PGPR HGV	ELBO	-116.20	-124.00	-120.70	-120.52	2.24
		Acc.	**0.9444**	**0.8611**	**0.9028**	**0.8944**	0.0299
		NLL	**0.4434**	0.2269	0.2999	0.3122	**0.0765**

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Banana N=400 M=40 D=2	**SVGP K-means**	ELBO	**-89.52**	**-103.25**	**-98.95**	**-98.63**	3.79
		Acc.	0.9750	**0.8250**	0.9125	0.9025	**0.0394**
		NLL	0.5381	0.0618	**0.2026**	0.2226	0.1239
	PGPR Uniform	ELBO	-107.42	-121.19	-113.86	-114.14	3.90
		Acc.	**1.0000**	**0.8250**	0.9125	0.9100	0.0561
		NLL	0.4400	0.0541	0.2043	0.2108	0.1064
	PGPR K-means	ELBO	-104.26	-117.65	-111.92	-111.79	3.54
		Acc.	**1.0000**	**0.8250**	0.9125	0.9100	0.0561
		NLL	**0.4333**	**0.0529**	0.2027	**0.2085**	0.1058
	PGPR GV	ELBO	-105.11	-115.94	-110.19	-110.54	3.69
		Acc.	**1.0000**	**0.8250**	0.9000	0.9075	0.0560
		NLL	0.4538	0.0691	0.2298	0.2387	0.1040
	PGPR HGV	ELBO	-103.15	-115.83	-110.75	-110.91	**3.23**
		Acc.	**1.0000**	**0.8250**	**0.9250**	**0.9125**	0.0448
		NLL	0.4570	0.0694	0.2158	0.2167	**0.0978**

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Breast N=569 M=50 D=30	SVGP K-means	ELBO	**-51.03**	-260.19	-79.26	-128.16	87.82
		Acc.	**1.0000**	0.8772	0.9561	0.9544	0.0344
		NLL	0.4388	0.0357	0.1476	0.1988	0.1450
	PGPR Uniform	ELBO	-63.71	-73.05	-69.59	-68.64	3.43
		Acc.	**1.0000**	0.9298	0.9649	0.9702	0.0249
		NLL	0.2447	0.0322	0.0911	0.1205	0.0793
	PGPR K-means	ELBO	-63.48	-72.85	-69.53	**-68.51**	3.43
		Acc.	**1.0000**	0.9298	0.9649	0.9702	0.0249
		NLL	0.2438	0.0322	0.0911	0.1203	0.0792
	PGPR GV	ELBO	-65.15	-72.81	-71.47	-70.44	2.44
		Acc.	**1.0000**	**0.9649**	**0.9825**	**0.9842**	**0.0146**
		NLL	0.1741	0.0241	**0.0479**	**0.0690**	0.0480
	PGPR HGV	ELBO	-65.06	**-71.67**	**-69.38**	-69.55	**1.81**
		Acc.	**1.0000**	0.9474	**0.9825**	0.9807	**0.0146**
		NLL	**0.1569**	**0.0227**	0.0857	0.0822	**0.0336**

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Pima N=768 M=60 D=8	SVGP K-means	ELBO	-335.88	**-346.81**	**-342.07**	**-341.64**	3.38
		Acc.	0.8441	0.7142	**0.7857**	0.7805	0.0351
		NLL	0.5224	**0.3641**	0.4434	0.4410	0.0476
	PGPR Uniform	ELBO	-339.66	-351.34	-343.86	-344.07	2.99
		Acc.	0.8571	0.7143	0.7597	0.7701	0.0427
		NLL	0.5402	0.3715	0.4733	0.4710	0.0431
	PGPR K-means	ELBO	-339.14	-350.28	-343.08	-343.22	**2.87**
		Acc.	0.8442	0.7143	0.7532	0.7675	0.0425
		NLL	0.5376	0.3699	0.4712	0.4696	0.0429
	PGPR GV	ELBO	**-333.93**	-349.47	-343.71	-343.78	4.83
		Acc.	**0.8701**	0.7143	0.7662	0.7883	0.0563
		NLL	0.6133	0.3872	0.4610	0.4665	0.0694
	PGPR HGV	ELBO	-341.61	-350.63	-345.75	-346.00	2.92
		Acc.	0.8571	**0.7662**	**0.7857**	**0.7974**	**0.0303**
		NLL	**0.4897**	0.3710	**0.4404**	**0.4345**	**0.0388**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Twonorm N=7,400 M=300 D=20	SVGP K-means	ELBO	-440.25	-4480.60	-4454.93	-3581.03	1314.97
		Acc.	0.9824	0.8108	0.9682	0.9436	0.0580
		NLL	0.6678	0.0543	0.6630	0.5167	0.2045
	PGPR Uniform	ELBO	**-431.67**	-467.89	-455.82	-455.12	9.19
		Acc.	**0.9865**	0.9675	0.9777	0.9782	0.0050
		NLL	0.0919	0.0442	0.0603	0.0610	0.0122
	PGPR K-means	ELBO	-431.69	-467.90	455.84	-455.14	9.19
		Acc.	**0.9865**	0.9675	0.9777	0.9782	0.0050
		NLL	0.0919	0.0442	0.0603	0.0609	0.0122
	PGPR GV	ELBO	-444.94	-465.35	-458.61	-457.37	**6.19**
		Acc.	0.9834	**0.9730**	0.9784	0.9777	**0.0039**
		NLL	**0.0738**	0.0458	0.0541	0.0561	**0.0087**
	PGPR HGV	ELBO	-441.42	**-464.02**	**-452.08**	**-451.61**	7.61
		Acc.	**0.9865**	**0.9730**	**0.9797**	**0.9793**	0.0048
		NLL	0.0759	**0.0433**	**0.0520**	**0.0544**	0.0107
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Ringnorm N=7,400 M=300 D=20	SVGP K-means	ELBO	**-768.29**	-4332.23	-2597.25	-2260.88	1301.82
		Acc.	0.9797	0.4959	0.8716	0.8032	0.1836
		NLL	0.6470	0.0431	0.3159	0.2907	0.2094
	PGPR Uniform	ELBO	-932.47	-980.56	-960.69	-959.43	15.23
		Acc.	0.9878	0.9675	0.9756	**0.9772**	0.0059
		NLL	**0.0823**	0.0484	0.0705	0.0667	0.0120
	PGPR K-means	ELBO	-933.98	-971.33	-954.93	-956.43	10.90
		Acc.	0.9878	0.9689	0.9756	**0.9772**	0.0059
		NLL	0.0828	0.0489	0.0707	0.0670	**0.0120**
	PGPR GV	ELBO	-926.20	-987.94	-965.57	-958.98	19.98
		Acc.	0.9838	**0.9703**	0.9736	0.9757	**0.0046**
		NLL	0.0880	0.0520	0.0663	0.0677	**0.0120**
	PGPR HGV	ELBO	-933.85	**-967.67**	**-952.89**	**-953.28**	**9.54**
		Acc.	**0.9892**	0.9648	**0.9804**	0.9766	0.0070
		NLL	0.0974	**0.0377**	**0.0548**	**0.0620**	0.0160

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Magic N=19,020 M=300 D=10	**SVGP K-means**	ELBO	**-5699.38**	**-5777.61**	**-5737.34**	**-5736.18**	24.04
		Acc.	0.8791	0.8565	0.8712	0.8699	0.0064
		NLL	**0.3355**	**0.2962**	**0.3168**	**0.3190**	**0.0105**
	PGPR Uniform	ELBO	-5907.59	-5980.16	-5935.34	-5936.95	24.63
		Acc.	0.8770	0.8601	0.8675	0.8677	0.0055
		NLL	0.3396	0.3002	0.3308	0.3260	0.0138
	PGPR K-means	ELBO	-5835.69	-5895.96	-5845.25	-5854.74	21.18
		Acc.	0.8770	0.8570	0.8693	0.8690	0.0060
		NLL	0.3356	0.2993	0.3269	0.3227	0.0130
	PGPR GV	ELBO	-5839.50	-5894.21	-5859.70	-5864.28	**17.87**
		Acc.	0.8775	**0.8649**	0.8691	0.8701	**0.0039**
		NLL	0.3463	0.3096	0.3317	0.3282	0.0117
	PGPR HGV	ELBO	-5801.45	-5868.47	-5828.07	-5831.01	19.70
		Acc.	**0.8822**	0.8623	**0.8738**	**0.8723**	0.0056
		NLL	0.3535	0.3008	0.3241	0.3244	0.0135
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Electricity N=45,312 M=300 D=8	**SVGP K-means**	ELBO	**-17992**	**-18166**	**-18105**	**-18099**	44.83
		Acc.	0.8133	0.7919	0.8036	**0.8037**	0.0060
		NLL	**0.4449**	**0.4200**	0.4249	**0.4271**	0.0075
	PGPR Uniform	ELBO	-18285	-18402	-18337	-18342	38.98
		Acc.	0.8056	**0.7966**	0.7994	0.8000	**0.0030**
		NLL	**0.4449**	0.4248	0.4356	0.4349	0.0051
	PGPR K-means	ELBO	-18149	-18284	-18209	-18208	36.17
		Acc.	0.8096	0.7955	0.8010	0.8013	0.0041
		NLL	0.4421	0.4235	0.4330	0.4319	**0.0050**
	PGPR GV	ELBO	-18282	-18422	-18389	-18373	44.60
		Acc.	0.8032	0.7855	0.7934	0.7942	0.0049
		NLL	0.4558	0.4345	0.4414	0.4421	0.0069
	PGPR HGV	ELBO	-18290	-18390	-18348	-18342	**33.79**
		Acc.	**0.8155**	0.7902	**0.8042**	0.8020	0.0071
		NLL	0.4514	0.4217	**0.4235**	0.4322	0.0081

Table D.1: Results for the evidence lower bound, accuracy and negative test log-likelihood on ten datasets. We assess the SVGP benchmark against four fixed initialisation methods paired with our model, PGPR. The best results in each criterion are emboldened, and the best model is chosen to be the one which wins the most criteria.

D.3.2 Gradient-Based Optimisation

We now evaluate models in which the inducing points are optimised through gradient-based optimisation. The first model that we evaluate is the state-of-the-art SVGP model where the inducing points are initialised with k-means and then optimised. The second model is the same, but using our model PGPR. We also duplicate the results of PGPR HGV from the previous section for comparison, as it was found to be the best performing fixed inducing point model.

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Crabs N=200 M=10 D=6	SVGP GO	ELBO	-35.76	-47.52	-41.12	-41.13	2.93
		Acc.	**1.0000**	0.9500	**1.0000**	0.9950	0.0150
		NLL	0.0990	0.0329	0.0388	0.0569	0.0248
	PGPR GO	ELBO	-33.37	-43.58	-37.19	-37.48	2.91
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	0.0737	0.0075	0.0366	0.0378	0.0193
	PGPR HGV	ELBO	**-29.46**	**-30.11**	**-29.96**	**-29.89**	**0.20**
		Acc.	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**0.0000**
		NLL	**0.0280**	**0.0025**	**0.0045**	**0.0079**	**0.0089**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Heart N=270 M=35 D=13	SVGP GO	ELBO	**-98.53**	-146.19	**-103.06**	-113.98	18.51
		Acc.	**0.9259**	0.7407	0.8148	0.8111	0.0560
		NLL	0.5517	**0.2247**	0.4486	0.4348	0.0969
	PGPR GO	ELBO	-102.26	**-109.82**	-105.61	**-105.63**	1.97
		Acc.	**0.9259**	0.7407	0.8333	0.8222	0.0593
		NLL	0.5256	0.2353	0.4012	0.3953	0.0755
	PGPR HGV	ELBO	-105.71	-109.83	-107.19	-107.37	**1.17**
		Acc.	0.8889	**0.8148**	**0.8518**	**0.8444**	**0.0277**
		NLL	**0.4247**	0.2407	**0.3479**	**0.3472**	**0.0509**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Ionosphere N=351 M=50 D=33	**SVGP GO**	ELBO	**-93.73**	**-99.60**	**-97.14**	**-96.84**	**1.63**
		Acc.	**0.9722**	**0.8889**	**0.9306**	**0.9361**	**0.0279**
		NLL	**0.3539**	**0.0727**	**0.1871**	**0.1916**	0.0856
	PGPR GO	ELBO	-115.06	-122.05	-119.39	-119.26	1.72
		Acc.	**0.9722**	0.8611	0.9167	0.9139	0.0339
		NLL	0.4404	0.1545	0.2287	0.2518	0.0809
	PGPR HGV	ELBO	-116.20	-124.00	-120.70	-120.52	2.24
		Acc.	0.9444	0.8611	0.9028	0.8944	0.0299
		NLL	0.4434	0.2269	0.2999	0.3122	**0.0765**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Banana N=400 M=40 D=2	**SVGP GO**	ELBO	**-91.61**	**-102.75**	**-98.14**	**-98.03**	3.91
		Acc.	0.9750	**0.8250**	**0.9500**	**0.9175**	0.0571
		NLL	0.4019	**0.0655**	0.2200	0.2174	0.1230
	PGPR GO	ELBO	-105.63	-115.63	-111.12	-111.17	3.39
		Acc.	**1.0000**	**0.8250**	0.9250	0.9100	0.0594
		NLL	**0.3772**	0.0693	0.2186	**0.2138**	0.1043
	PGPR HGV	ELBO	-103.15	-115.83	-110.75	-110.91	**3.23**
		Acc.	**1.0000**	**0.8250**	0.9250	0.9125	**0.0448**
		NLL	0.4570	0.0694	**0.2158**	0.2167	**0.0978**

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Breast N=569 M=50 D=30	SVGP GO	ELBO	**-49.45**	**-60.15**	**-55.71**	**-55.18**	2.72
		Acc.	**1.0000**	0.9474	**0.9825**	0.9772	0.0158
		NLL	0.2269	**0.0145**	0.0749	0.0938	0.0613
	PGPR GO	ELBO	-70.21	-73.54	-71.78	-71.69	**1.06**
		Acc.	**1.0000**	**0.9825**	**0.9825**	**0.9895**	**0.0086**
		NLL	**0.0896**	0.0310	**0.0659**	**0.0605**	**0.0166**
	PGPR HGV	ELBO	-65.06	-71.67	-69.38	-69.55	1.81
		Acc.	**1.0000**	0.9474	**0.9825**	0.9807	0.0146
		NLL	0.1569	0.0227	0.0857	0.0822	0.0336
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Pima N=768 M=60 D=8	SVGP GO	ELBO	**-333.57**	**-346.31**	**-338.80**	**-339.44**	3.40
		Acc.	0.8442	0.7013	0.7792	0.7844	0.0386
		NLL	0.5138	**0.3569**	0.4480	0.4397	0.0414
	PGPR GO	ELBO	-338.36	-350.56	-342.95	-343.67	3.29
		Acc.	0.8442	0.7013	**0.7857**	0.7857	0.0378
		NLL	0.5077	0.3572	0.4506	0.4406	0.0402
	PGPR HGV	ELBO	-341.61	-350.63	-345.75	-346.00	**2.92**
		Acc.	**0.8571**	**0.7662**	**0.7857**	**0.7974**	**0.0303**
		NLL	**0.4897**	0.3710	**0.4404**	**0.4345**	**0.0388**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Twonorm N=7,400 M=300 D=20	SVGP GO	ELBO	-4064.00	-4501.34	-4471.07	-4402.97	151.04
		Acc.	0.9743	0.8608	0.9662	0.9487	0.0349
		NLL	0.6714	0.4862	0.6667	0.6348	0.0652
	PGPR GO	ELBO	**-436.05**	-471.81	-459.24	-456.35	12.27
		Acc.	**0.9892**	0.9689	0.9777	0.9778	0.0058
		NLL	0.0878	**0.0402**	0.0572	0.0609	0.0160
	PGPR HGV	ELBO	-441.42	**-464.02**	**-452.08**	**-451.61**	**7.61**
		Acc.	0.9865	**0.9730**	**0.9797**	**0.9793**	**0.0048**
		NLL	**0.0759**	0.0433	**0.0520**	**0.0544**	**0.0107**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Ringnorm N=7,400 M=300 D=20	SVGP GO	ELBO	**-717.26**	-3711.06	**-739.04**	-1321.61	1179.37
		Acc.	0.9851	0.5541	0.9777	0.8988	0.1622
		NLL	0.5188	0.0458	0.0599	0.1487	0.1840
	PGPR GO	ELBO	-1045.65	-1133.13	-1119.40	-1101.00	31.81
		Acc.	0.9878	**0.9703**	0.9770	**0.9786**	**0.0054**
		NLL	**0.0807**	0.0604	0.0720	0.0699	**0.0068**
	PGPR HGV	ELBO	-933.85	**-967.67**	-952.89	**-953.28**	**9.54**
		Acc.	**0.9892**	0.9648	**0.9804**	0.9766	0.0070
		NLL	0.0974	**0.0377**	**0.0548**	**0.0620**	0.0160

Dataset	Method	Metric	Max	Min	Median	Mean	Std
Magic N=19,020 M=300 D=10	**SVGP GO**	ELBO	**-5579.40**	**-5667.30**	**-5649.15**	**-5639.90**	28.98
		Acc.	0.8812	0.8591	0.8733	**0.8727**	0.0066
		NLL	**0.3434**	**0.2864**	**0.3102**	**0.3103**	0.0144
	PGPR GO	ELBO	-5716.90	-5825.83	-5776.89	-5780.08	28.37
		Acc.	0.8801	0.8565	0.8722	0.8711	0.0070
		NLL	0.3479	0.2903	0.3146	0.3147	0.0140
	PGPR HGV	ELBO	-5801.45	-5868.47	-5828.07	-5831.01	**19.70**
		Acc.	**0.8822**	**0.8623**	**0.8738**	0.8723	**0.0056**
		NLL	0.3535	0.3008	0.3241	0.3244	**0.0135**
Dataset	Method	Metric	Max	Min	Median	Mean	Std
Electricity N=45,312 M=300 D=8	SVGP GO	ELBO	**-17440**	**-17586**	**-17524**	**-17515**	47.35
		Acc.	0.8177	**0.8036**	**0.8135**	0.8117	0.0046
		NLL	0.4222	**0.3992**	0.4063	0.4082	0.0074
	PGPR GO	ELBO	-17621	-17736	-176780	-17685	38.84
		Acc.	**0.8193**	0.8012	**0.8135**	**0.8118**	**0.0044**
		NLL	**0.4216**	0.3998	**0.4050**	**0.4073**	**0.0065**
	PGPR HGV	ELBO	-18290	-18390	-18348	-18342	**33.79**
		Acc.	0.8155	0.7902	0.8042	0.8020	0.0071
		NLL	0.4514	0.4217	0.4235	0.4322	0.0081

Table D.2: Results for the evidence lower bound, accuracy and negative test log-likelihood on ten datasets. We assess the SVGP gradient-optimised benchmark against two variants of PGPR with and without gradient-based optimisation of the inducing points. The best results in each criterion are emboldened, and the best model is chosen to be the one which wins the most criteria.