4th Symposium on Advances in Approximate Bayesian Inference, 2021 1–7

Improved Inverse-Free Variational Bounds for Sparse Gaussian Processes

Mark van der Wilk M.VDWILK@IMPERIAL.AC.UK
Artem Artemev A.ARTEMEV20@IMPERIAL.AC.UK
Department of Computing, Imperial College London, UK

James Hensman[*] HENSMAN@AMAZON.COM
Amazon, UK

Abstract

The need for matrix decompositions (inverses) is often named as a major impediment to scaling Gaussian process (GP) models, even in efficient approximations. To address this, Van der Wilk et al. (2020) introduced a variational lower bound that can be computed without these costly operations. We improve this bound by **1)** simplifying it by removing the need for iterative procedures, and **2)** making it more numerically stable. While these improvements do not result in a procedure that is faster in wall-clock time than existing variational bounds, they are likely to be necessary steps along the way.

1. Introduction

Gaussian processes (GPs) (Rasmussen and Williams, 2006) are distributions on functions with many properties that make them convenient for Bayesian modelling. In particular, their nonparametric properties improve uncertainty estimation and simplify hyperparameter tuning. Variational approximations (Titsias, 2009) were originally introduced to address the $O(N^3)$ computational cost of manipulating GPs when considering N datapoints. However, these approximations also enabled Gaussian processes to be used in much more sophisticated models. Deep Gaussian Processes (DGPs) in particular are intriguing as they present a way to merge the benefits of GPs with Neural Networks (NNs). While DGPs can be seen as direct Bayesian analogues of deep NNs (Dutordoir et al., 2021), DGPs are still not a convenient way to perform Bayesian Deep Learning. One contributing factor to this, is the high cost of each training iteration, despite the development of minibatching (Hensman et al., 2013). One cause is the need to compute an $M \times M$ matrix inverse and determinant, making the cost for each layer $O(M^3 + BM^2)$, where M is analogous to the number of neurons in a layer and B is the minibatch size. These matrix operations are costly and limit the effectiveness of minibatching for speeding up each training iteration. In addition, these matrix operations are serial and require high-precision floating point operations, which is poorly suited to modern day hardware. It therefore seems likely that these operations will need to be removed for Deep Gaussian Processes to truly scale.

We investigate variational approximations that do not require the computation of costly matrix operations at every iteration. Van der Wilk et al. (2020) introduced such an "inverse-free" approximation, which could be computed without performing matrix decompositions or inverses to completion, while also being a drop-in replacement for schemes based on

[*] Work completed prior to joining Amazon.

Hensman et al. (2013) (e.g. Salimbeni and Deisenroth (2017) for DGPs). We make the inverse-free bound **1)** less cumbersome, by removing iterative procedures, and **2)** more numerically stable. While issues with training still prevent a wall-clock speed-up, we believe our improvements are necessary steps on the way.

2. Variational Inference for Gaussian Process Models

For the sake of brevity, we consider the simplest possible model to introduce our new bound. We aim to learn some function $f : \mathcal{X} \to \mathbb{R}$ in a Bayesian manner by placing a GP prior on $f(\cdot)$, and observing data $\mathbf{y} \in \mathbb{R}^N$ through some arbitrary pointwise factorised likelihood:

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot')) , \qquad\qquad p(\mathbf{y}|f(X)) = \prod_{n=1}^{N} p(y_n|f(\mathbf{x}_n)) . \tag{1}$$

2.1. Marginal Parameterisation

Variational inference for GPs[1] decouples the size of matrix inverses from the dataset size (Titsias, 2009), allows minibatching (Hensman et al., 2013), and addresses non-conjugacy (Hensman et al., 2015). The approximation introduces a set of tractable approximate posteriors by conditioning the prior on M *inducing variables* $\mathbf{u} \in \mathbb{R}^M$, e.g. observations $f(Z)$ for the M inputs collected in $Z \in \mathcal{X}^M$, and specifying the marginal distribution on $\mathbf{u}$ as $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$. This results in the approximate posterior for arbitrary inputs X^*:

$$\begin{aligned} q(f(X^*)) &= \int p(f(X^*|\mathbf{u})q(\mathbf{u})\mathrm{d}\mathbf{u} \\ &= \mathcal{N}\big(f(X^*); \quad \mathbf{k}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m}, \quad \mathbf{K}_{**} - \mathbf{k}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}*} + \mathbf{k}_{*\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}*}\big) , \end{aligned} \tag{2}$$

where the subscripts of $\mathbf{K}$ matrices determine its elements as e.g. $[\mathbf{K}_{\mathbf{uu}}]_{ij} = \mathrm{Cov}[u_i, u_j] = k(z_i, z_j)$. The KL between approximation and posterior (Matthews et al., 2016) can be minimised by maximising the ELBO (Hensman et al., 2013, 2015):

$$\mathcal{L}_{\text{sv}} = \sum_{n=1}^{N} \mathbb{E}_{q(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] , \tag{3}$$

$$\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \frac{1}{2}\Big(\mathrm{Tr}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}) + \mathbf{m}^{\mathsf{T}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{m} - M + \log|\mathbf{K}_{\mathbf{uu}}| - \log|\mathbf{S}|\Big) . \tag{4}$$

2.2. Likelihood Parameterisation

While the former parameterisation is the most common, it is numerically unstable as $\mathbf{K}_{\mathbf{uu}}$ can be arbitrarily badly conditioned (e.g. singular for repeated inducing inputs). Panos et al. (2018) noted this in passing, and suggested using the *likelihood parameterisation*:

$$q(\mathbf{u}) = \frac{\mathcal{N}(\mathbf{u}; \tilde{\mathbf{y}}, \boldsymbol{\Sigma})p(\mathbf{u})}{\int \mathcal{N}(\mathbf{u}; \tilde{\mathbf{y}}, \boldsymbol{\Sigma})p(\mathbf{u})\mathrm{d}\mathbf{u}} = \mathcal{N}\Big(\mathbf{u}; \underbrace{\mathbf{K}_{\mathbf{uu}}(\mathbf{K}_{\mathbf{uu}}+\boldsymbol{\Sigma})^{-1}\tilde{\mathbf{y}}}_{=\,\mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu}\,=\,\mathbf{m}}, \underbrace{\mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}(\mathbf{K}_{\mathbf{uu}}+\boldsymbol{\Sigma})^{-1}\mathbf{K}_{\mathbf{uu}}}_{=\,(\mathbf{K}_{\mathbf{uu}}^{-1}+\boldsymbol{\Sigma}^{-1})^{-1}\,=\,\mathbf{S}}\Big) . \tag{5}$$

[1] See Van der Wilk (2019) for a gentle introduction, or Matthews (2017) for a rigorous discussion.

This can simply be seen as a straightforward reparameterisation of $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \mathbf{m}, \mathbf{S})$ (note the equalities to $\mathbf{m}, \mathbf{S}$ above)[2], but with the matrix operations now performed on $\mathbf{K_{uu}} + \mathbf{\Sigma}$:

$$q(f(\mathbf{x}_n)) = \mathcal{N}\big(f(\mathbf{x}_n); \quad \mathbf{k}_{n\mathbf{u}}\boldsymbol{\mu}, \quad k_{nn} - \mathbf{k}_{n\mathbf{u}}(\mathbf{K_{uu}} + \mathbf{\Sigma})^{-1}\mathbf{k}_{\mathbf{u}n}\big) \,, \tag{6}$$

$$\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] = \frac{1}{2}\big(-\operatorname{Tr}\big((\mathbf{K_{uu}} + \mathbf{\Sigma})^{-1}\mathbf{K_{uu}}\big) + \boldsymbol{\mu}^{\mathsf{T}}\mathbf{K_{uu}}\boldsymbol{\mu} + \log|\mathbf{K_{uu}} + \mathbf{\Sigma}| - \log|\mathbf{\Sigma}|\big) \,. \tag{7}$$

This has one big advantage: by enforcing a small minimum diagonal term in $\mathbf{\Sigma}$, we only need to decompose matrices with lower bounded minimum eigenvalues. In the marginal parameterisation, this is typically ensured in an ad-hoc manner by adding “jitter” to $\mathbf{K_{uu}}$ directly. In the likelihood parameterisation, numerical stability is ensured through a well-defined variational parameter. If it has to be constrained to be larger than what is optimal because of limited floating-point precision, this is penalised in the ELBO, in a manner that is unified with all other properties of the approximation. Typically though, the uncertainty on $\mathbf{u}$ will be many times larger than jitter matrices, leading to much more stable inverses.[3] This improved numerical stability is even more crucial for inverse-free variational bounds.

3. Improved Inverse-Free Variational Bounds for Gaussian Processes

3.1. Inverse-Free Marginal Parameterisation

Van der Wilk et al. (2020) noted that substituting an over-estimate of the predictive variance of eq. (2) into the ELBO calculation resulted in a further lower bound to $\mathcal{L}_{\text{sv}}$, for concave log-likelihoods. This allows further inverses to be removed by bounding $\mathbf{k}_{n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n}$ using

$$\big(\mathbf{a}_n - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n}\big)^{\mathsf{T}}\mathbf{K}_{\mathbf{uu}}^{-1}\big(\mathbf{a}_n - \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n}\big) \geq 0 \tag{8}$$

$$\implies k_{nn} - \mathbf{k}_{n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n} \leq k_{nn} + \mathbf{a}_n^{\mathsf{T}}\mathbf{K_{uu}}\mathbf{a}_n - 2\mathbf{a}_n^{\mathsf{T}}\mathbf{k}_{\mathbf{u}n} \,, \tag{9}$$

with equality when $\mathbf{a}_n = \mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n}$. In principle, this could lead to a method where one $\mathbf{a}_n$ per datapoint would be optimised alongside with the existing variational parameters. To avoid adding a parameter for each datapoint, we can reparameterise $\mathbf{a}_n = \mathbf{T}\mathbf{k}_{\mathbf{u}n}$, where $\mathbf{T} \in \mathbb{R}^{M\times M}$ is now a free matrix which takes $\mathbf{T} = \mathbf{K}_{\mathbf{uu}}^{-1}$ when optimised to a maximum. Van der Wilk et al. (2020) suggested to optimise $\mathbf{T}$ along with the other variational parameters.

Herein lies the problem of numerical stability of this parameterisation: As the conditioning of $\mathbf{K_{uu}}$ worsens, entries in $\mathbf{K}_{\mathbf{uu}}^{-1}$ tend to infinity. Gradient-based optimisers (e.g. Adam) have limited step sizes and cannot be expected to successfully recover such solutions.

The second problem of using this parameterisation lies in the log-determinants that need to be computed for the KL term (eq. (4)). These terms were not removed by the reparameterisation, and are equally costly as inverses. The same Cholesky decomposition that is often used for computing the inverse is used for computing log-determinant terms.

To still obtain an inverse-free method, this is addressed by using randomly truncated Conjugate Gradients (CG) to compute unbiased estimates of the gradient of the log-determinant (Filippone and Engler, 2015). To avoid needing to run M iterations of CG and

[2] Bui et al. (2017) discuss this to unify variational and EP methods, and note that it works by specifying variational parameters more as data that is combined with the prior by Bayes rule. The approximation would be exact if the “variational” data and observation noise $\tilde{\mathbf{y}}, \mathbf{\Sigma}$ would match the real data.

[3] This is interesting, because it links the statistical precision from our inference (i.e. the uncertainty we have) to how much numerical precision we need (i.e. to invert the matrix successfully).

ending back at the $O(M^3)$ computational cost, $\mathbf{T}$ was used as a preconditioner, which at least would ensure convergence within a single CG iteration at the optimal value $\mathbf{T} = \mathbf{K}_{\mathbf{uu}}^{-1}$. This procedure is cumbersome as it introduces nested iterations: one inner loop for running CG to compute the bound, and an outer loop for optimising all the parameters. This is difficult to implement, and slow to run.

3.2. Deriving the Inverse-Free Likelihood Parameterisation

Applying the same inverse-free trick in the likelihood parameterisation solves both problems of the former approach. We begin in the same way: by lower-bounding the predictive variance and introducing $\mathbf{T}$, only we do so in the likelihood parameterisation of eq. (6):

$$k_{nn} - \mathbf{k}_{n\mathbf{u}}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1}\mathbf{k}_{\mathbf{u}n} \leq k_{nn} + \mathbf{k}_{n\mathbf{u}}\mathbf{T}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})\mathbf{T}\mathbf{k}_{\mathbf{u}n} - 2\mathbf{k}_{n\mathbf{u}}\mathbf{T}\mathbf{k}_{\mathbf{u}n} \tag{10}$$

In this case, equality will be obtained when $\mathbf{T} = (\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1}$, which is much more numerically stable, as discussed in section 2.2.

Next, our goal is to remove log-determinant terms from the $\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]$ computation. Our likelihood parameterisation allows us to derive an upper bound to the KL without matrix decompositions. We start by considering whether the upper bound on the predictive variance in eq. (10) could have been obtained by a particular choice of variational distribution $q(\mathbf{u})$. We find this by equating the upper bound in eq. (10), with the predictive variance term in the marginal parameterisation from eq. (2), and solving for $\mathbf{S}$:

$$k_{nn} + \mathbf{k}_{n\mathbf{u}}\mathbf{T}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})\mathbf{T}\mathbf{k}_{\mathbf{u}n} - 2\mathbf{k}_{n\mathbf{u}}\mathbf{T}\mathbf{k}_{\mathbf{u}n} = k_{nn} - \mathbf{k}_{n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n} + \mathbf{k}_{n\mathbf{u}}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{k}_{\mathbf{u}n} \tag{11}$$

We find the $\mathbf{S}$ that results in our upper bounded predictive variance in two forms:

$$\begin{aligned}\mathbf{S} &= \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}(2\mathbf{T} - \mathbf{T}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})\mathbf{T})\mathbf{K}_{\mathbf{uu}} &(12)\\ &= \mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1}\mathbf{K}_{\mathbf{uu}} + \\ &\quad \underbrace{\mathbf{K}_{\mathbf{uu}}(\mathbf{T} - (\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1})(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})(\mathbf{T} - (\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1})\mathbf{K}_{\mathbf{uu}}}_{=\mathbf{\Delta}\geq 0}\,. &(13)\end{aligned}$$

We obtain one compact form, and one form that shows that it is equivalent to the likelihood parameterisation, but with an additional PSD term added.

We now find $\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]$ for our new inverse-free parameterisation by substituting in eqs. (12) and (13) and $\mathbf{m} = \mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu}$ from eq. (5) into eq. (4):

$$\begin{aligned}\mathrm{KL} &= \frac{1}{2}\Big(\mathrm{Tr}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}) + \boldsymbol{\mu}^\intercal\mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu} - M + \log|\mathbf{K}_{\mathbf{uu}}| - \log|\mathbf{S}|\Big)\\ &= \frac{1}{2}\Big(\mathrm{Tr}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}) + \boldsymbol{\mu}^\intercal\mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu} - M + \log|\mathbf{K}_{\mathbf{uu}}| - \log\big|\mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1}\mathbf{K}_{\mathbf{uu}} + \mathbf{\Delta}\big|\Big)\end{aligned} \tag{14}$$

By using the fact that $\log|\mathbf{A} + \mathbf{\Delta}| \geq \log|\mathbf{A}|$ if $\mathbf{A}, \mathbf{\Delta}$ are PSD, we can bound the KL:

$$\begin{aligned}\mathrm{KL} &\leq \frac{1}{2}\Big(\mathrm{Tr}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}) + \boldsymbol{\mu}^\intercal\mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu} - M + \log|\mathbf{K}_{\mathbf{uu}}| - \log\big|\mathbf{K}_{\mathbf{uu}} - \mathbf{K}_{\mathbf{uu}}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})^{-1}\mathbf{K}_{\mathbf{uu}}\big|\Big)\\ &= \frac{1}{2}\Big(\mathrm{Tr}(\mathbf{K}_{\mathbf{uu}}^{-1}\mathbf{S}) + \boldsymbol{\mu}^\intercal\mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu} - M + \log|\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma}| - \log|\mathbf{\Sigma}|\Big)\\ &= \frac{1}{2}\Big(\mathrm{Tr}((\mathbf{T}(\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma})\mathbf{T} - 2\mathbf{T})\mathbf{K}_{\mathbf{uu}}) + \boldsymbol{\mu}^\intercal\mathbf{K}_{\mathbf{uu}}\boldsymbol{\mu} + \log|\mathbf{K}_{\mathbf{uu}} + \mathbf{\Sigma}| - \log|\mathbf{\Sigma}|\Big)\,.\end{aligned} \tag{15}$$

Simply by reparameterisation, the inverse in the trace from eq. (7) has been removed. Now, we apply the bound $\log|\mathbf{A}| \leq \mathrm{Tr}(\mathbf{A}-\mathbf{I})$ to remove the final problematic $\log|\mathbf{K_{uu}} + \mathbf{\Sigma}|$ term:

$$\begin{aligned}\log|\mathbf{K_{uu}} + \mathbf{\Sigma}| &= \log|(\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T}| - \log|\mathbf{T}| && (16)\\ &\leq \mathrm{Tr}((\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T} - \mathbf{I}) - \log|\mathbf{T}|\,. && (17)\end{aligned}$$

By applying the bound to $(\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T}$, we ensure that equality holds when $\mathbf{T} = (\mathbf{K_{uu}} + \mathbf{\Sigma})^{-1}$, at which point we recover the original likelihood parameterisation.

3.3. Summarising the Inverse-Free Likelihood Parameterisation

Through our derivation, we obtained a marginal likelihood bound for a GP model that **1)** is more numerically stable, and **2)** does not require iterative procedures for computing the KL. It can be obtained in two steps. First, we use eqs. (6) and (12) to reparameterise the approximate posterior, which also implies an inverse-free predictive distribution:

$$q(\mathbf{u}) = \mathcal{N}\Big(\mathbf{u};\quad \underbrace{\mathbf{K_{uu}}\boldsymbol{\mu}}_{\mathbf{m}},\quad \underbrace{\mathbf{K_{uu}} - \mathbf{K_{uu}}(2\mathbf{T} - \mathbf{T}(\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T})\mathbf{K_{uu}}}_{\mathbf{S}}\Big)\,, \tag{18}$$

$$q_{\mathrm{iflp}}(f(\mathbf{x}_n)) = \mathcal{N}(f(\mathbf{x}_n);\quad \mathbf{k}_{n\mathbf{u}}\boldsymbol{\mu},\quad k_{nn} + \mathbf{k}_{n\mathbf{u}}\mathbf{T}(\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T}\mathbf{k}_{\mathbf{u}n} - 2\mathbf{k}_{n\mathbf{u}}\mathbf{T}\mathbf{k}_{\mathbf{u}n}) \tag{19}$$

Next, we remove matrix decompositions by applying additional bounds to the KL:

$$\mathcal{L}_{\mathrm{iflp}} = \sum_{n=1}^{N} \mathbb{E}_{q_{\mathrm{iflp}}(f(\mathbf{x}_n))}[\log p(y_n|f(\mathbf{x}_n))] - \mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})]\,, \tag{20}$$

$$\begin{aligned}\mathrm{KL}[q(\mathbf{u})||p(\mathbf{u})] \leq \frac{1}{2}\Big(&\mathrm{Tr}[(\mathbf{T}(\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T} - 2\mathbf{T})\mathbf{K_{uu}}] + \\ &\boldsymbol{\mu}^{\intercal}\mathbf{K_{uu}}\boldsymbol{\mu} + \mathrm{Tr}[(\mathbf{K_{uu}} + \mathbf{\Sigma})\mathbf{T} - \mathbf{I}] - \log|\mathbf{T}| - \log|\mathbf{\Sigma}|\Big)\,. && (21)\end{aligned}$$

Training now requires optimising $\mathbf{T}$ along with the existing variational parameters $\boldsymbol{\mu}, \mathbf{\Sigma}$. To ensure the final log determinants are efficiently computable, we parameterise $\mathbf{T}$ and $\mathbf{\Sigma}$ using their Cholesky decompositions. We note that when $\mathbf{T}$ is optimised to its optimal value of $(\mathbf{K_{uu}} + \mathbf{\Sigma})^{-1}$, the predictive variance and KL bounds are both equalities. This implies that if optimisation is successful, no performance is lost by using this inverse-free bound.

4. Results

To test the behaviour of the method, we train a GP regression model with the original SVGP bound, and the our inverse free bound (RSVGP). To train the model, we follow the common procedure of using autodiff for finding gradients, and Adam to perform optimisation. We note that our new RSVGP bound was significantly easier to implement than that of van der Wilk et al. (2020), because the latter required custom gradient ops with CG inner loops.

Figure 1 shows a fit on a toy 1D dataset, together with a plot of the ELBO objective with time. We see that the approximations are all very similar, indicating that the inverse free bound is behaving similarly to the commonly used SVGP approximation (Hensman et al., 2013). However, even for this simple example, we note that our inverse-free method takes more iterations to converge than the existing SVGP.

Significant problems arise with training when running RSVGP on the more realistic kin40nm UCI dataset (fig. 2). We notice that hyperparameters converge much more slowly for RSVGP than SVGP, which leads to poor performance. In particular, there is very strange behaviour of the hyperparameters moving in the correct direction, to only reverse direction suddenly. This is reflected by a metric of how good the variational parameter $\mathbf{T}$ approximates $(\mathbf{K_{uu}} + \mathbf{\Sigma})^{-1}$. It seems that very sudden changes in the quality of the approximation occur, which significantly hinders optimisation.

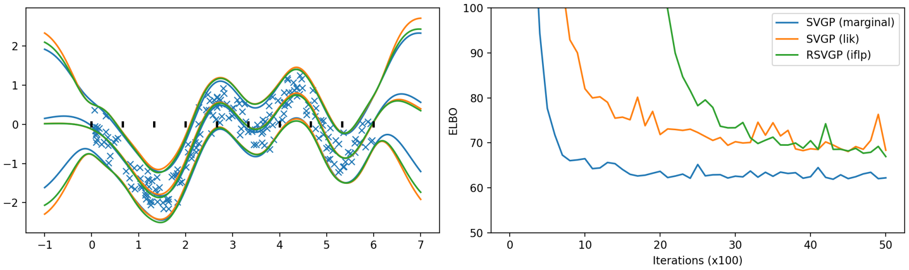

Figure 1: Fit on 1D “snelson” dataset, with optimisation traces of the ELBO.

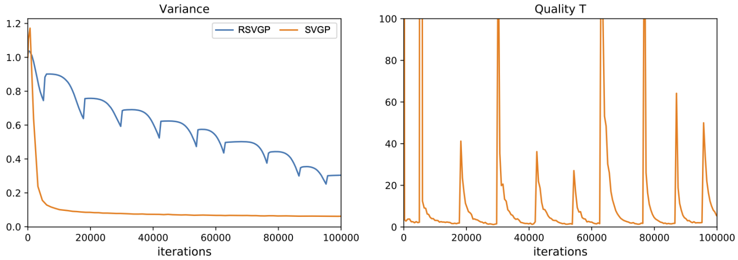

Figure 2: Optimisation of SVGP and RSVGP on kin40nm UCI dataset. Left: variance hyperparameter, right: a measure of the quality of the variational matrix T.

5. Discussion

We introduced a new and more convenient lower bound on the GP marginal likelihood that does not require matrix decompositions to compute. While our method works on simple examples, its optimisation behaviour on more realistic datasets is currently not acceptable for a practically useful method. The existence of these bounds hints at exciting possibilities, but significant optimisation challenges do need to be overcome. One ray of hope, is that currently the most naive optimisation procedure is being used. Perhaps a more tailored approach is needed.

References

Thang D. Bui, Josiah Yan, and Richard E. Turner. A unifying framework for Gaussian process pseudo-point approximations using power expectation propagation. *Journal of Machine Learning Research*, 2017.

Vincent Dutordoir, James Hensman, Mark van der Wilk, Carl Henrik Ek, Zoubin Ghahramani, and Nicolas Durrande. Deep neural networks as point estimates for deep Gaussian processes. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*. 2021.

Maurizio Filippone and Raphael Engler. Enabling scalable stochastic gradient-based inference for Gaussian processes by employing the Unbiased LInear System SolvEr (ULISSE). In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015.

James Hensman, Nicoló Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. *Journal of Machine Learning Research*, 2015.

Alexander G. de G. Matthews. *Scalable Gaussian process inference using variational methods.* PhD thesis, University of Cambridge, 2017.

Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Aristeidis Panos, Petros Dellaportas, and Michalis K Titsias. Fully scalable Gaussian processes using subspace inducing inputs. *arXiv preprint arXiv:1807.02537*, 2018.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning. 2006.* The MIT Press, Cambridge, MA, USA, 2006.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems 30 (NIPS)*. 2017.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

Mark van der Wilk. *Sparse Gaussian process approximations and applications.* PhD thesis, University of Cambridge, 2019.

Mark van der Wilk, ST John, Artem Artemev, and James Hensman. Variational Gaussian process models without matrix inverses. In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference (AABI)*, Proceedings of Machine Learning Research. PMLR, 2020.