Stochastic Variational Gaussian Process Regression for Big Data Applications

Josiah Lim

May 4, 2024
Course: 553.724 Probabilistic Machine Learning (Final Project)
Instructor: Holden Lee

Abstract

This project focuses on the paper by Hensman et al. [2013], which combines stochastic variational inference with Gaussian process regression, along with a reformulation of the model that introduces sparsity for scalable computation. This paper summarizes their work, reorganizes some of their prose, includes additional motivation for the approach, and conducts numerical experiments to test the effectiveness of their method. We conclude that the proposed technique is mathematically interesting and rich, with many connections to the class material, optimization, variational inference, and applied mathematics broadly. However, the practical performance of the proposed method is unconvincing because of sensitivity of an optimization step to initializations. Improving upon the authors' work is a new research project on its own.[1]

1 Introduction

1.1 Gaussian Process Regression

Gaussian processes is a probabilistic method for learning functions, given noisy observations of the unknown function. The strength of this method primarily lies in the intepretability of the approach, which is Bayesian statistics. However, an obvious limitation is in the computational complexity of the inference, for it requires matrix inversions on matrices with dimensions size of the data.

Thus, there has been work in around the last 20 years to scale Gaussian processes for large data. This project learns one such approach: stochastic variational inference Gaussian process models.

[1] This final project topic deviates from the initial proposal to read Gal and Turner [2015] and Jung and Park [2019] because there were not ready-made packages online for them. We sought for this project to be more about understanding the blend of variational inference with GPR, than about coding from scratch. Therefore, we chose a new topic, which is arguably more exciting, actually.

1.2 Outline of project

The project consisted of:

1. **Understanding** the variational formulation of GPR as given by the main paper of interest, Hensman et al. [2013].
2. **Fill in intermediate calculation** not stated in the paper.
3. **Supplement motivation for the proposed method** by consulting cited papers and external resources.
4. **Conduct numerical experiments** to understand the effectiveness and limitations of the proposed method.

The main ideas of the method proposed in Hensman et al. [2013] are:

1. **Sparsity.** Carefully selecting the amount of data used to make inferences and reducing dimensionality of problem.
2. **Variational Inference.** Reframe the objective function in typical GPR to a variational problem that serves as a "good enough" approximation.
3. **Stochastic Optimization.** Apply methods in SVI for GPR for scalability to large datasets.

Code is available at: github.com/limyutaro/724-pml-svigp.

2 Vanilla Gaussian Process Regression (GPR)

The following summary for ordinary GPR is adapted from Rasmussen and Williams [2005].

2.1 Model

Let there be n observations (training data) of an unknown function f, modelled by $y_i = f(x_i) + \varepsilon_i$, where $f : \mathbb{R} \to \mathbb{R}$ and i.i.d. noise $\varepsilon_i \sim \mathcal{N}\left(0, \beta^{-1}\right)$. The conditional distribution of observations y_i given the true function f is thus

$$\mathbf{y}|\mathbf{f} \;\sim\; \mathcal{N}\left(\mathbf{f},\; \beta^{-1}\mathbf{I_n}\right),$$

where $\mathbf{y} = (y_1, \ldots, y_n)$ and $\mathbf{f} = (f(x_1), \ldots, f(x_n))$.

Since we do not know the function f, we model its function values $\mathbf{f}$ by a zero-mean[2] normal distribution

$$\mathbf{f} \;\sim\; \mathcal{N}\left(\mathbf{0},\; \mathbf{K_{xx}}\right),$$

[2] Other means can be chosen, but a zero-mean is used here for simplicity. The method generalizes for any arbitrary mean.

where $\mathbf{K_{xx}}$ is a matrix whose entries are determined by a covariance function $k(x, x')$, i.e., $[\mathbf{K_{xx}}]_{i,j} = k(x_i, x_j)$.

Morally, the covariance function captures the covariance/correlation between points x and x', and is often a function of the distance between points $|x - x'|$. Here is a simplified definition for covariance functions that suffices for the scope of this project.

Definition 2.1 (Covariance function)**.** *A covariance function $k(x, x') : \mathbb{R}^2 \to \mathbb{R}$ is a function that is:*

- ***Symmetric.*** $k(x, x') = k(x', x)$.
- ***Positive definite.*** $\int f(x)k(x, x')f(x')\ d\mu(x)\ d\mu(x') \geq 0$ *for all* $f \in L_2(\mathbb{R}, \mu)$.

Covariance functions often have parameter values, denote as λ, which can be a scalar or vector depending on the context.

Example 2.1. *Examples of covariance functions with pictures in Figure 1 are:*

1. ***Squared exponential/radial basis/Gaussian.***

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

2. ***Rational quadratic.***

$$k(x, x'; \lambda_1, \lambda_2) = \left(1 + \frac{|x - x'|^2}{2\lambda_1\lambda_2^2}\right)^{\lambda_1}.$$

3. ***Periodic.***

$$k(x, x'; \lambda_1, \lambda_2) = \exp\left(-\frac{2\sin^2\left(\frac{|x-x'|}{2\lambda_2}\right)}{\lambda_1^2}\right).$$

4. ***Nonnegative sums*** *of covariance functions. For example,* ***locally periodic.***

$$k(x, x'; \lambda_1, \lambda_2) = \exp\left(-\frac{|x - x'|^2}{2\lambda_1}\right) \cdot \exp\left(-\frac{2\sin^2\left(\frac{|x-x'|}{2\lambda_2}\right)}{\lambda_1^2}\right).$$

5. ***Products*** *of covariance functions.*

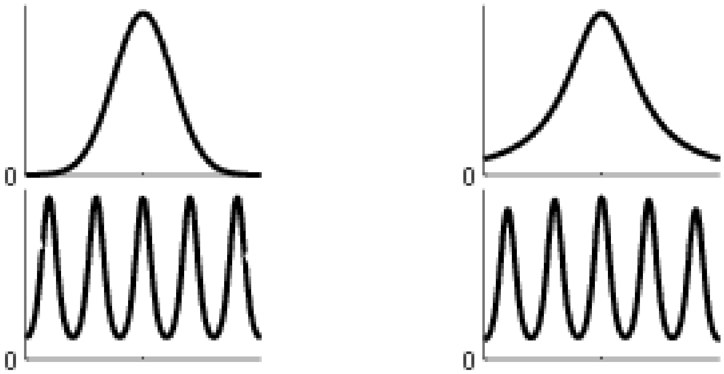


Figure 1: Covariance functions. Top left: Squared exponential/radial basis/Gaussian, often the first covariance function one learns. Top right: Rational quadratic, similar to squared exponential but with heavier tails. Bottom left: Periodic, allows covariance to be large between points at a specific period away. Bottom right: Locally periodic, like periodic but accounts for tapering covariance as distance between points becomes further. Taken from Duvenaud.

2.2 Generating predictions and confidence intervals

A prediction of f at k test points $x_1^*, \ldots, x_k^*$ is typically performed by computing a conditional distribution on test values $\mathbf{y}^* = (y_1^*, \ldots, y_k^*)$ given the observations $\mathbf{y}$. Define

- $\mathbf{K_{xx}}$ to be the covariance matrix between training points,
- $\mathbf{K_{x^*x^*}}$ to be the covariance matrix between training points, and
- $\mathbf{K_{xx^*}}$ to be the covariance matrix between training points (rows) and test points (columns). And $\mathbf{K_{x^*x}} = \mathbf{K_{xx^*}}^T$.

Then using Section 5.1, the conditional distribution is

$$\mathbf{y}^* \mid \mathbf{y} \;\sim\; \mathcal{N}\left(\boldsymbol{\mu}^*,\; \boldsymbol{\Sigma}^*\right). \tag{1}$$

where $\boldsymbol{\mu}^* = \mathbf{K_{x^*x}}\left(\mathbf{K_{xx}} + \beta^{-1}\mathbf{I_n}\right)^{-1}\mathbf{y}$ and $\boldsymbol{\Sigma}^* = \mathbf{K_{x^*x^*}} - \mathbf{K_{x^*x}}[\mathbf{K_{xx}} + \beta^{-1}\mathbf{I_n}]^{-1}\mathbf{K_{xx^*}}$.

We take the prediction of $f(\mathbf{x}^*)$ to be $\boldsymbol{\mu}^*$ and obtain confidence intervals for our prediction by extracting the diagonal terms of the covariance matrix $\boldsymbol{\Sigma}^*$.

2.3 Fine-tuning parameters

The covariance function typically has parameters λ that affect the quality of the prediction and confidence intervals. See Figure 2.

The optimal parameters λ can be chosen in any principled way. A typical approach is to seek λ that maximize the log marginal likelihood

$$\log p(\mathbf{y}|\mathbf{x}) = \log \int p(\mathbf{y}|\mathbf{f})p(\mathbf{f}|\mathbf{x})\ d\mathbf{f},$$

which is implicitly a function of λ through the covariance matrix in the distribution of $\mathbf{y}|\mathbf{f}$.

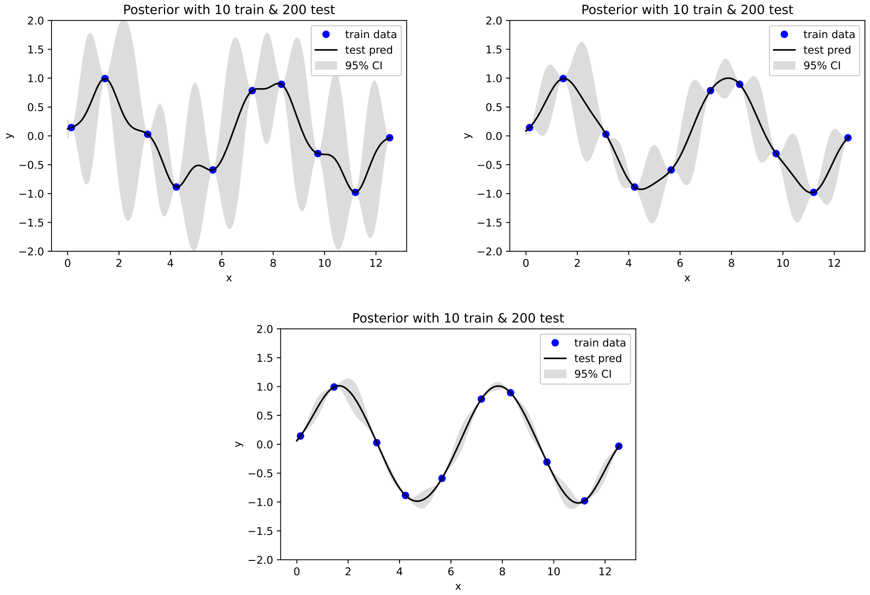


Figure 2: Example of varying covariance function parameter λ for squared exponential/radial basis/Gaussian. Top left: $\lambda = 0.5$. Top right: $\lambda = 1$. Bottom: $\lambda = 2$. λ is typically chosen to Image contains some uncontextualized information because it was used in a previous presentation.

3 Stochastic Variational Inference for Gaussian Process (SVIGP)

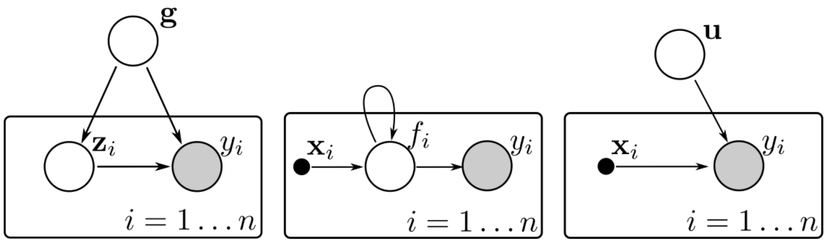
3.1 Overview of SVIGP

SVIGP was developed in the paper Hensman et al. [2013], which combines tools from stochastic variational inference (SVI) (Hoffman et al. [2013]) and sparse inducing point GPR for scalable inference (Titsias [2009]). Unfortunately, we did not have time to thoroughly understand the papers that first introduced SVI and sparse inducing point GPR, but only used selected portions of the texts as references.

A brief description of SVI is also given by [Murphy, 2023, Section 10.1.4] as well.

3.2 Sparse representation: Inducing variables

The purpose of introducing inducing variables is to satisfy a key assumption in SVI, which is the requirement that there are global variables $\mathbf{g}$ that factorize in the observed data $\mathbf{y}$ and latent variables $\mathbf{x}$. This concept is illustrated in the plate notation in Figure 3 and explained in the accompanying description. Besides satisfying this requiement, inducing variables also confer the added benefit of computational scalability in SVIGP by introducing some notion of sparsity.



(a) Requirements for SVI (b) Gaussian Process regression (c) Variational GP regression

Figure 3: Obtained from [Hensman et al., 2013, Figure 1]. Data y_i in shaded circle represents the observed variables; all other variables in white circles represent unobserved or latent variables. (a) shows the graphical model assumed by SVI with global variable $\mathbf{g}$, whereas (b) shows the graphical model used by vanilla GPR without any global variable. To bridge this gap, SVIGP introduced the graphical model in (c), with new inducing (global) variable $\mathbf{u}$.

Definition 3.1 (Inducing variables)**.** *Let there be m inducing (random) variables $\mathbf{u} = (u_1, \dots, u_m)$, which represent values of unknown function f at the points $\mathbf{z} = \{z_1, \dots, z_m\} \subset \mathbf{x}$ subset of training data.*

We place a prior on inducing variables $\mathbf{u}$,

$$\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{mm}}),$$

where $\mathbf{K_{mm}}$ is the covariance matrix between points $\mathbf{z}$ using any covariance function k of our choice.

Then, we model observed data $\mathbf{f}$ and inducing variables $\mathbf{u}$ as a joint normal

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K_{nn}} & \mathbf{K_{nm}} \\ \mathbf{K_{mn}} & \mathbf{K_{mm}} \end{bmatrix} \right),$$

where $\mathbf{K_{nn}}$, $\mathbf{K_{mn}}$, and $\mathbf{K_{nm}}$ are the respective covariance matrices between training and inducing points. Then, by Section 5.1, the conditional distribution of $\mathbf{f}$ given inducing variables $\mathbf{u}$ is

$$\mathbf{f}|\mathbf{u} \ \sim \ \mathcal{N}\left(\mathbf{K_{nm}}\mathbf{K_{mm}^{-1}}\mathbf{u},\ \tilde{\mathbf{K}}\right), \tag{2}$$

where, for convenience of notation, $\tilde{\mathbf{K}} := \mathbf{K_{nn}} - \mathbf{K_{nm}}\mathbf{K_{mm}^{-1}}\mathbf{K_{mn}}$.

Summarizing, we have the following sparse inducing variables GPR model:

$$\begin{aligned} \mathbf{y}|\mathbf{f} \ &\sim \ \mathcal{N}\left(\mathbf{f},\ \beta^{-1}\mathbf{I_n}\right) && \text{(noisy obs given true)} \\ \mathbf{f}|\mathbf{u} \ &\sim \ \mathcal{N}\left(\mathbf{K_{nm}}\mathbf{K_{mm}^{-1}}\mathbf{u},\ \tilde{\mathbf{K}}\right) && \text{(true given inducing)} \\ \mathbf{u} \ &\sim \ \mathcal{N}\left(\mathbf{0},\ \mathbf{K_{mm}}\right). && \text{(inducing)} \end{aligned} \tag{3}$$

3.3 Variational Approach to Likelihood Estimation

Recall from Section 2.3 that we seek to fine-tune the parameters λ in the covariance function $k(x_i, x_j; \lambda)$ in order to produce predictions and confidence intervals that are optimal and meaningful for applications, in the sense of choosing parameter λ that maximize the log marginal likelihood $\log p(\mathbf{y}|\mathbf{x}; \lambda)$.

However, in SVIGP, we will not directly maximize $\log p(\mathbf{y}|\mathbf{x}; \lambda)$, but instead maximize a lower bound $\mathcal{L}_2$ (to be defined) that is itself parameterized by variational distribution $q(\mathbf{u})$ on the inducing variables, which itself is further parameterized by mean and covariance matrices $\mathbf{m}(\lambda)$ and $\mathbf{S}(\lambda)$ that are functions of the parameter λ.[3]

3.4 Derivation of variational lower bound $\mathcal{L}_2$

First, we define an intermediate lower bound $\mathcal{L}_1$ for the log density of observations given the inducing variables

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{u}) &= \log \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}\left[p(\mathbf{y}|\mathbf{f})\right] && \text{(marginalizing and chain rule)} \\ &\geq \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right] && \text{(Jensen's)} \\ &=: \mathcal{L}_1. \end{aligned} \tag{4}$$

We use $\mathcal{L}_1$ and introduce variational distribution $q(\mathbf{u})$ to define the desired lower bound $\mathcal{L}_2$ (with intermediate steps outlined in Section 5.2)

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{x}) &\geq \mathbb{E}_{q(\mathbf{u})}\left[\mathcal{L}_1 + \log \frac{p(\mathbf{u})}{q(\mathbf{u})}\right] \\ &=: \mathcal{L}_2. \end{aligned} \tag{5}$$

[3] In the subsequent sections that derive the lower bound $\mathcal{L}_2$, we will suppress the dependence on parameter λ for brevity.

3.5 Formulating new objective function

From Section 5.3, we can see that the optimal variational distribution $q(\mathbf{u})$ is Gaussian, hence we can parameterize[4] q by mean $\mathbf{m}$ and covaraince $\mathbf{S}$

$$q(\mathbf{u}) \;\sim\; \mathcal{N}\left(\mathbf{m}, \mathbf{S}\right). \tag{6}$$

Then applying normality of q to Equation (5), we see that

- the first term can be simplified (steps shown in Section 5.4),
- the second term is the KL divergence from q to p.

So, we can rewrite $\mathcal{L}_2$ as

$$\begin{aligned}\mathcal{L}_2 \;=\; & \sum_{i=1}^{n}\left[\log\mathcal{N}\left(y_i \mid \mathbf{k}_i^T\mathbf{K}_{\mathbf{mm}}^{-1}\mathbf{m},\ \beta^{-1}\right) - \frac{1}{2}\beta\tilde{\mathbf{K}}_{i,i} - \frac{1}{2}\mathrm{Tr}\left(\mathbf{S}\boldsymbol{\Lambda}_i\right)\right] \\ & - \mathrm{KL}\left(q(\mathbf{u}) \parallel p(\mathbf{u})\right),\end{aligned} \tag{7}$$

where $\mathbf{k}_i$ is the ith column of $\mathbf{K}_{\mathbf{mn}}$ and $\boldsymbol{\Lambda}_i := \beta\ \mathbf{K}_{\mathbf{mm}}^{-1}\ \mathbf{k}_i\mathbf{k}_i^T\ \mathbf{K}_{\mathbf{mm}}^{-1}$.

This is nice because the new objective function $\mathcal{L}_2$, using $q(\mathbf{u})$, is decomposed as a sum over the data $\{x_i, y_i\}_{i=1}^n$. Hence, we can take stochastic gradients, which requires the objective function to be "separable" in terms of the individual datum.

3.6 Natural Gradients

We have now shifted our goal to optimizing $\mathcal{L}_2$ instead, which is simply an optimization over variational distribution q, which in turn is fully determined by its parameters $\mathbf{m}, \mathbf{S}$.

However, instead of iteratively optimizing in the parameter space of $\mathbf{m}$ and $\mathbf{S}$, SVI works by optimizing with two key modifications:

1. **Canonical parameter space.** Optimize in the canonical parameter space $\boldsymbol{\theta}$, which for a normal distribution $\mathcal{N}(\mathbf{m}, \mathbf{S})$ is the reparameterization
$$\boldsymbol{\theta}_1 \;=\; \mathbf{S}^{-1}\mathbf{m}, \qquad \boldsymbol{\theta}_2 \;=\; -\frac{1}{2}\mathbf{S}^{-1}.$$
2. **Take natural gradients.** We do not take steps in the usual gradient $\mathbf{g}(\boldsymbol{\theta}) := \frac{\partial\mathcal{L}_2}{\partial\boldsymbol{\theta}}$, but in the natural gradient $\tilde{\mathbf{g}}(\boldsymbol{\theta})$, which is the usual gradient rescaled by the Fisher information matrix,
$$\tilde{\mathbf{g}}(\boldsymbol{\theta}) \;:=\; G(\boldsymbol{\theta})^{-1}\mathbf{g}(\boldsymbol{\theta}). \tag{8}$$

Before continuing with the method for SVIGP with natural gradients, we first motivate why natural gradients make sense by citing several other sources:

[4] Note that mean and covariance of $q(\mathbf{u})$, $\mathbf{m}(\lambda)$ and $\mathbf{S}(\lambda)$, depend on parameter λ via the covariance matrix $\mathbf{K}_{\mathbf{mm}}$ as shown in the model outlined in Equation (3).

- Natural gradient is a generalization of the usual gradient by taking the curvature of the objective function ($\mathcal{L}_2$) into account. Usual (stochastic) gradient takes steps in high curvature areas, and makes slow progress in low curvature areas, see Figure 4.

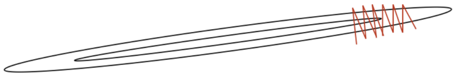


Figure 4: Taken from Grosse lecture slides. Usual gradient descent seeks high curvature steps, which may not necessarily be appropriate for optimizing functions whose optima requires low-curvature steps for faster convergence.

- Usual gradients can be viewed as gradients in the ℓ_2 norm; natural gradients can be viewed as gradients with the KL-divergence, thus more "friendly" and suitable for optimization over probability distributions.
- When performing a Taylor expansion of the KL-divergence, the Fisher matrix is seen as an approximation to the Hessian, Grosse; Jones.

3.7 Expectation parameter trick for SVIGP

Despite the benefit of natural gradients, we may be bothered by the matrix inversion required from the inverse Fisher matrix. The magic of Hensman et al. [2012] is in the simplification of the optimization update steps by considering the expectation (moment) parameters of a normal distribution, which are

$$\boldsymbol{\eta}_1 \; = \; \mathbf{m}, \qquad \boldsymbol{\eta}_2 \; = \; \mathbf{m}\mathbf{m}^T + \mathbf{S}.$$

They show that natural gradient steps simplifies to a usual gradient over the expectation parameters

$$\tilde{\mathbf{g}}(\boldsymbol{\theta}) \; = \; G(\boldsymbol{\theta})^{-1}\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\theta}} \; = \; \frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\eta}}. \tag{9}$$

Thus, we win computationally. Using step size [5] h for the parameter update of the form $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + h\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\eta}}$, the update steps (details in Section 5.5) for each canonical parameter are now just

$$\begin{aligned} \boldsymbol{\theta}_{1,t+1} \; &= \; \boldsymbol{\theta}_{1,t} + h\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\eta}_1} \\ &= \; \boldsymbol{\theta}_{1,t} + h\left(\beta \mathbf{K}_{\mathbf{mm}}^{-\mathbf{1}}\mathbf{K}_{\mathbf{mn}}\mathbf{y} - \boldsymbol{\theta}_{1,t}\right) \\ \boldsymbol{\theta}_{2,t+1} \; &= \; \boldsymbol{\theta}_{2,t} + h\frac{\partial \mathcal{L}_2}{\partial \boldsymbol{\eta}_2} \\ &= \; \boldsymbol{\theta}_{2,t} + h\left(-\frac{1}{2}\boldsymbol{\Lambda} - \boldsymbol{\theta}_{1,t}\right), \end{aligned} \tag{10}$$

[5] Setting $h = 1$ for natural gradient step recovers a coordinate update for usual gradients Hoffman et al. [2013].

where $\boldsymbol{\theta}_{1,t} = -\mathbf{S}_{(t)}^{-1}\mathbf{m}_{(t)}$, $\boldsymbol{\theta}_{2,t} = -\frac{1}{2}\mathbf{S}_{(t)}^{-1}$, and $\boldsymbol{\Lambda} = \beta\mathbf{K_{mm}^{-1}}\mathbf{K_{mm}}\mathbf{K_{nm}}\mathbf{K_{mm}^{-1}} + \mathbf{K_{mm}^{-1}}$.

A nice feature of using natural gradients for the update of $\boldsymbol{\theta}_2$ is that every iterate is a positive definite matrix Hensman et al. [2012, 2013]. Therefore, the reparametrization allows us to remove the positive definite constraint on $\mathbf{S}$ and implicitly bake it in to the natural gradient.

3.8 Summarizing overall SVIGP method

We have reframed the log marginal likelihood optimization as an optimization instead on the variational lower bound $\mathcal{L}_2$. The benefit of this is creating an objective function that relies on only a sparse subset of data $\mathbf{u}$ that is smaller than the full data $\mathbf{x}$, though at the cost of introducing an additional variational inference problem.

But now that $\mathbf{u}$ is a global variable, we may leverage SVI to efficiently optimize the variational parameters $\mathbf{m}$ and $\mathbf{S}$ via the reparametrization to the canonical and expectation parameters, with natural gradients.

In principle, with the more tractable variational objective function $\mathcal{L}_2$ instead of $\log p(\mathbf{y}|\mathbf{x})$, we can now optimize the GPR covariance function parameters λ jointly with the variational parameters of $\mathcal{L}_2$ using stochastic gradient descent.

4 Numerical Experiments

We compare SVIGP to vanilla GPR by showing the predictive plots obtained from both methods, using the package GPyTorch, Gardner et al. [2018]. We considered real data of the atmospheric CO_2 concentration made at Mauna Loa, Hawaii from 1958 to 2002.

The data has been rescaled in the y-axis by the function $r(y) = 0.2(y - 300)$ prior to feeding it into the package, which makes the function values smaller and was found to give better predictions, possibly due to numerical reasons. Right before plotting, we rescaled it back to original units by r^{-1}.

4.1 Experiment setup

The data sets have training size $n = 88$. The number of inducing points for SVIGP is $m = 29$, about one-third of the training data.

For all experiments, we fixed the number of iterations to be constant $k_0 = 3000$.

- For vanilla GP, the number of epochs is also k_0.
- For SVIGP, we fix the mini batch size to be 10. Hence, the number of iterations per epoch is 9, so the number of epochs is $k_0/9 \approx 333$.

4.2 Custom covariance function

The covariance function used was a customized sum of four covariance functions, which we have taken from [Rasmussen and Williams, 2005, Chapter 5]. There is a total of 11 parameters $\lambda_1, \dots, \lambda_{11}$. The reason for using this customized covariance function is to better model the rising and periodic nature of the data.

$$k(x, x'; \lambda) \;:=\; k_1(x, x'; \lambda_{1:2}) + k_2(x, x'; \lambda_{3:6}) + k_3(x, x'; \lambda_{7:9}) + k_4(x, x'; \lambda_{10:11})$$

where the components are defined by

$$k_1(x, x'; \lambda_{1:2}) = \lambda_1^2 \, \exp\left(-\frac{|x - x'|^2}{2\,\lambda_2}\right) \qquad \text{(Long term rising)}$$

$$k_2(x, x'; \lambda_{3:6}) = \lambda_3^2 \, \exp\left(-\frac{|x - x'|^2}{2\,\lambda_4} - \lambda_5 \, \sin^2\left(\frac{\pi\,|x - x'|}{\lambda_6}\right)\right) \qquad \text{(Periodicity)}$$

$$k_3(x, x'; \lambda_{7:9}) = \lambda_7^2 \left[1 + \frac{|x - x'|^2}{2\,\lambda_8\,\lambda_9}\right]^{-\lambda_8} \qquad \text{(Medium term irregularities)}$$

$$k_4(x, x'; \lambda_{10:11}) = \lambda_{10}^2 \, \exp\left(-\frac{|x - x'|^2}{2\,\lambda_{11}}\right) \qquad \text{(Short term smoothing)}$$

4.3 Prediction Plots

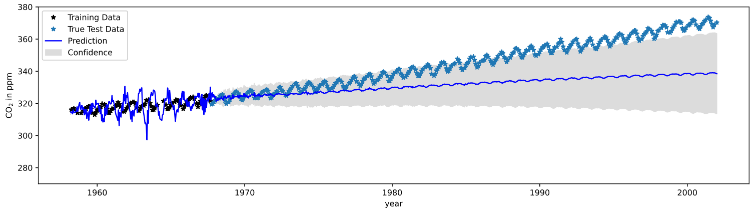


Figure 5: Vanilla GP with custom kernel on CO_2 data.

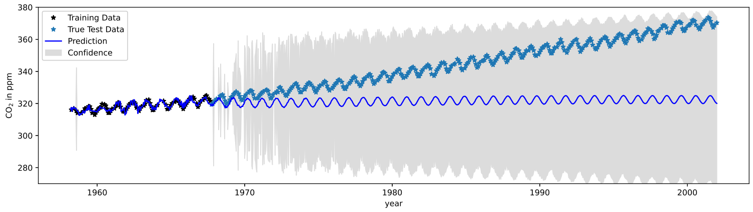


Figure 6: SVIGP with custom kernel on CO_2 data.

4.4 Discussion on Experiment results

We see that while the the number of iterations was fixed and the runtime for vanilla GP and SVIGP were both similar at about 40 seconds, the mean prediction for SVIGP fits more desirably. The true data within the SVIGP confidence interval, it is outside that of vanilla GP. Moreover, the periodicity is much more apparent in the SVIGP prediction, which indicates that it learned the periodic patterns better.

While it is difficult to discern the exact cause for the better fit, we may remark that Titsias [2009] mentions the potential strength of sparse representations in overcoming overfitting.

However, we can also see that the confidence region for SVIGP is much wider than vanilla GP, indicating that the posterior distribution has larger variance and the predictions made by SVIGP method does not yield a confident prediction of the future CO_2 trajectory.

That said, neither methods seem to satisfactorily make good predictions beyond 5 years of the training data. One may play this down to initialization of the parameter optimization, which is known weakness of GPR.

References

D. Duvenaud. The kernel cookbook: Advice on covariance functions, howpublished = https://www.cs.toronto.edu/~duvenaud/cookbook/. accessed: 2024-05-04.

Y. Gal and R. Turner. Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 655–664, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/galb15.html.

J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018.

R. Grosse. Natural gradient, csc2541 lecture 5. https://csc2541-f17.github.io/slides/lec05a.pdf. accessed: 2024-05-02.

J. Hensman, M. Rattray, and N. D. Lawrence. Fast variational inference in the conjugate exponential family. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'12, page 2888–2896, Red Hook, NY, USA, 2012. Curran Associates Inc.

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In A. Nicholson and P. Smyth, editors, *Uncertainty in Artificial Intelligence*, volume 29. AUAI Press, 2013. URL http://inverseprobability.com/publications/gaussian-processes-for-big-data.html.

M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347, may 2013. ISSN 1532-4435.

A. Jones. Natural gradients. https://andrewcharlesjones.github.io/journal/natural-gradients.html. accessed: 2024-05-02.

Y. Jung and J. Park. Spectral mixture kernel approximation using reparameterized random fourier feature. In *Second Symposium on Advances in Approximate Bayesian Inference*, 2019. URL https://openreview.net/forum?id=HJlvKy3VFS.

K. P. Murphy. *Probabilistic Machine Learning: Advanced Topics*. MIT Press, 2023. URL http://probml.github.io/book2.

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 11 2005. ISBN 9780262256834. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

M. Titsias. Variational learning of inducing variables in sparse gaussian processes. In D. van Dyk and M. Welling, editors, *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 567–574, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/titsias09a.html.

5 Appendix

5.1 Conditional probability for normal distributions

This formula was invoked in Equation (1) and Equation (2).

Given the following joint distribution

$$\begin{bmatrix} \mathbf{Y} \\ \mathbf{Y}^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix} \right),$$

conditioning on $\mathbf{Y}$ gives

$$\mathbf{Y}^* | \mathbf{Y} = \mathbf{y} \sim \mathcal{N}\big(K_3 K_1^{-1} \mathbf{y},\ K_4 - K_3 K_1^{-1} K_2\big). \tag{11}$$

5.2 Defining $\mathcal{L}_2$

This derivation works out Equation (5).

$$\begin{aligned}
\log p(\mathbf{y}|\mathbf{x}) &= \log \int p(\mathbf{y}, \mathbf{u}|\mathbf{x}) p(\mathbf{u}|\mathbf{x})\ d\mathbf{u} && \text{(marginalize, todo: x?)} \\
&\geq \log \int \exp\left(\mathcal{L}_1\right) p(\mathbf{u})\ d\mathbf{u} && \text{(Equation (4), } \mathbf{u} \perp \mathbf{x}\text{)} \\
&= \log \int \exp\left(\mathcal{L}_1\right) \frac{p(\mathbf{u})}{q(\mathbf{u})} q(\mathbf{u})\ d\mathbf{u} \\
&\geq \int \left(\mathcal{L}_1 + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right) q(\mathbf{u})\ d\mathbf{u} && \text{(Jensen's)} \\
&\geq \mathbb{E}_{q(\mathbf{u})} \left[\mathcal{L}_1 + \log \frac{p(\mathbf{u})}{q(\mathbf{u})} \right] \\
&=: \mathcal{L}_2. && \text{(define } \mathcal{L}_2\text{)}
\end{aligned} \tag{12}$$

5.3 Optimal q is Gaussian

Consider Equation (5). todo: unsure why still

5.4 Deriving first term in summation form of $\mathcal{L}_2$

This derivation works out the summation term of Equation (7).

We seek to show

$$\mathbb{E}_{q(\mathbf{u})}\left[\mathcal{L}_1\right] \ = \ \sum_{i=1}^{n} \left[\log \mathcal{N}\left(y_i \mid \mathbf{k}_i^T \mathbf{K}_{\mathbf{mm}}^{\mathbf{-1}} \mathbf{m},\ \beta^{-1}\right) - \frac{1}{2} \beta \tilde{\mathbf{K}}_{i,i} - \frac{1}{2} \mathrm{Tr}\left(\mathbf{S} \boldsymbol{\Lambda}_i\right) \right]$$

Recall from Equation (4) that

$$\mathcal{L}_1 \ := \ \mathbb{E}_{p(\mathbf{f}|\mathbf{u})}\left[\log p(\mathbf{y}|\mathbf{f})\right].$$

Under the assumption that $p(\mathbf{y}|\mathbf{f})$ factorizes across the data $p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} p(y_i|f_i)$, we can show that todo: fill in rest

$$\exp\left(\mathcal{L}_1\right) = \prod_{i=1}^{n} \mathcal{N}\left(y_i \mid \mu_i, \beta^{-1}\right) \exp\left(-\frac{1}{2}\beta \tilde{k}_{i,i}\right),$$

where $\mu_i = \left[\mathbf{K_{nm}K_{mm}^{-1}u}\right]_i$ ith element and $\tilde{k}_{i,i} = [\tilde{\mathbf{K}}]_{i,i}$ diagonal elements. Then, we get

$$\begin{aligned}
\mathbb{E}_{q(\mathbf{u})}\left[\mathcal{L}_1\right] &= \mathbb{E}_{q(\mathbf{u})}\left[\sum_{i=1}^{n}\left(\log \mathcal{N}\left(y_i \mid \mu_i, \beta^{-1}\right) - \frac{1}{2}\beta\tilde{k}_{i,i}\right)\right] \\
&= \sum_{i=1}^{n} \mathbb{E}_{q(\mathbf{u})}\left[\log \mathcal{N}\left(y_i \mid \mu_i, \beta^{-1}\right)\right] - \frac{1}{2}\beta\tilde{k}_{i,i} \\
&= todo : couldnotderivefurther
\end{aligned} \tag{13}$$

5.5 SVI updates

Derive the update steps for Equation (10). todo: was not able to figure out details