Scalable Gaussian Processes[*]

Submitted by:

Flavio Argentieri
Ritwik Vashistha
Supervised by:

Dr. Nhat Ho

January 23, 2025

Abstract

Most real world phenomena occur across both space and time and they exhibit complex dependencies. Consider the issue of modeling housing prices across a city. Such a problem involves large amounts of noisy data with strong spatial (eg. location of houses) and temporal effects. Similar situations demand a probabilistic model, allowing for incorporation of prior information and quantification of uncertainty. Gaussian Processes (GP) provide a highly efficient probabilistic modeling paradigm for working with such complex data. However, their application is limited due to their inherent cubic computational scaling in number of data, N. Recently, a lot of advances have been made to improve computation time of Gaussian processes. We review different approaches to improving scalability of GP based models. Our report is mainly based on Quiñonero Candela and Rasmussen (2005), Hensman et al. (2013) and Wu et al. (2022).

[*] This report has been prepared towards the partial fulfillment of the requirements of the course *SDS 383C: Machine Learning Optimization.*

Contents

1 Introduction

Gaussian Processes (GPs) provide a very flexible approach to model a wide variety of phenomena, thanks to their non-parametric nature. However, this comes at the cost of computation time, which is cubic in nature. We'll review the approaches taken in the literature to tackle such trade-off. Our focus will be on Gaussian Processes as a tool for non-parametric regression. We are interested in modeling the relationship between a real valued response variable y, and a finite set of explanatory variables collected in a vector $\mathbf{x}$ of dimension d. We consider y as a random variable and $\mathbf{x}$ as a random vector, and we assume that there exists a relationship between them, let's say

$$y = f(\mathbf{x})$$

We are interested in making inference about f, and we assume to have data on y, and $\mathbf{x}$. In particular, we assume to have an independent and identically distributed (iid) training set $\{(y_i, \mathbf{x}_i), i = 1, \dots, N\}$. In the following, we will always condition on the sigma algebra generated by the values of $\{\mathbf{x}_i\}_{i=1}^N$, so we will suppress the conditioning in the notation. We consider $f : \mathbb{R}^d \to \mathbb{R}$ as a random function - i.e. a stochastic process - and put a Gaussian Process prior on it

$$f(\cdot) \sim GP\left(m(\mathbf{s}), k\left(\mathbf{s}, \mathbf{s}'\right)\right)$$

where $m(\mathbf{s})$ is the mean function, and $k(\mathbf{s}, \mathbf{s}')$ is the kernel of the Gaussian Process. This amounts to say that, $\forall m \in \mathbb{N}$, $\forall \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m\} \subset \mathbb{R}^d$

$$\begin{pmatrix} f(\mathbf{s}_1) \\ f(\mathbf{s}_2) \\ \vdots \\ f(\mathbf{s}_m) \end{pmatrix} \sim \mathscr{N}\left[\begin{pmatrix} m(\mathbf{s}_1) \\ m(\mathbf{s}_2) \\ \vdots \\ m(\mathbf{s}_m) \end{pmatrix}, \begin{pmatrix} k(\mathbf{s}_1, \mathbf{s}_1) & k(\mathbf{s}_1, \mathbf{s}_2) & k(\mathbf{s}_1, \mathbf{s}_m) \\ k(\mathbf{s}_2, \mathbf{s}_1) & k(\mathbf{s}_2, \mathbf{s}_2) & k(\mathbf{s}_2, \mathbf{s}_m) \\ \vdots & \vdots & \vdots \\ k(\mathbf{s}_m, \mathbf{s}_1) & k(\mathbf{s}_m, \mathbf{s}_2) & k(\mathbf{s}_m, \mathbf{s}_m) \end{pmatrix}\right]$$

For the sake of simplicity we will assume $m(\mathbf{s}) = 0$. Let $X = \left(\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_N^T\right)^T$ The assumptions we made so far imply

$$\mathbf{f} := \begin{pmatrix} f(\mathbf{x}_1) \\ f(\mathbf{x}_2) \\ \vdots \\ f(\mathbf{x}_m) \end{pmatrix} \sim \mathscr{N}\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, K_{\mathbf{ff}} := \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & k(\mathbf{x}_1, \mathbf{x}_m) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & k(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \vdots \\ k(\mathbf{x}_m, \mathbf{x}_1) & k(\mathbf{x}_m, \mathbf{x}_2) & k(\mathbf{x}_m, \mathbf{x}_m) \end{pmatrix}\right)$$

which we will write as

$$\mathbf{f} \sim \mathscr{N}(0, K_{\mathbf{ff}}) \tag{1}$$

In a model without noise, we would also have $\mathbf{y} \sim \mathscr{N}(0, K_{\mathbf{ff}})$, where $\mathbf{y} := (y_1, y_1, \ldots, y_N)^T$. However, we allow for noisy measurements of $\mathbf{y}$, and we write

$$y_i = f(\mathbf{x}_i) + \varepsilon_i$$

where $\varepsilon_i \overset{iid}{\sim} \mathscr{N}(0, \sigma^2)$ for all $i \in \{1, \ldots, N\}$. This amounts to saying that $f(\mathbf{x})$ is the regression function of y on $\mathbf{x}$. In vector notation, we get

$$\mathbf{y} = \mathbf{f} + \varepsilon \tag{2}$$

where $\varepsilon \sim \mathscr{N}(0, \sigma^2 I_N)$ and I_N is the NxN identity matrix. Our likelihood specification follows naturally

$$\mathbf{y} \mid \mathbf{f} \sim \mathscr{N}(\mathbf{f}, \sigma^2 I_N) \tag{3}$$

The problem we will focus on is prediction. We assume to have a single data point in the test set. For such data point we only have the value of $\mathbf{x} = \mathbf{x}^\star$, and we are interested in making inference about $f^\star = f(\mathbf{x}^\star)$. Our Gaussian Process specification implies

$$\begin{pmatrix} \mathbf{f} \\ f^\star \end{pmatrix} \sim \mathscr{N}\left(\begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} k(\mathbf{x}_1,\mathbf{x}_1) & k(\mathbf{x}_1,\mathbf{x}_2) & k(\mathbf{x}_1,\mathbf{x}_m) & k(\mathbf{x}_1,\mathbf{x}^\star) \\ k(\mathbf{x}_2,\mathbf{x}_1) & k(\mathbf{x}_2,\mathbf{x}_2) & k(\mathbf{x}_2,\mathbf{x}_m) & k(\mathbf{x}_2,\mathbf{x}^\star) \\ \vdots & \vdots & \vdots & \vdots \\ k(\mathbf{x}_m,\mathbf{x}_1) & k(\mathbf{x}_m,\mathbf{x}_2) & k(\mathbf{x}_m,\mathbf{x}_m) & k(\mathbf{x}_m,\mathbf{x}^\star) \\ k(\mathbf{x}^\star,\mathbf{x}_1) & k(\mathbf{x}^\star,\mathbf{x}_2) & k(\mathbf{x}^\star,\mathbf{x}_m) & k(\mathbf{x}^\star,\mathbf{x}^\star) \end{pmatrix}\right)$$

and, by partitioning the variance covariance matrix, we can rewrite our joint prior as

$$\begin{pmatrix} \mathbf{f} \\ f^\star \end{pmatrix} \sim N\left(0, \begin{pmatrix} K_{\mathbf{ff}} & K_{\mathbf{f}\star} \\ K_{\mathbf{f}\star}^T & K_{\star\star} \end{pmatrix}\right) \tag{4}$$

Combining (3) and (4) leads to a posterior expression for the joint distribution of $\mathbf{f}$ and $f^\star$

$$p(\mathbf{f}, f^\star \mid \mathbf{y}) \propto p(\mathbf{f}, f^\star)\, p(\mathbf{y} \mid \mathbf{f})$$

that we leave implicit. In turn, this allows to evaluate the predictive posterior by integrating out $\mathbf{f}$

$$p(f^\star \mid \mathbf{y}) = \int p(\mathbf{f}, f^\star \mid \mathbf{y})\, \mathrm{d}\mathbf{f} \propto \int p(\mathbf{f}, f^\star)\, p(\mathbf{y} \mid \mathbf{f}) \mathrm{d}\mathbf{f}$$

The convenient properties of multivariate normal distributions lead to the following result

$$f^* \mid \mathbf{y} \sim N\left(K_{\mathbf{f}\star}^T \left(K_{\mathbf{ff}} + \sigma^2 I_N\right)^{-1} \mathbf{y}, \quad K_{\star\star} - K_{\mathbf{f}\star}^T \left(K_{\mathbf{ff}} + \sigma^2 I_N\right)^{-1} K_{\mathbf{f}\star}\right) \tag{5}$$

The posterior distribution for f^* derived in (5) is exact and comes from a nonparametric, very flexible model. Thus, it is highly attractive. However, actually computing such posterior requires the inversion of a matrix of dimension N, which involves a cubic cost $O(N^3)$. Our main focus will be on surveying methods that have been proposed in the literature to improve on such computational constraint. The first big improvement has been provided by sparse GPs. Sparsity can be induced by reducing the rank of the variance covariance matrix, in such a way that the information loss is bounded, but computations are made easier. There are many ways of inducing sparsity, and they often result from very different procedures. Quiñonero Candela and Rasmussen (2005) present a unifying view of such methods by reframing sparsity as a restriction imposed on the GP prior, allowing for comparison across the different methods. They argue that any sparse approach boils down to selecting a set $\mathbf{u}$ of inducing variables of size m and assuming conditional independence of $\mathbf{f}$ and f^* given $\mathbf{u}$. All these approaches lead to improve the computation time to $O(N \cdot M^2)$, where $M \ll N$, making GPs an effective tool for medium sized datasets. Titsias (2009) provides a variationally inference justified procedure that introduces sparsity as a result of optimization, thus avoiding overfitting and regularizing the estimation. Hensman et al. (2013) improve on such variational approach by maximizing a bound that allows for application of stochastic gradient descent, leading to another big improvement in computation time, which decreases to $O(M^3)$. Wu et al. (2022) uses the idea of K nearest neighbours approach developed in Datta et al. (2016) and combines it with stochastic variation inference to reduce the computation time to $O(K^3)$, where $K \ll M \ll N$.

2 Methods

2.1 Sparse Gaussian Processes

2.1.1 Subset of Regressors (SoR)

Before studying the general framework for sparse GPs, it is useful to present some specific instances of such methods. Smola and Bartlett (2000) present a greedy selection of a subset of regressors that reduces the rank of $K_{\mathbf{ff}}$ so as to speed up the inversion of the variance covariance matrix. First of all, they reparameterize the model. Combining (1) and (2), they write

$$\mathbf{y} = K_{\mathbf{ff}}\alpha + \varepsilon \tag{6}$$

where $\alpha \sim \mathcal{N}(0, K_{\mathbf{ff}}^{-1})$, and $\varepsilon \sim \mathcal{N}(0, I_N)$, which is easily seen as equivalent to the original model (2). This reduces the problem of prediction to that of finding the posterior of the parameter α, which is

$$p(\alpha \mid \mathbf{y}) \propto \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{y} - K_{\mathbf{ff}}\alpha\|^2\right)\exp\left(-\frac{1}{2}\alpha^\top K_{\mathbf{ff}}\alpha\right)$$

Once we have a Maximum A Posteriori (MAP) estimate for α, let's say $\hat{\alpha}$, such estimate allows to compute

$$\hat{f}^\star = K_{\mathbf{ff}}\hat{\alpha}$$

which is the MAP estimate for $f^\star$. Taking the negative of the log posterior for α, our problem then becomes

$$\min_{\alpha \in \mathbb{R}^N}\left[Q(\alpha) := -\mathbf{y}^\top K_{\mathbf{ff}}\alpha + \frac{1}{2}\alpha^\top\left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)\alpha\right] \tag{7}$$

The solution of this problem can be easily shown to be

$$\hat{\alpha} = \left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)^{-1} K_{\mathbf{ff}}^T \mathbf{y}$$

which requires computing the inverse of an NxN matrix, with computational cost $O(N^3)$. Smola and Bartlett introduce sparsity by guessing a form for α, i.e.

$$\alpha = P\beta$$

with $\beta \in \mathbb{R}^M$ and $P \in \mathbb{R}^{NxM}$, with $M \in \mathbb{N}$. This expression amounts to a simple reparameterization if P is of rank N. However, by imposing $\text{rank}(P) = M \ll N$, we are putting a constraint on α. The return for this loss of generality is that we can solve (7) for β and get as a solution

$$\hat{\beta} = \left(P^T \left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right) P\right)^{-1} P^T K_{\mathbf{ff}}^T \mathbf{y}$$

whose computation requires

1. the computation of $P^T K_{\mathbf{ff}}^T$, which is the product of an MxN matrix by an NxN matrix, with a cost $O(MN)$;
2. the computation of $(K_{\mathbf{ff}}P)^T(K_{\mathbf{ff}}P)$, whose computation cost is $O(M^2N)$;
3. the computation of the inverse of $P^T\left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)P$, which is an MxM matrix and thus has a cost of $O(M^3)$.

Under the assumption that $M \ll N$, the dominant cost is $O(M^2N)$, which is a great improvement w.r.t. $O(N^3)$. The challenge is then to choose P in such a way to control the approximation error and still get a matrix with the lowest possible rank. The basic idea is to induce sparsity through the choice of P. If we think of $\mathbf{y}$ in (6) as a linear combination of the columns of $K_{\mathbf{ff}}$, then sparsity amounts to many of the α_i being 0, which can be achieved by choosing matrices P whose columns are unit vectors $\mathbf{e}_j$. The basic idea behind the algorithm that the authors propose is based on a greedy optimization:

1. Initialize the matrix P to be the $Nx1$ matrix whose only column is the unit vector $\hat{e}_i$ minimizing $Q(\alpha) = Q(e_i\hat{\beta})$
2. Expand P by setting it equal to $\left[P, \hat{e}_j\right]$, where $\hat{e}_j$ minimizes $Q(\alpha) = Q(\left[P, \hat{e}_j\right]\hat{\beta})$, and $\hat{e}_j$ is a unit vector that has not been chosen yet
3. Stop when $Q\left(P\hat{\beta}\right) + \sigma^2 Q^*\left(P\hat{\beta}^*\right) + \frac{1}{2}\|\mathbf{y}\|^2 \leq \frac{\varepsilon}{2}\left(\left|Q\left(P\hat{\beta}\right)\right| + \left|\sigma^2 Q^*\left(P\hat{\beta}^*\right) + \frac{1}{2}\|\mathbf{y}\|^2\right|\right)$, where ε is arbitrary, $Q^*(\alpha) := -\mathbf{y}^\top\alpha + \frac{1}{2}\alpha^\top\left(\sigma^2 I + K_{\mathbf{ff}}\right)\alpha$, and consequently $\hat{\beta}^* = \left(P^T\left(\sigma^2 I + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)P\right)^{-1} P^T\mathbf{y}$.

The authors' main result is indeed a bound on the approximation error resulting from the above algorithm.

Theorem 1. *The algorithm described achieves $Q(\alpha) \leq Q(\hat{\alpha}) + \varepsilon$ when α has*

$$n \leq \frac{18n^*(\varepsilon/4)}{\lambda_1^2} \ln\left(\frac{\|A^{-1}K_{\mathbf{ff}}y\|}{\varepsilon}\right)$$

non-zero components, where $A := \left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)^{\frac{1}{2}}$, $n^*(\varepsilon/4)$ is the minimal number of nonzero components in vectors α for which $Q(\alpha) \leq Q(\hat{\alpha}) + \varepsilon/4, A = \left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)^{1/2}$, and λ_1 is the minimum of the magnitudes of the singular values of $\mathbf{A}$, the matrix obtained by normalizing the columns of A.

The authors also show that the complexity of the pseudo algorithm is dominated by the computation of $\left(P^T\left(\sigma^2 K_{\mathbf{ff}} + K_{\mathbf{ff}}^\top K_{\mathbf{ff}}\right)P\right)^{-1}$, so that the overall computational cost is $O(M^2N)$.

2.1.2 Sparse Pseudo-input Gaussian processes (SPGPs)

Sparse pseudo-input Gaussian processes (SPGP) by Snelson and Ghahramani (2005) use an alternative approach to induce sparsity in the model. They employ the idea of using a pseudo data set $\bar{D}$ of size $M \ll N$, which helps in reducing the computational complexity from $O(N^3)$ to $O(NM^2)$. Formally, consider a pseudo data set $\bar{D}$ with pseudo-inputs $\overline{X}$ and pseudo targets $\mathbf{u}$. Then we can write the likelihood for a single point $\mathbf{x}$ as follows

$$p(y \mid \mathbf{x}, \overline{X}, \mathbf{u}) = \mathcal{N}\left(y \mid K_{\mathbf{f}\star}^{\top} K_{\mathbf{uu}}^{-1} \mathbf{u}, K_{\mathbf{ff}} - K_{\mathbf{f}\star}^{\top} K_{\mathbf{uu}}^{-1} K_{\mathbf{f}\star} + \sigma^2\right), \tag{8}$$

where $[K_{\mathbf{uu}}]_{mm'} = K\left(\overline{X}_m, \overline{X}_{m'}\right)$ and $[K_{\mathbf{f}\star}]_m = K\left(\overline{X}_m, \mathbf{x}\right)$, for $m, m' = 1, \ldots, M$.

If we assume that target data are generated i.i.d given the inputs, we can easily obtain the likelihood for complete data as follows-

$$p(\mathbf{y} \mid \mathbf{X}, \overline{X}, \mathbf{u}) = \prod_{n=1}^{N} p\left(y_n \mid \mathbf{x}_n, \overline{X}, \mathbf{u}\right) = \mathcal{N}\left(\mathbf{y} \mid K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \Lambda\right), \tag{9}$$

where $\Lambda = \operatorname{diag}(\lambda), \lambda_n = K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}} + \sigma^2 I$, $Q_{\mathbf{a},\mathbf{b}} = K_{\mathbf{a},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} K_{\mathbf{u},\mathbf{b}}$

Now, the parameters in this model are the pseudo data and the parameters associated with the covariance matrix. We can learn these parameters by maximizing (9). However, the authors propose placing a Gaussian prior on the pseudo targets and integrating it out of the likelihood in (9).

$$\begin{aligned} p(\mathbf{u}|\mathbf{X_m}) &= \mathcal{N}(\mathbf{u}|0, K_{\mathbf{u},\mathbf{u}}) && (10) \\ \Rightarrow p(\mathbf{y} \mid \mathbf{X}, \overline{X}, \Theta) &= \int p(\mathbf{y} \mid \mathbf{X}, \overline{X}, \mathbf{u}) p(\mathbf{u} \mid \overline{X}) \mathrm{d}\mathbf{u} && (11) \\ &= \mathcal{N}\left(\mathbf{y} \mid \mathbf{0}, Q_{\mathbf{f},\mathbf{f}} + \Lambda\right) && (12) \end{aligned}$$

The likelihood in (12) is maximized with respect to the parameters using gradient ascent. After obtaining the parameters, the prediction distribution is found. First, we find the posterior distribution of pseudo targets $\mathbf{u}$ using Bayes rule on (8) and (10)-

$$p(\mathbf{u} \mid D, \overline{X}) = \mathcal{N}\left(\mathbf{u} \mid K_{\mathbf{u},\mathbf{u}} \Sigma K_{\mathbf{u},\mathbf{f}} \Lambda^{-1} \mathbf{y}, K_{\mathbf{u},\mathbf{u}} \Sigma K_{\mathbf{uu}}\right) \tag{13}$$

where $\Sigma = (K_{\mathbf{uu}} + K_{\mathbf{u},\mathbf{f}} \Lambda^{-1} K_{\mathbf{f},\mathbf{u}})^{-1}$. Then, given a new input $\mathbf{x}_*$, the predictive distribution is obtained by integrating the likelihood (8) with the posterior (13):

$$p\left(y_* \mid \mathbf{x}_*, D, \overline{X}\right) = \int p\left(y_* \mid \mathbf{x}_*, \overline{X}, \mathbf{u}\right) p(\mathbf{u} \mid D, \overline{X}) \mathrm{d}\mathbf{u} = \mathcal{N}\left(y_* \mid \mu_*, \sigma_*^2\right)$$

where

$$\mu_* = K_{*,\mathbf{u}} \Sigma K_{\mathbf{u},\mathbf{f}} \Lambda^{-1} \mathbf{y}$$
$$\sigma_*^2 = K_{*,*} - Q_{*,*} + K_{*,\mathbf{u}} \Sigma K_{\mathbf{u},*} + \sigma^2$$

The authors note that the inversion of the matrix Λ is not a problem because it is a diagonal matrix. However, the computational cost is dominated by $K_{\mathbf{u},\mathbf{f}} \Lambda^{-1} K_{\mathbf{f},\mathbf{u}}$ in the calculation of Σ which is $\mathcal{O}\left(M^2 N\right)$. After various precomputations, prediction can be made in $\mathcal{O}(M)$ for the mean and $\mathcal{O}\left(M^2\right)$ for the variance per test case.

2.1.3 Unifying framework for Sparse Gaussian processes

One of the main approaches identified in the literature to reduce the computational complexity for GPs is to use low rank approximations to the covariance matrix in the model. Deterministic Training Conditions (DTC), Fully Independent Training Conditions (FITC), and Partially Independent Training Conditions are examples of these early approximations. The paper Quiñonero Candela and Rasmussen (2005) discusses these approaches from a unified point of view as methods having an approximate generative model but performing exact inference. The approximate generative model has a sparser covariance matrix imposed by a set of 'inducing points' and the methods seek to modify the joint prior $p(\mathbf{f}, \mathbf{f}_*)$ with the help of these 'inducing points'.

Consider a set of m latent variables $\mathbf{u} = [u_1, ...u_m]^T$, which are referred as 'inducing variables' or 'inducing points' in the literature. These latent variables are values of the function $\mathbf{f}$ corresponding to set of input locations X_u, that is $u = f(X_u)$. Now, we can rewrite the joint prior $p(\mathbf{f}, \mathbf{f}_*)$ as follows -

$$p(\mathbf{f}, \mathbf{f}_*) = \int p(\mathbf{f}, \mathbf{f}_*, \mathbf{u}) d\mathbf{u} = \int p(\mathbf{f}, \mathbf{f}_* | \mathbf{u}) p(\mathbf{u}) d\mathbf{u}, \quad \text{where,} p(\mathbf{u}) = N(\mathbf{0}, \mathbf{K_{uu}}) \tag{14}$$

A fundamental approximation made by all approaches considered in Quiñonero Candela and Rasmussen (2005) is to assume that conditioned on the inducing variables, $\mathbf{f}$ and $\mathbf{f}_*$ are independent of each others ($p(\mathbf{f}, \mathbf{f}_* | \mathbf{u}) = q(\mathbf{f}_* | \mathbf{u}) q(\mathbf{f} | \mathbf{u})$). This approximation leads to a simpler model where dependencies between $\mathbf{f}$ and $\mathbf{f}_*$ are allowed to be captured by $\mathbf{u}$. After use of this approximation, the joint prior becomes

$$p(\mathbf{f}, \mathbf{f}_*) \approx \int q(\mathbf{f}_* | \mathbf{u}) q(\mathbf{f} | \mathbf{u}) p(\mathbf{u}) d\mathbf{u} \tag{15}$$

Now, the approaches in Quiñonero Candela and Rasmussen (2005) take different directions by making extra assumptions about $q(\mathbf{f}_* | \mathbf{u})$ and $q(\mathbf{f} | \mathbf{u})$. However as, $p(\mathbf{u}) = N(\mathbf{0}, \mathbf{K_{uu}})$, we can use properties of Normal distribution to find exact conditional distribution as follows -

$$p(\mathbf{f} | \mathbf{u}) = N(K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}}), \tag{16}$$
$$p(\mathbf{f}_* | \mathbf{u}) = N(K_{*,\mathbf{f}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, K_{*,*} - Q_{*,*}), \tag{17}$$

where $Q_{\mathbf{a},\mathbf{b}} = K_{\mathbf{a},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} K_{\mathbf{u},\mathbf{b}}$

SPGP (Snelson and Ghahramani (2005)) can be shown to be a particular case of this unifying framework developed by Quiñonero Candela and Rasmussen (2005). The authors provided an equivalent formulation called Fully Independent Conditional (FITC) based on the following inducing conditionals-

$$q_{\text{FITC}}(\mathbf{f} \mid \mathbf{u}) = \prod_{i=\text{f}}^{n} p\left(f_i \mid \mathbf{u}\right) = \mathcal{N}\left(K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \operatorname{diag}\left[K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}}\right]\right), \quad \text{and} \quad q_{\text{FTTC}}\left(f_* \mid \mathbf{u}\right) = p\left(f_* \mid \mathbf{u}\right)$$

If we compare (16) to $q_{\text{FITC}}(\mathbf{f} \mid \mathbf{u})$ mentioned above, we can notice that FITC uses the assumption of independence as an approximation to the training conditional distribution of $\mathbf{f}$ given $\mathbf{u}$. The effective prior implied by the FITC is given by

$$q_{\text{FITC}}\ \left(\mathbf{f}, f_*\right) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\mathbf{f},\mathbf{f}} - \operatorname{diag}\left[Q_{\mathbf{f},\mathbf{f}} - K_{\mathbf{f},\mathbf{f}}\right] & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & K_{*,*} \end{bmatrix}\right).$$

The predictive distribution is

$$q_{\text{FITC}}\ \left(f_* \mid \mathbf{y}\right) = \mathcal{N}\left(Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \Lambda\right)^{-1} \mathbf{y}, K_{*,*} - Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \Lambda\right)^{-1} Q_{\mathbf{f},*}\right) \tag{18}$$

$$= \mathcal{N}\left(K_{*,\mathbf{u}} \Sigma K_{\mathbf{u},\mathbf{f}} \Lambda^{-1} \mathbf{y}, K_{*,*} - Q_{*,*} + K_{*,\mathbf{u}} \Sigma K_{\mathbf{u},*}\right) \tag{19}$$

where we have defined $\Sigma = \left(K_{\mathbf{u},\mathbf{u}} + K_{\mathbf{u},\mathbf{f}} \Lambda^{-1} K_{\mathbf{f},\mathbf{u}}\right)^{-1}$ and $\Lambda = \operatorname{diag}\left[K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}} + \sigma_{\text{noise}}^2 I\right]$. The computational complexity is identical to that of SoR.
SoR can also be framed in terms of prior conditional independence of the regression functions of the training and the test sets given the inducing variables. Indeed, in the SoR illustration, we had reparameterized the model as in (6), so that $\mathbf{y} = K_{\mathbf{ff}} \alpha + \varepsilon$, with $\alpha \sim \mathcal{N}(0, K_{\mathbf{ff}}^{-1})$. After assuming $\alpha = P\beta$, we are inducing sparsity, so that many α_i's are almost surely zero. Thus, we can permute the elements of α and the columns of $K_{\mathbf{ff}}$ in such a way that $\mathbf{y} = K_{\mathbf{ff}}^1 \alpha^1 + K_{\mathbf{ff}}^2 \alpha^2 + \varepsilon$, with $\alpha^2 = 0$. Let $K_{\mathbf{uu}} := K_{\mathbf{ff}}^1$. Given the properties of degenerate random variables, $\alpha^1 \sim \mathcal{N}(0, K_{\mathbf{uu}}^{-1})$. Thus, the regression function evaluated at the test input $\mathbf{f}_*$ becomes

$$\mathbf{f}_* = K_{*\mathbf{f}} \alpha^1$$

Noticing that $K_{\mathbf{uu}}^{-1}$ is of full rank, we can define

$$\mathbf{u} := K_{\mathbf{uu}} \alpha^1$$

so that $\mathbf{u} \sim \mathcal{N}(0, K_{\mathbf{uu}})$ and $\alpha^1 = K_{\mathbf{uu}}^{-1} \mathbf{u}$. The regression function at the test value becomes

$$\mathbf{f}_* = K_{*\mathbf{u}} K_{\mathbf{uu}}^{-1} \mathbf{u}$$

The same steps can be followed to derive

$$\mathbf{f} = K_{\mathbf{fu}} K_{\mathbf{uu}}^{-1} \mathbf{u}$$

We have constructed $\mathbf{u}$ in such a way that there is a deterministic relation both between $\mathbf{f}_*$ and $\mathbf{u}$ and between $\mathbf{f}$ and $\mathbf{u}$, which implies conditional independence of $\mathbf{f}$ and $\mathbf{f}_*$ given $\mathbf{u}$, in accordance with the framework outlined above. In particular, we can use degenerate normal distributions to write

$$\begin{aligned} q(\mathbf{f} \mid \mathbf{u}) &= \mathcal{N}\left(K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \mathbf{0}\right) \\ q\left(\mathbf{f}_* \mid \mathbf{u}\right) &= \mathcal{N}\left(K_{*,\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \mathbf{0}\right) \end{aligned}$$

so that we can recover the SoR prior by marginalizing out $\mathbf{u}$

$$q_{\mathrm{SoR}}\left(\mathbf{f}, \mathbf{f}_*\right) = \mathcal{N}\left(\mathbf{0}, \left[\begin{array}{cc} Q_{\mathbf{f},\mathbf{f}} & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & Q_{*,*} \end{array}\right]\right)$$

The predictive distribution can then be computed as

$$q_{\mathrm{SoR}}\left(\mathbf{f}_* \mid \mathbf{y}\right) = \mathcal{N}\left(Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \sigma_{\text{noise}}^2 I\right)^{-1} \mathbf{y}, Q_{*,*} - Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \sigma_{\text{noise}}^2 I\right)^{-1} Q_{\mathbf{f},*}\right)$$

in line with Quiñonero Candela and Rasmussen (2005) framework.

2.2 Variational Gaussian Processes

As shown by Quinonero et al, the sparse approximation methods for GPs can be reconsidered in a framework where the prior is modified by inducing sparsity on it. From (2), we can infer the marginal distribution of $\mathbf{y}$

$$\mathbf{y} \sim \mathcal{N}\left(0, \sigma^2 I + K_{\mathbf{ff}}\right) \tag{20}$$

The different methods we have discussed above all involve the choice of a different, lower rank matrix $Q_{\mathbf{ff}}$ such that

$$\mathbf{y} \sim \mathcal{N}\left(0, \sigma^2 I + Q_{\mathbf{ff}}\right) \tag{21}$$

In Quinonero et al. framework, $Q_{\mathbf{ff}}$ is a function of the inducing variables $\mathbf{u}$. We repeat that $\mathbf{u}$ is a vector of length M that is distributed according to the GP prior, and that corresponds to a matrix of data pseudo inputs X_m. As it's clear from (21), the usage of $Q_{\mathbf{ff}}$ amounts to a modification of the kernel hyperparameters of the GP prior. Choosing $Q_{\mathbf{ff}}$ could be done by optimizing the approximate log marginal distribution (21) over the inducing variables. However, since the low rank matrix $Q_{\mathbf{ff}}$ is derived from an approximate prior, this optimization would not guarantee a good approximation for the exact GP model. To solve this issue, following Titsias (2009), we first write the equivalent augmented model

$$p(\mathbf{y},\mathbf{f},\mathbf{u}) = p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f} \mid \mathbf{u})\,p(\mathbf{u}) \tag{22}$$

We notice that integrating over $\mathbf{u}$ leads to the original model $p(\mathbf{y},\mathbf{f}) = p(\mathbf{y} \mid \mathbf{f})p(\mathbf{f})$, so the two models are actually equivalent. Next, for any set of function points $\mathbf{z}$, we can write

$$p(\mathbf{z}|\mathbf{y}) = \int\int p(\mathbf{z} \mid \mathbf{f},\mathbf{u},\mathbf{y})\,p(\mathbf{f} \mid \mathbf{u},\mathbf{y})\,p(\mathbf{u} \mid \mathbf{y})\,d\mathbf{u}d\mathbf{f}$$

$\mathbf{y}$ is just a noisy version of $\mathbf{f}$, so we have that $p(\mathbf{z} \mid \mathbf{f},\mathbf{u},\mathbf{y}) = p(\mathbf{z} \mid \mathbf{f},\mathbf{u})$. Next, if we assume that the inducing variables $\mathbf{u}$ are sufficient for predicting $\mathbf{z}$, we get conditional independence of $\mathbf{z}$ and $\mathbf{f}$ given $\mathbf{u}$, which leads to $p(\mathbf{z} \mid \mathbf{f},\mathbf{u},\mathbf{y}) = p(\mathbf{z} \mid \mathbf{u})$. Thus, we get

$$p(\mathbf{z}|\mathbf{y}) = \int p(\mathbf{z} \mid \mathbf{u})\,p(\mathbf{u} \mid \mathbf{y})\,d\mathbf{u}$$

If $\mathbf{u}$ is not a sufficient statistics for the model, it is still possible to approximate $p(\mathbf{z}|\mathbf{y})$ using $q(\mathbf{z}) := \int p(\mathbf{z} \mid \mathbf{u})\,\phi(\mathbf{u})\,d\mathbf{u}$, where $\phi(\mathbf{u})$ is a function used to approximate $p(\mathbf{u} \mid \mathbf{y})$. Using the augmented model (22), we can approximate the posterior for $\mathbf{z} = (\mathbf{f},\mathbf{u})$ through $q(\mathbf{f},\mathbf{u})$. Our goal is then to minimize the Kullback leibler (KL) divergence between the two distributions. Using variational inference techniques, such minimization can be equivalently expressed as the maximization of the following lower bound of the true log marginal distribution

$$F_V(X_M,\phi) = \int p(\mathbf{f} \mid \mathbf{u})\,\phi(\mathbf{u})\log\frac{p(\mathbf{y} \mid \mathbf{f})p(\mathbf{u})}{\phi(\mathbf{u})}d\mathbf{f}d\mathbf{u} \tag{23}$$

where X_M is the matrix of pseudo data that induce $\mathbf{u}$. For a given set of inducing inputs (and inducing variables), we can maximize such bound choosing ϕ over the set of M-multivariate normal distributions – as we know that the true posterior is normal. Such maximization leads to the optimal function

$$\phi^*(\mathbf{u}) = \mathcal{N}(\mathbf{u} \mid \mu, A)$$

– where $\mu = \sigma^{-2}K_{\mathbf{uu}}\Sigma K_{\mathbf{uf}}\mathbf{y}$, $A = K_{\mathbf{uu}}\Sigma K_{\mathbf{uu}}$, and $\Sigma = \left(K_{\mathbf{uu}} + \sigma^{-2}K_{\mathbf{uf}}K_{\mathbf{fu}}\right)^{-1}$, – and to the optimized lower bound

$$F_V(X_m) = \log\left[\mathcal{N}\left(\mathbf{y} \mid \mathbf{0}, \sigma^2 I + Q_{\mathbf{ff}}\right)\right] - \frac{1}{2\sigma^2}\operatorname{Tr}(\widetilde{K}) \tag{24}$$

where $Q_{\mathbf{ff}} = K_{\mathbf{fu}}\left[K_{\mathbf{uu}}\right]^{-1}K_{\mathbf{uu}}$ and $\widetilde{K} = K_{\mathbf{ff}} - K_{\mathbf{fu}}\left[K_{\mathbf{uu}}\right]^{-1}K_{\mathbf{uu}}$ for any given vector of inducing variables $\mathbf{u}$. Such optimization requires a computational time of $O(NM^2)$. In turn, we can maximize (24) over the set of inducing inputs X_M, which can be done either through gradient-based optimization, or through a greedy approach proposed by the authors. The computational times of these further optimization steps, however, are dominated. The key insight of the variational approach is the way kernel hyperparameters are learned. In sparse approaches, maximizing the log marginal likelihood after having introduced a set of inducing variables would be inconsistent with the original model, since the prior changes. Though it doesn't lead to an improvement in computational time, the basic variational approach provides a rigorous way to learn the kernel hyperparameters – and also the variance of the noise – allowing for regularization and avoiding arbitrariness and overfitting. Moreover, as we are going to see next, such approach is the building block for a stochastic variational inference formulation that leads to a big improvement in computation time.

2.2.1 Stochastic Variational Inference on GPs (SVGP)

The theoretical improvements that are brought by a variational inference approach to GPs do not lead to computational improvements, as long as an exact optimization is performed. However, if stochastic variational inference can be applied, then a huge improvement is achieved, as computational requirements become $O(M^3)$. In what follows, the notation $\langle\cdot\rangle$ will be used to denote expectation. Following Hensman et al. (2013), we notice that Jensen's inequality implies

$$\begin{aligned}\log p(\mathbf{y} \mid \mathbf{u}) &= \log\langle p(\mathbf{y} \mid \mathbf{f})\rangle_{p(\mathbf{f}|\mathbf{u})}\\ &\geq \langle \log p(\mathbf{y} \mid \mathbf{f})\rangle_{p(\mathbf{f}|\mathbf{u})} \triangleq \mathscr{L}_1\end{aligned}$$

When the likelihood can be factorized, it is also possible to factorize the above bound as follows

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

where $\mu = K_{nm}K_{mm}^{-1}\mathbf{u}$ and $\widetilde{K} = K_{nn} - K_{nm}K_{mm}^{-1}K_{mn}$ are defined as in the previous subsection. We can use $\mathscr{L}_1$ to recover Titsias' variational bound, as in

$$\begin{aligned}\log p(\mathbf{y} \mid \mathbf{X}) &= \log \int p(\mathbf{y} \mid \mathbf{u}) p(\mathbf{u}) \mathrm{d}\mathbf{u}\\ &\geq \log \int \exp\left\{\mathscr{L}_1\right\} p(\mathbf{u}) \mathrm{d}\mathbf{u} \triangleq \mathscr{L}_2\end{aligned}$$

As we noticed in the previous subsection, such bound can be rewritten as

$$\mathscr{L}_2 = \log \mathscr{N}\left(\mathbf{y} \mid \mathbf{0}, K_{nm}K_{mm}^{-1}K_{mn} + \beta^{-1}\mathbf{I}\right) - \frac{1}{2}\beta\,\mathrm{tr}(\widetilde{K})$$

leading to an approximating distribution $q(\mathbf{u})$ with mean and precision

$$\begin{aligned}\hat{\mathbf{u}} &= \beta\Lambda^{-1}K_{mm}^{-1}K_{mn}\mathbf{y}\\ \Lambda &= \beta K_{mm}^{-1}K_{mn}K_{nm}K_{mm}^{-1} + K_{mm}^{-1}\end{aligned}$$

where we highlight precision in the parameterization. The authors derive a third bound as follows

$$\begin{aligned}\log p(\mathbf{y}) &= \log \int p(\mathbf{y} \mid \mathbf{u}) q(\mathbf{u}) d\mathbf{u} \\ &\geq \log \langle \exp(\mathscr{L}_1) \rangle_{q(\mathbf{u})} \\ &\geq \langle \log [\exp(\mathscr{L}_1)] \rangle_{q(\mathbf{u})} \\ &\geq \langle \mathscr{L}_1 - \mathrm{KL}(q(\mathbf{u}) \| p(\mathbf{u})) \rangle_{q(\mathbf{u})} \triangleq \mathscr{L}_3\end{aligned}$$

where we have used the previously derived inequality, Jensen's inequality, and the fact that the Kullback-Leibler divergence is always positive. Using the insight from the previous section, we guess $q(\mathbf{u}) \sim \mathscr{N}(\mathbf{m}, \mathbf{S})$, so that we maximize over the set of multivariate normal distributions. Moreover, using the fact that $\mathscr{L}_1$ can be factorized, it is possible to show that

$$\mathscr{L}_3 = \sum_{i=1}^{n} \left\{ \log \mathscr{N}\left(y_i \mid \mathbf{k}_i^\top K_{mm}^{-1} \mathbf{m}, \beta^{-1} \right) - \frac{1}{2} \beta \widetilde{k}_{i,i} - \frac{1}{2} \operatorname{tr}(\mathbf{S} \Lambda_i) \right\} - \mathrm{KL}(q(\mathbf{u}) \| p(\mathbf{u}))$$

where $\mathbf{k}_i$ are the columns of K_{mn} and $\Lambda_i = \beta K_{mm}^{-1} \mathbf{k}_i \mathbf{k}_i^\top K_{mm}^{-1}$. Next, we can minimize the Kullback Leibler divergence by maximizing $\mathscr{L}_3$ with respect to the parameters of $\mathbf{q}(\mathbf{u})$. The derivatives w.r.t. $\mathbf{m}$ and $\mathbf{S}$ are

$$\begin{aligned}\frac{\partial \mathscr{L}_3}{\partial \mathbf{m}} &= \beta K_{mm}^{-1} K_{mn} \mathbf{y} - \Lambda \mathbf{m} \\ \frac{\partial \mathscr{L}_3}{\partial \mathbf{S}} &= \frac{1}{2} \mathbf{S}^{-1} - \frac{1}{2} \Lambda\end{aligned}$$

with optimal solutions $\mathbf{S} = \Lambda^{-1}$, $\mathbf{m} = \hat{\mathbf{u}}$. This implies $\mathscr{L}_3 \leq \mathscr{L}_2$, but also implies that equality holds at the optimal value for $\mathscr{L}_3$. The important difference between the two bounds, however, is that $\mathscr{L}_3$ is expressed as a sum of terms, and so it can be maximized by applying stochastic gradient methods on $\mathbf{q}(\mathbf{u})$. Importantly, we can reparameterize $\mathbf{q}(\mathbf{u})$ so that it belongs to the natural exponential family. This leads to the canonical and expectational parameters

$$\begin{aligned}\theta_1 = \mathbf{S}^{-1} \mathbf{m}, \quad \theta_2 = -\frac{1}{2} \mathbf{S}^{-1} \\ \eta_1 = \mathbf{m}, \quad \eta_2 = \mathbf{m}\mathbf{m}^\top + \mathbf{S}\end{aligned}$$

which allows to write the natural gradient as

$$\widetilde{\mathbf{g}}(\theta) = G(\theta)^{-1} \frac{\partial \mathscr{L}_3}{\partial \theta} = \frac{\partial \mathscr{L}_3}{\partial \eta}$$

In turns, this leads to the updates

$$
\begin{aligned}
\theta_{2(t+1)} &= -\frac{1}{2}\mathbf{S}_{(t+1)}^{-1} \\
&= -\frac{1}{2}\mathbf{S}_{(t)}^{-1} + \ell\left(-\frac{1}{2}\Lambda + \frac{1}{2}\mathbf{S}_{(t)}^{-1}\right) \\
\theta_{1(t+1)} &= \mathbf{S}_{(t+1)}^{-1}\mathbf{m}_{(t+1)} \\
&= \mathbf{S}_{(t)}^{-1}\mathbf{m}_{(t)} + \ell\left(\beta K_{mm}^{-1}K_{mn}\mathbf{y} - \mathbf{S}_{(t)}^{-1}\mathbf{m}_{(t)}\right)
\end{aligned}
$$

so that taking a step of unit length allows to recover the optimal solution for $\mathscr{L}_3$. Finally, the fact that the objective function is expressed as a sum provides the key improvement, since we can stochastically approximate the natural gradient by taking the data in mini-batches. This eliminates the dependency on N for the complexity, so that the inversion of $K_{m,m}$ dominates the algorithm, and the complexity becomes $O(M^3)$.

2.3 Variational Nearest Neighbor Gaussian Processes

Stochastic Variational GPs (SVGP) discussed in the last section form a low rank approximation to the prior covariance matrix. These approximations have been found to preform well in various situations. However, there are also many instances where the idea of low rank approximation doesn't work very well. Variational Nearest Neighbor Gaussian Processes (VNNGP) proposed by Wu et al. (2022) improve upon SVGP by using sparse approximation to the prior precision matrix. The sparse precision matrix assumes that observations are conditionally independent given their closest neighbors. This lets the joint GP prior be approximated by the product of one-dimensional conditionals, each of which only depends on a small subset of ”nearby” observations.

2.3.1 Nearest Neighbor Gaussian Processes (NNGP)

Nearest Neighbor Gaussian Processes (NNGPs) Datta et al. (2016) are sparse and fast approximations for Gaussian process (GP) models. They introduce sparisty in the covariance matrix using specified neighbor sets.

Consider any finite dimensional realization $\mathbf{f} = \{f_i\}_{i=1}^N$ of a Gaussian process at locations $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N \in \mathbb{R}^D$:

$$
\begin{gathered}
f(\cdot) \sim GP(0, K(\cdot,\cdot)) \\
f_i \equiv f(\mathbf{x}_i) \quad \text{for } i = 1:N
\end{gathered}
$$

where $K(\cdot,\cdot)$ is a kernel function that encodes the properties of the prior. We can formulate $p(\mathbf{f})$ as follows:

$$
\begin{aligned}
p(\mathbf{f}) &= p(f_1, f_2 \dots, f_n) && (25)\\
&= p(f_1).p(f_2|f_1).p(f_3|f_1, f_2)\dots f_n|f_1, f_2, \dots f_n)) && (26)
\end{aligned}
$$

$$p(\mathbf{f}) = p(f_1)\prod_{i=2}^{N} p(f_i \mid \mathbf{f}_{1:i-1}), \text{ where } p(f_1) = \mathcal{N}(f_1 \mid 0, K_{1,1}) \text{ and} \tag{27}$$

$$p(f_i \mid \mathbf{f}_{1:i-1}) = \mathcal{N}\left(f_i \mid K_{1:i-1,i}^{\top} K_{1:i-1,1:i-1}^{-1} \mathbf{f}_{1:i-1}, K_{i,i} - K_{1:i-1,i}^{\top} K_{1:i-1,1:i-1}^{-1} K_{1:i-1,i}\right) \tag{28}$$

and we define $K_{i,i} \equiv K(\mathbf{x}_i, \mathbf{x}_i), K_{1:i-1,i} \equiv K(\mathbf{x}_{1:i-1}, \mathbf{x}_i)$, and $K_{1:i-1,1:i-1} \equiv K\left(\mathbf{x}_{1:i-1,i}, \mathbf{x}_{1:(i-1)}\right)$.

Consider an observation at a specific point $\mathbf{x}_i$ and some observations at a distant point $\mathbf{x}_j$. In many situations, it might be safe to think that f_i doesn't have much of an effect on f_j and vice versa. We can also think about other similar observations that are far enough away to have little effect on $\mathbf{x}_i$. Then, we can ignore conditioning based on these observations in the density, $p(f_i|\mathbf{f}_{1:i-1})$.

Consider conditioning only on K samples, where $K <<< N$. These K samples are referred to as ”neighbors” of $\mathbf{x}_i$ and are denoted as $n(i)$. We will have the following approximation to (27)-

$$p(\mathbf{f}) \approx p(f_1)\prod_{i=2}^{N} p\left(f_i \mid \mathbf{f}_{n(i)}\right), \quad \text{where} \quad p(f_1) = \mathcal{N}(f_1 \mid 0, K_{1,1}), \tag{29}$$

$$p\left(f_i \mid \mathbf{f}_{n(i)}\right) = \mathcal{N}\left(f_i \mid K_{n(i),i}^{\top} K_{n(i),n(i)}^{-1} \mathbf{f}_{n(i)}, K_{i,i} - K_{n(i),i}^{\top} K_{n(i),n(i)}^{-1} K_{n(i),i}\right)$$

and $n(i)$ denotes the indices of K nearest neighbors of $\mathbf{x}_i$ in $\{\mathbf{x}_k\}_{k=1}^{i-1}$. This approximation results in a sparse Cholesky factor of the precision matrix. As a consequence, the model complexity scales $O\left(NK^3\right)$. Vecchia (1988) proposed specifying $n(i)$ to be K nearest neighbours of $\mathbf{x}_i$ among $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{i-1}$ with respect to Euclidean distance.

2.3.2 Variational Inference

The stochastic variational Gaussian process (SVGP) (Hensman et al. (2013)) defines a small set of M inducing $\mathbf{u} = \{u_j\}_{j=1}^{M}$ GP latent variables at locations $\left\{\mathbf{X}_{u_j}\right\}_{j=1}^{M}$. It considers the joint latent generative process:

$$p(\mathbf{u}, \mathbf{f}) = \mathcal{N}\left(\begin{pmatrix} \mathbf{u} \\ \mathbf{f} \end{pmatrix} \mid 0, \begin{pmatrix} K_{\mathbf{u},\mathbf{u}} & K_{\mathbf{u},\mathbf{f}} \\ K_{\mathbf{f},\mathbf{u}} & K_{\mathbf{f},\mathbf{f}} \end{pmatrix}\right)$$

and an independent observation model

$$p(\mathbf{y} \mid \mathbf{f}) = \prod_{i=1}^{N} p(y_i \mid f_i) \tag{30}$$

where $p(y_i \mid f_i)$ denotes any likelihood function.

Variational Nearest Neighbor GP (VNNGP) considers the same observation model in (30) and makes use of the NNGP approximation (29) to the latent generative process-

$$p(\mathbf{u}) \approx \prod_{j=1}^{M} p\left(u_i \mid \mathbf{u}_{n(j)}\right), \quad p(\mathbf{f} \mid \mathbf{u}) \approx \prod_{i=1}^{N} p\left(f_i \mid \mathbf{u}_{n(i)}\right) \tag{31}$$

where $n(j)$ denotes the inducing point indices corresponding to $\mathbf{X}_{\mathbf{u_j}}$'s (at most) K nearest neighbors chosen from $\left\{\mathbf{X}_{\mathbf{u_1}}, \cdots, \mathbf{X}_{\mathbf{u_{j-1}}}\right\}$.

Due to the NNGP approximation, VNNGP has a sparse approximation the prior precision matrix $K_{\mathbf{u},\mathbf{u}}^{-1}$. The cholesky factor of the precision matrix has at most K+1 non-zero elements per row, corresponding to the K-nearest-neighbor structure. The authors note that the nearest neighbors approximation in (31) is dependent on a particular ordering of the inducing points. But after a huge number of tests, they found that orderings don't make a big difference in how well the VNNGP works.

Optimization The authors consider mean-field variational approximation for **u** and approximate the posterior for **f** with the nearest neighbor prior:

$$q(\mathbf{u}) = \prod_{j=1}^{M} q\left(u_j\right) = \prod_{j=1}^{M} \mathcal{N}\left(u_j \mid m_j, s_j\right) \tag{32}$$

$$q(\mathbf{f}) = \prod_{i=1}^{N} q\left(f_i\right) = \prod_{i=1}^{N} \int p\left(f_i \mid \mathbf{u}_{n(i)}\right) q\left(\mathbf{u}_{n(i)}\right) \mathrm{d}\mathbf{u}_{n(i)} \tag{33}$$

Now, the goal is to maximize the VNNGP's ELBO as follows

$$\mathscr{L}_{\text{VNNGP}} = \sum_{i=1}^{N} \underset{q(f_i)}{\mathbb{E}} \left[\log p\left(y_i \mid f_i\right)\right] - \mathrm{KL}[q(\mathbf{u}) \| p(\mathbf{u})] \tag{34}$$

The first term in the equation is the data likelihood while the second term is the KL divergence. Now, let us focus first on the data likelihood term. Similar to SVGP, the expression for likelihood can be factorize over data points due to (30). However, unlike SVGP which considers all M inducing pons for making prediction for each data point, VNNGP only uses K nearest inducing points due to its structure. Therefore, the data likelihood terms in (34) becomes:

$$\sum_{i=1}^{N} \int p\left(f_i \mid \mathbf{u}_{n(i)}\right) q\left(\mathbf{u}_{n(i)}\right) \log p\left(y_i \mid f_i\right) d f_i d\mathbf{u}_{n(i)} \tag{35}$$

For KL divergence, the authors show that it can be written as a summation over inducing points. They also note that doing the same is not possible for SVGP as SVGP's KL requires accessing parameters for all inducing points at the same time.

$$\mathrm{KL}[q(\mathbf{u})\|p(\mathbf{u})] = \mathrm{KL}\left[\prod_{j=1}^{M} q\left(u_j\right) \| \prod_{j=1}^{M} p\left(u_j \mid \mathbf{u}_{n(j)}\right)\right] \tag{36}$$

$$= \underset{q(\mathbf{u})}{\mathbb{E}}\left[\sum_{j=1}^{M} \log \frac{q\left(u_j\right)}{p\left(u_j \mid \mathbf{u}_{n(j)}\right)}\right] \tag{37}$$

$$= \sum_{j=1}^{M} \underset{q\left(\mathbf{u}_{n(j)}\right)}{\mathbb{E}}\left[\mathbb{E}_{q\left(u_j\right)}\left[\log \frac{q\left(u_j\right)}{p\left(u_j \mid \mathbf{u}_{n(j)}\right)}\right]\right] \tag{38}$$

$$= \sum_{j=1}^{M} \underset{q\left(\mathbf{u}_{n(j)}\right)}{\mathbb{E}}\left[\mathrm{KL}\left[q\left(u_j\right) \| p\left(u_j \mid \mathbf{u}_{n(j)}\right)\right]\right] \tag{39}$$

where

$$p\left(u_j \mid \mathbf{u}_{n(j)}\right) = \mathscr{N}\left(u_j \mid K_{n(j),j}^{\top} K_{n(j),n(j)}^{-1} \mathbf{u}_{n(j)}, K_{j,j} - K_{n(j),j}^{\top} K_{n(j),n(j)}^{-1} K_{n(j),j}\right).$$

Now, it can be seen that ELBO in (34) has been simplified and has a summation structure due to (35) and (39). Stochastic Optimization can be used for optimizing the ELBO. The authors propose randomly sampling a mini-batch of training data indices $\mathscr{I} = \{i_k\}_{k=1}^{N_b}$ and a mini-batch of inducing point indices $\mathscr{J} = \{j_l\}_{l=1}^{M_b}$. Then the unbiased estimate of the ELBO can be optimized as follows

$$\mathscr{L}_{\mathrm{VNNGP}} \approx \frac{N}{N_b} \sum_{i \in \mathscr{I}} \underset{q(f_i)}{\mathbb{E}}\left[\log p\left(y_i \mid f_i\right)\right] \tag{40}$$

$$-\frac{M}{M_b} \sum_{j \in \mathscr{J}} \underset{q\left(\mathbf{u}_{n(j)}\right)}{\mathbb{E}}\left[\mathrm{KL}\left[q\left(u_j\right) \| p\left(u_j \mid \mathbf{u}_{n(j)}\right)\right]\right]. \tag{41}$$

The exact computation of the ELBO is omitted here for brevity. The interested reader is directed to Appendix A of Wu et al. (2022). The authors note that optimizing (41) requires $O((N_b + M_b)K^3)$ computations. The main computational load comes from finding nearest neighbor structures for observations and inducing points.

3 Conclusion

Gaussian Processes are a very powerful tool for nonparametric inference. Their intrinsic cubic computational time prevents exact inference when dealing with large datasets, but many approaches have been proposed to tackle such issue. Sparse methods all share the idea of approximating the variance covariance matrix with a low rank matrix that is easier to invert. We have seen how this can be done by directly selecting only a subset of the columns of the covariance matrix and setting the others to zero (SoR, Smola and Bartlett (2000), $O(NM^2)$). Another approach is to assume that the Gaussian Process realizations are conditionally independent given a set of pseudo inputs (SPGP, Snelson and Ghahramani (2005), $O(NM^2)$. All sparse methods lead to a restriction on the GP prior, which is constrained in such a way that the variance covariance matrix is not of full rank. The kernel hyperparameters and the variance of the noise can then be learned maximizing the likelihood. This approach, however, leads to fitting a modified model, with few theoretical guarantees about the original model. A variational inference (VI) approach allows to regularize the model and avoid overfitting by expliciting optimizing a lower bound on the marginal distribution of the data ($O(NM^2)$). If the bound can be factorized, we can even perform stochastic variational inference (SVI) with great improvements in terms of computational time (SVGP, Hensman et al. (2013), $O(M^3)$). Recently, Variational NNGP (VNNGP, Wu et al. (2022), $O(K^3)$) has recently been presented as an improvement to SVGP. VNNGP approximates the prior precision matrix with a sparse approximation as opposed to a low rank approximation, resulting in a considerable reduction in time complexity. There are also other recent approaches for improving scalability of GPs which are not discussed in this report. Hamelijnck et al. (2021) , for example, derive a sparse approximation that constructs a state-space model over a reduced number of spatial inducing points, and demonstrate that it recovers the conventional variational GP while exhibiting favorable computational properties. In future, we can investigate such approaches for improving scalbility of GPs.

No.	Method	Remarks	Computation Time
1	SoR	Low rank approximation to covariance matrix	$O(NM^2)$
2	SPGP	Low rank approximation to covariance matrix	$O(NM^2)$
3	Variational GP	VI for inference in GP	$O(NM^2)$
4	Stochastic Variational GP	Stochastic VI for inference in GP	$O(M^3)$
5	Variational NNGP	Stochastic VI and sparse approximation to covariance matrix	$O(K^3)$

Table 1: Comparison of computation time of different methods, where $K \ll M \ll N$. (VI: Variational Inference)

4 Acknowledgements

We take this opportunity to heartily thank our instructor Prof. Nhat Ho for teaching the course and allowing us to work on this project. His suggestions were enriching and helped us work on our project.

References

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016). Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812. PMID: 29720777.

Hamelijnck, O., Wilkinson, W. J., Loppi, N. A., Solin, A., and Damoulas, T. (2021). Spatio-temporal variational gaussian processes. *Advances in Neural Information Processing Systems*, 34:23621–23633.

Hensman, J., Fusi, N., and Lawrence, N. (2013). Gaussian processes for big data. *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 282–290.

Quiñonero Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959.

Smola, A. and Bartlett, P. (2000). Sparse greedy gaussian process regression. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

Snelson, E. and Ghahramani, Z. (2005). Sparse gaussian processes using pseudo-inputs. *Advances in neural information processing systems*, 18.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In *Artificial intelligence and statistics*, pages 567–574. PMLR.

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):297–312.

Wu, L., Pleiss, G., and Cunningham, J. P. (2022). Variational nearest neighbor Gaussian process. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24114–24130. PMLR.