Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. TR-162

Sparse Multiscale Gaussian Process Regression

Christian Walder, Kwang In Kim,
Bernhard Schölkopf

August 2007

Department Schölkopf, email: first.last@tuebingen.mpg.de

Sparse Multiscale Gaussian Process Regression

Christian Walder, Kwang In Kim, Bernhard Schölkopf

Abstract. Most existing sparse Gaussian process (g.p.) models seek computational advantages by basing their computations on a set of m basis functions that are the covariance function of the g.p. with one of its two inputs fixed. We generalise this for the case of Gaussian covariance function, by basing our computations on m Gaussian basis functions with arbitrary diagonal covariance matrices (or length scales). For a fixed number of basis functions and any given criteria, this additional flexibility permits approximations no worse and typically better than was previously possible. Although we focus on g.p. regression, the central idea is applicable to all kernel based algorithms, such as the support vector machine. We perform gradient based optimisation of the marginal likelihood, which costs $O(m^2 n)$ time where n is the number of data points, and compare the method to various other sparse g.p. methods. Our approach outperforms the other methods, particularly for the case of very few basis functions, *i.e.* a very high sparsity ratio.

1 Introduction

The Gaussian process (g.p.) is a popular non-parametric model for supervised learning problems. Although g.p.'s have been shown to perform well on a wide range of tasks, their usefulness is severely limited by the $O(n^3)$ time and $O(n^2)$ storage requirements where n is the number of data points. A large amount of work has been done to alleviate this problem, either by approximating the posterior distribution, or constructing degenerate covariance functions for which the exact posterior is less expensive to evaluate [1, 2, 3, 4, 5] — for a unifying overview see [6]. The majority of such methods achieve an $O(m^2 n)$ time complexity for training where $m \ll n$ is the number of points on which the computations are based.

Due to the non parametric nature of the g.p. there are potentially as many parameters to estimate as there are training points. An exception is the case where the covariance function has finite rank, such as the linear covariance function on $\mathbb{R}^d \times \mathbb{R}^d$ given by $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}'$, which has rank d. In this case the g.p. collapses to a parametric method and it is possible to derive algorithms with $O(d^2 n)$ time complexity by basing the computations on d basis functions.

For non-degenerate covariance functions $k(\cdot, \cdot)$, the various existing sparse g.p. algorithms are quite different to one another, but all have in common that they base their computations on m basis functions of the form $k(\boldsymbol{z}_i, \cdot)$. Typically the set $\mathcal{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_m\}$ is taken to be a subset of the training set [1, 2, 4]. For example Seeger *et al.* [4] employ a highly efficient approximate information gain criteria to incrementally select points from the training set in a greedy manner.

More recently Snelson and Ghahramani [5] have shown that further improvements in the quality of the model for a given m can be made — especially for small m — by removing the restriction that $\mathcal{Z}$ be a subset of the training set. For this they introduced a new sparse g.p. model which has the advantage of being closer to the full g.p., and also of being more amenable to gradient based optimisation of the marginal likelihood with respect to the set $\mathcal{Z}$. A further advantage of their continuous optimisation of $\mathcal{Z}$ is that the hyper parameters of the model can be optimised at the same time — this is more difficult when $\mathcal{Z}$ is taken to be a subset of the training set, since choosing such a subset is a non-differentiable combinatoric problem.

In this paper we take a logical step forward in the development of sparse g.p. algorithms. We also base our computations on a finite set of basis functions, but remove the restriction that the basis functions be of the form $k(\boldsymbol{z}_i, \cdot)$ where k is the covariance of the g.p. This requires computing integrals involving the basis and covariance functions, and so cannot always be done in closed form. Fortunately however, closed form expressions can be attained for arguably the most useful scenario, namely that of Gaussian covariance function (with arbitrary diagonal covariance matrix) along with Gaussian basis functions (again each with their own arbitrary diagonal covariance matrix).

The central idea is that under some mild restrictions we can compute the likelihood — under the g.p. model with Gaussian covariance — of arbitrary Gaussian mixtures. Although our analysis is new, there is some precedent for

it in the literature. In particular, Walder *et al.* [7] employ a similar idea, but from a reproducing kernel Hilbert space (r.k.h.s.) rather than a g.p. perspective, and for a different basis and regulariser (namely B_3-spline and thin-plate, respectively). Also related is the work of Franz *et al.* [8], who perform their analysis from a g.p. perspective with arbitrary basis and covariance function, but with the important difference that they do not take infinite limits.

The work has a direct analogy in r.k.h.s. theory. Indeed the central idea can be applied to any kernel machine, but in the present paper we focus on the g.p. framework. The main reason for this is that it allows us to build on Snelson and Ghahramani's sparse g.p. model [5], which has already been shown to be amenable to gradient based optimisation of the marginal likelihood.

The paper is structured as follows. Section 2 provides an introduction to g.p. regression models. In Section 3 we derive the likelihood of arbitrary Gaussian mixtures under the g.p. model with Gaussian covariance, and clarify the link to r.k.h.s.'s. In Section 4 we discuss and motivate the precise probabilistic model which we use to take advantage of our theoretical results. Experimental results and conclusions are presented in Sections 5 and 6, respectively.

2 Gaussian Process Regression

Given an independent and identically distributed (i.i.d.) sample

$$\mathcal{S} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\} \subset \mathbb{R}^d \times \mathbb{R}$$

drawn from an unknown distribution, the goal is to estimate $P(y|\boldsymbol{x})$. We introduce a latent variable $u \in \mathbb{R}$, and make the assumption that $P(y|u, \boldsymbol{x}) = P(y|u)$. Hence we can think of y as a noisy realisation of u, which we model by $P(y|u) = \mathcal{N}(y|u, \sigma_n^2)$ where σ_n is a hyper parameter.[1]

The relationship $\boldsymbol{x} \to u$ is a random process $u(\cdot)$, namely a zero mean g.p. with covariance function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Typically k will be defined in terms of further hyper parameters. We shall denote such a g.p. as $\mathcal{G}(k)$, which is defined by the fact that its joint evaluation at a finite number of input points is a zero mean Gaussian random variable with covariance

$$\mathbb{E}_{f \sim \mathcal{G}(k)} \left[f(\boldsymbol{x}) f(\boldsymbol{z}) \right] = k(\boldsymbol{x}, \boldsymbol{z}).$$

One can show that conditioned upon the hyper parameters the posterior $P(\boldsymbol{u}|\mathcal{S})$, where[2] $[\boldsymbol{u}]_i = u(\boldsymbol{x}_i)$, is given by

$$\begin{aligned} P(\boldsymbol{u}|\mathcal{S}) &\propto P(\boldsymbol{u}) \mathcal{N}(\boldsymbol{y}|\boldsymbol{u}, \sigma_n^2 I) \\ &\propto \mathcal{N}\left(\boldsymbol{u}|K_{xx}(K_{xx} + \sigma_n^2 I)^{-1} y, \sigma_n^2 K_{xx}(K_{xx} + \sigma_n^2 I)^{-1}\right), \end{aligned} \tag{1}$$

where $[K_{xx}]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Note that for simplicity we have, like many authors, neglected to notate the conditioning upon the hyper parameters, as we shall do throughout the paper. Now, it can furthermore be shown that the latent function $u_* = u(\boldsymbol{x}_*)$ at an arbitrary test point $\boldsymbol{x}_*$ is distributed according to $P(u_*|\boldsymbol{x}_*, \mathcal{S}) = \int P(u_*|\boldsymbol{u}) P(\boldsymbol{u}|\mathcal{S}) \, \mathrm{d}\boldsymbol{u} = \mathcal{N}(u_*|\mu_*, \sigma_*^2)$, where

$$\mu_* = \boldsymbol{y}^\top (K_{xx} + \sigma_n^2 I)^{-1} \boldsymbol{k}_*, \tag{2}$$

$$\sigma_*^2 = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^\top (K_{xx} + \sigma_n^2 I)^{-1} \boldsymbol{k}_*, \tag{3}$$

and we have defined $[\boldsymbol{k}_*]_i = k(\boldsymbol{x}_*, \boldsymbol{x}_i)$.

In a Bayesian setting, one places priors over the hyper parameters and computes the hyper posterior, but this typically necessitates computationally expensive numerical integration techniques. Alternatively one may fix the hyper parameters to those obtained by maximising some criteria such as the marginal likelihood conditioned upon them, $P(\boldsymbol{y}|\mathcal{X}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, K_y)$, where $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ and $K_{yy} = K_{xx} + \sigma_n^2 I$ is the covariance matrix for $\boldsymbol{y}$. This can be computed using the result that

$$\log\left(P(\boldsymbol{y}|\mathcal{X})\right) \propto -\boldsymbol{y}^\top K_{yy}^{-1} \boldsymbol{y} - \log|K_{yy}| + c, \tag{4}$$

where c is a term independent of the hyper parameters. Even when one neglects the cost of choosing the hyper parameters however, it typically costs $O(n)$ and $O(n^2)$ time to evaluate the posterior mean and variance respectively, after an initial setup cost of $O(n^3)$.

[1] We adopt the common convention of writing $\mathcal{N}(x|\mu, \sigma)$ for the probability density at x of the Gaussian random variable with mean μ and variance σ.

[2] Square brackets with subscripts denote elements of matrices and vectors, and a colon subscript denotes an entire row or column of a matrix.

3 Sparse Multiscale Gaussian Process Regression

Let u be drawn from $\mathcal{G}(k)$. As we mentioned previously, this means that the vector of joint evaluations at an arbitrary ordered set of points $\mathcal{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ is a random variable, call it $\boldsymbol{u}_\mathcal{X}$, distributed according to

$$P_{\boldsymbol{u}_\mathcal{X}}(\boldsymbol{u}) \propto \mathcal{N}(\boldsymbol{u}|\boldsymbol{0}, K_{xx}). \tag{5}$$

Hence the probability density function (p.d.f.) of this random variable is

$$f_{\boldsymbol{u}_\mathcal{X}}(\textstyle\sum_{i=1}^m c_i \boldsymbol{u}_i) = \left|2\pi K_{xx}^{-1}\right|^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}\textstyle\sum_{i,j=1}^m c_i c_j \boldsymbol{u}_i^\top K_{xx}^{-1} \boldsymbol{u}_j\right),$$

where $|\cdot|$ denotes the matrix determinant. Note that this is simply the p.d.f. of $\boldsymbol{u}_\mathcal{X}$ where we have set the argument to be $\sum_{i=1}^m c_i \boldsymbol{u}_i$, for some $c_i \in \mathbb{R}$. We have done this because later we will wish to determine the likelihood of a function expressed as a summation of fixed basis functions. To this end we now consider an infinite limit of the above case. Taking the limit $n \to \infty$ of uniformly distributed points[3] $\boldsymbol{x}_i$ leads to the following p.d.f. for $\mathcal{G}(k)$,

$$f_{\mathcal{G}(k)}(\textstyle\sum_{i=1}^m c_i u_i) = \left|2\pi k^{-1}\right|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\sum_{i,j=1}^m c_i c_j \underbrace{\int\int u_i(\boldsymbol{x}) u_j(\boldsymbol{y}) k^{-1}(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}}_{\triangleq \Psi_k(u_i,u_j)}\right). \tag{6}$$

We will discuss the factor of $\left|2\pi k^{-1}\right|^{-\frac{1}{2}}$ shortly. Note that in the previous case of finite n, if we let $\boldsymbol{u} = K_{xx}\boldsymbol{\alpha}$ and assume that K_{xx} is invertible, then $\boldsymbol{\alpha} = K_{xx}^{-1}\boldsymbol{u}$. Following this finite analogy, by k^{-1} we now intend a sloppy notation for the function which, for $u = \int \alpha(\boldsymbol{x}) k(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x}$, satisfies $\int u(\boldsymbol{x}) k^{-1}(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x} = \alpha(\cdot)$. Hence if we define

$$M_k : \alpha \mapsto M_k\alpha = \int \alpha(\boldsymbol{x}) k(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x},$$

then k^{-1} is by definition the Green's function [9] of M_k, as it satisfies

$$\int \big(M_k\alpha\big)(\boldsymbol{x}) k^{-1}(\boldsymbol{x}, \cdot)\,\mathrm{d}\boldsymbol{x} = \alpha(\cdot). \tag{7}$$

Let us now consider now the covariance function $k(\boldsymbol{x}, \boldsymbol{y}) = cg(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma})$, where $c > 0$, $\boldsymbol{\sigma} > \boldsymbol{0} \in \mathbb{R}^d$ and g is a normalised Gaussian on $\mathbb{R}^d \times \mathbb{R}^d$ with diagonal covariance matrix, that is[4]

$$g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) \triangleq |2\pi \mathrm{diag}\,(\boldsymbol{\sigma})|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^\top \mathrm{diag}\,(\boldsymbol{\sigma})^{-1}(\boldsymbol{x}-\boldsymbol{y})\right). \tag{8}$$

If we assume furthermore that our function is an arbitrary mixture of such Gaussians, so that

$$u_i(\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i), \tag{9}$$

then the well known integral (for the convolution of two Gaussians)

$$\int g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i) g(\boldsymbol{x}, \boldsymbol{v}_j, \boldsymbol{\sigma}_j)\,\mathrm{d}\boldsymbol{x} = g(\boldsymbol{v}_i, \boldsymbol{v}_j, \boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j) \tag{10}$$

leads to

$$\left(\frac{1}{c} M_{cg(\cdot,\cdot,\boldsymbol{\sigma})} g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i - \boldsymbol{\sigma})\right)(\boldsymbol{x}) = g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i) = u_i(\boldsymbol{x}). \tag{11}$$

[3] Although any non-vanishing distribution leads to the same result.

[4] We use diag in a sloppy fashion — for $\boldsymbol{a} \in \mathbb{R}^n$, $\mathrm{diag}(\boldsymbol{a}) \in \mathbb{R}^{n\times n}$ is a diagonal matrix satisfying $[\mathrm{diag}(\boldsymbol{a})]_{ii} = [\boldsymbol{a}]_i$. But for $A \in \mathbb{R}^{n\times n}$, $\mathrm{diag}(A) \in \mathbb{R}^n$ is a column vector with $[\mathrm{diag}(A)]_i = [A]_{ii}$

Hence we can derive the closed form expression

$$\begin{aligned}\Psi_k(u_i,u_j) &\overset{(6,9,11)}{=} \int\int g(\boldsymbol{x},\boldsymbol{v}_i,\boldsymbol{\sigma}_i)\left(\frac{1}{c}M_{cg(\cdot,\cdot,\boldsymbol{\sigma})}g(\cdot,\boldsymbol{v}_j,\boldsymbol{\sigma}_j-\boldsymbol{\sigma})\right)(\boldsymbol{y})k^{-1}(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\boldsymbol{y}\\ &\overset{(7)}{=} \frac{1}{c}\int g(\boldsymbol{x},\boldsymbol{v}_i,\boldsymbol{\sigma}_i)g(\boldsymbol{x},\boldsymbol{v}_j,\boldsymbol{\sigma}_j-\boldsymbol{\sigma})\,\mathrm{d}\boldsymbol{x}\\ &\overset{(10)}{=} \frac{1}{c}\,g(\boldsymbol{v}_i,\boldsymbol{v}_j,\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j-\boldsymbol{\sigma}).\end{aligned}$$

For clarity we have noted above each equals sign the number of the equation which implies the corresponding logical step. The following expression summarises the main idea of the present section

$$P_{\mathcal{G}(0,cg(\cdot,\cdot,\boldsymbol{\sigma}))}\left(\sum_{i=1}^m c_i g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i)\right)\propto\exp\left(-\frac{1}{2}\sum_{i,j=1}^m\frac{1}{c}c_ic_jg(\boldsymbol{v}_i,\boldsymbol{v}_j,\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j-\boldsymbol{\sigma})\right). \tag{12}$$

We give only an unnormalised form by neglecting the factor $\left|2\pi k^{-1}\right|^{-\frac{1}{2}}$ in (6). The neglected factor is equal to the inverse of the integral of the right hand side of the above expression with respect to all functions $\sum_{i=1}^m c_i g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i)$. We need not concern ourselves with choosing a measure with respect to which this integral is finite, due to the fact that, since we will be working only with ratios of the above likelihood (*i.e.* for *maximum a posteriori* (m.a.p.) estimation and marginal likelihood maximisation), we need only the unnormalised form. Note that this peculiarity is not particular to our proposed sparse approximation to the g.p., but is a property of g.p.'s in general.

Let us now make a few remarks regarding the expression (12).

1. If $\boldsymbol{\sigma}_1=\boldsymbol{\sigma}_2=\cdots=\boldsymbol{\sigma}_n=\boldsymbol{\sigma}$ and we reparameterise $c_i=cc_i'$ then it simplifies to (5).
2. Let $c=1$ and $h(\boldsymbol{x})=\exp\left(-\frac{1}{2}\boldsymbol{x}^\top\mathrm{diag}\left(\boldsymbol{\sigma}_1\right)\boldsymbol{x}\right)$, an unnormalised Gaussian. Using (8) and (12) we can derive the log-likelihood of h under the g.p. prior,

$$\log\left(P_{\mathcal{G}(0,g(\cdot,\cdot,\boldsymbol{\sigma}))}\big(h(\cdot)\big)\right)\propto-\sqrt{\frac{|\mathrm{diag}\left(\boldsymbol{\sigma}_1\right)|}{|\mathrm{diag}\left(2\boldsymbol{\sigma}_1-\boldsymbol{\sigma}\right)|}}. \tag{13}$$

   Simple analysis of this expression shows that the most likely such function h is that with $\boldsymbol{\sigma}_1=\boldsymbol{\sigma}$. From this extremal point, as any component of $\boldsymbol{\sigma}_1$ increases, the log likelihood of h decreases without bound. Similarly decreasing any component of $\boldsymbol{\sigma}_1$ also decreases the log likelihood, and as any component of $\boldsymbol{\sigma}_1$ approaches half the value of the corresponding component of $\boldsymbol{\sigma}$, then the log likelihood decreases without bound. To be more precise, we have for all $j=1,2,\ldots,d$ that

$$\lim_{[\boldsymbol{\sigma}_1]_j\to\left(\frac{1}{2}[\boldsymbol{\sigma}]_j\right)^+}\log\left(P_{\mathcal{G}(0,g(\cdot,\cdot,\boldsymbol{\sigma}))}\big(h(\cdot)\big)\right)=-\infty.$$

An interesting consequence of the second point is that, roughly speaking, it is not possible to recover a Gaussian function using a g.p. with Gaussian covariance, if the covariance function is more than twice as broad as the function to be recovered. Although this may at first appear to contradict proven consistency results for the Gaussian covariance function (for example [10]), this is not the case. On the contrary, such results hold only for compact domains, and our analysis is for $\mathbb{R}^d$.

Before proceeding, note that (12) has a direct analogy in the theory of r.k.h.s.'s, as made clear by the following lemma. The lemma follows from (12) and the well understood relationship between every g.p. and the corresponding r.k.h.s. of functions.

Lemma 3.1. *Let $\mathcal{H}$ be the r.k.h.s. with reproducing kernel $g(\cdot,\cdot,\boldsymbol{\sigma})$. If the conditions $\boldsymbol{\sigma}_i>\frac{1}{2}\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_j>\frac{1}{2}\boldsymbol{\sigma}$ are satisfied component-wise, then*

$$\langle g(\cdot,\boldsymbol{v}_i,\boldsymbol{\sigma}_i),g(\cdot,\boldsymbol{v}_j,\boldsymbol{\sigma}_j)\rangle_{\mathcal{H}}=g(\boldsymbol{v}_i,\boldsymbol{v}_j,\boldsymbol{\sigma}_i+\boldsymbol{\sigma}_j-\boldsymbol{\sigma}). \tag{14}$$

If either condition is not satisfied, then the corresponding function on the left hand side is not in $\mathcal{H}$.

Naturally this can also be proven directly, but doing so for the general case is more involved (see Appendices A and B). However, by assuming that the conditions $\boldsymbol{\sigma}_i > \boldsymbol{\sigma}$ and $\boldsymbol{\sigma}_j > \boldsymbol{\sigma}$ are satisfied component-wise, then it is straightforward to attain the main result. The basic idea is as follows. Using (10) we substitute $g(\cdot, \boldsymbol{v}_p, \boldsymbol{\sigma}_p) = \int g(\cdot, \boldsymbol{x}_p, \boldsymbol{\sigma}) g(\boldsymbol{x}_p, \boldsymbol{v}_p, \boldsymbol{\sigma}_p - \boldsymbol{\sigma}) \, \mathrm{d}\boldsymbol{x}_p$ for $p = i, j$ into the l.h.s. of (14). By linearity we can write the two integrals outside the inner product. Next we use the r.k.h.s. reproducing property — the fact that $\langle f(\cdot), g(\cdot, \boldsymbol{x}, \boldsymbol{\sigma}) \rangle_{\mathcal{H}} = f(\boldsymbol{x}), \forall f \in \mathcal{H}, \boldsymbol{x} \in \mathbb{R}^d$ — to evaluate the inner product. Using (10) we integrate to obtain the r.h.s. of (14).

4 Inference with the Sparse Model

4.1 A Simple Approach

The g.p. likelihood over the restricted function space defines a distribution over functions of the form $\sum_{i=1}^{m} c_i g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$ where g as given previously is deterministic and the c_i are, by inspection of (12), normally distributed according to

$$\boldsymbol{c} \sim \mathcal{N}\left(\boldsymbol{0}, U_{\Psi}^{-1}\right), \tag{15}$$

where $[U_{\Psi}]_{i,j} = \Psi_k(u_i, u_j)$. Let us write $\mathcal{U} = \{u_1, \ldots, u_m\}$ (which we refer to as the basis) and refer to the random process thus defined as $\mathcal{G}_{\mathcal{U}}(k)$. This new random process is equivalent to a full g.p. with covariance function of rank at most m given by

$$\begin{aligned} \mathbb{E}_{f \sim \mathcal{G}_{\mathcal{U}}(k)} \left[f(\boldsymbol{x}) f(\boldsymbol{z}) \right] &= \mathbb{E}_{\boldsymbol{c} \sim \mathcal{N}\left(\boldsymbol{0}, U_{\Psi}^{-1}\right)} \left[\left(\boldsymbol{u}_{vx}^{\top} \boldsymbol{c} \right) \left(\boldsymbol{u}_{vz}^{\top} \boldsymbol{c} \right)^{\top} \right] \\ &= \boldsymbol{u}_{vx}^{\top} U_{\Psi}^{-1} \boldsymbol{u}_{vz}, \end{aligned} \tag{16}$$

where $[\boldsymbol{u}_{vx}]_i = g(\boldsymbol{x}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$ and $[\boldsymbol{u}_{vz}]_i = g(\boldsymbol{z}, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$.

As an aside, note that if we choose as the basis $\mathcal{U} = \{g(\cdot, \boldsymbol{x}, \boldsymbol{\sigma}), g(\cdot, \boldsymbol{z}, \boldsymbol{\sigma})\}$, then it is easy to verify using (16) that $\mathbb{E}_{f \sim \mathcal{G}_{\mathcal{U}}(0, g(\cdot, \cdot, \boldsymbol{\sigma}))} \left[f(\boldsymbol{x}) f(\boldsymbol{z}) \right] = \mathbb{E}_{f \sim \mathcal{G}(k)} \left[f(\boldsymbol{x}) f(\boldsymbol{z}) \right]$. This is analogous to a special case of the representer theorem from the theory of r.k.h.s.'s, and agrees with the interpretation that (15) is such that $\mathcal{G}_{\mathcal{U}}(k)$ approximates $\mathcal{G}(k)$ well in some sense, for the given basis $\mathcal{U}$.

Returning to the main thread, the new posterior can be derived as it was at the end of Section 2 for the exact g.p., but using the new covariance function (16). Hence after some algebra we have from (2) and (3) that, conditioned again upon the hyper parameters, the latent function $u_* = u(\boldsymbol{x}_*)$ at an arbitrary test point is distributed according to $P_{u \sim \mathcal{G}_{\mathcal{U}}(k)}(u_* | \boldsymbol{x}_*, \mathcal{S}) = \mathcal{N}(u_* | \mu_*, \sigma_*^2)$, where

$$\mu_* = \left(U_{vx} \boldsymbol{y} \right)^{\top} \left(U_{vx} U_{vx}^{\top} + \sigma_n^2 U_{\Psi} \right)^{-1} \boldsymbol{u}_{v*}, \tag{17}$$

$$\sigma_*^2 = \sigma_n^2 \boldsymbol{u}_{v*}^{\top} \left(U_{vx} U_{vx}^{\top} + \sigma_n^2 U_{\Psi} \right)^{-1} \boldsymbol{u}_{v*}, \tag{18}$$

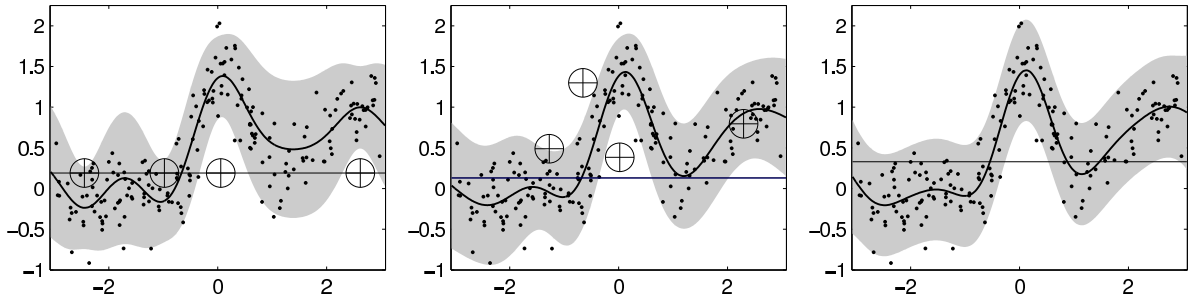
and we have defined $[U_{vx}]_{i,j} = g(\boldsymbol{x}_j, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$, *etc.* Note that these expressions can be evaluated in $O(m)$ and $O(m^2)$ time respectively, after an initial setup or training cost of $O(m^2 n)$. This is the usual improvement over the full g.p. attained by such sparse approximation schemes. It turns out however that by employing an idea introduced by Snelson and Ghahramani [5], we can retain this improvement while switching to a different model that is closer to the full g.p.

4.2 Inference with Improved Variance

A fair criticism of the previous model, evident in (17) and (18), is that the predictive variance approaches zero far away from the basis function centres $\boldsymbol{v}_i$. It turns out that this is particularly problematic to gradient based methods for choosing the basis (the $\boldsymbol{v}_i$ and $\boldsymbol{\sigma}_i$) by maximising the marginal likelihood [5]. An effective but still computationally attractive way of healing the model is to switch to a different g.p. — which we denote $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$ — whose covariance function satisfies

$$\mathbb{E}_{\tilde{\mathcal{G}}_{\mathcal{U}}(k)} \left[f(\boldsymbol{x}) f(\boldsymbol{z}) \right] = \delta_{\boldsymbol{x},\boldsymbol{z}} k(\boldsymbol{x}, \boldsymbol{z}) + (1 - \delta_{\boldsymbol{x},\boldsymbol{z}}) \boldsymbol{u}_{vx}^{\top} U_{\Psi}^{-1} \boldsymbol{u}_{vz}, \tag{19}$$

where δ is the Kronecker delta function. The process defined in this way is referred to the sparse pseudo-input Gaussian process (s.p.g.p.)Ṅote that if $\boldsymbol{x} = \boldsymbol{z}$ then the covariance is that of the original g.p. $\mathcal{G}(k)$, otherwise it is that of $\mathcal{G}_{\mathcal{U}}(k)$. Unlike (16), the prior variance in this case is the same as that of the full g.p., even though in general the covariance is not. Once again the posterior can be found as before by replacing the covariance



(a) *spgp-full* (b) *vsgp-full* (c) exact g.p.

Figure 1: Predictive distributions (mean curve with $\pm$ two standard deviations shaded). For the *spgp-full* and *vsgp-full* algorithms, we plot the $(\boldsymbol{v}_i, \boldsymbol{\sigma}_i) \in \mathbb{R} \times \mathbb{R}$ of the basis as crossed circles. The horizontal lines denote the resulting $\boldsymbol{\sigma} \in \mathbb{R}$ of the covariance function $cg(\cdot, \cdot, \boldsymbol{\sigma})$.

function in (2) and (3) with the right hand side of (19). In this case we attain after some algebra the expression $P_{u \sim \tilde{\mathcal{G}}_{\mathcal{U}}(k)}(u_* | \boldsymbol{x}_*, \mathcal{S}) = \mathcal{N}(u_* | \mu_*, \sigma_*^2)$ where

$$\mu_* = \boldsymbol{u}_{v*}^\top Q^{-1} U_{vx} \left(\Lambda + \sigma_n^2 I\right)^{-1} \boldsymbol{y} \tag{20}$$

$$\sigma_*^2 = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{u}_{v*}^\top \left(U_\Psi^{-1} - Q^{-1}\right) \boldsymbol{u}_{v*}, \tag{21}$$

$\Lambda = \text{diag}\,(\boldsymbol{\lambda})$, and

$$[\boldsymbol{\lambda}]_i = k(\boldsymbol{v}_i, \boldsymbol{v}_i) - [U_{vv}]_{:,i}^\top U_\Psi^{-1} [U_{vv}]_{:,i}$$

$$Q = U_\Psi + U_{vx} \left(\Lambda + \sigma_n^2 I\right)^{-1} U_{vx}^\top.$$

To compute the marginal likelihood we can use (4), but it is most efficiently computed using Cholesky decompositions. The derivation of the gradients of the marginal likelihood with respect to the various parameters is long and tedious. Our derivation, which closely follows [4], can be found in Appendix C. Note that by factorising appropriately, all of the required gradients can be attained in $O(m^2 n + mnd)$.

5 Experiments

Our main goal is to demonstrate the value of being able to vary the $\boldsymbol{\sigma}_i$ individually. Let us clarify the terminology we use to refer to the various algorithms under comparison. Our new method is the variable sigma Gaussian process (v.s.g.p.). The *vsgp-full* variant consists of optimising the marginal likelihood with respect to the m basis centers $\boldsymbol{v}_i \in \mathbb{R}^d$ and length scales $\boldsymbol{\sigma}_i \in \mathbb{R}^d$ of our basis functions $u_i = g(\cdot, \boldsymbol{v}_i, \boldsymbol{\sigma}_i)$ where g is defined in (8). Also optimised are the following hyper parameters — the noise variance $\sigma_n \in \mathbb{R}$ of (1), and the parameters $c \in \mathbb{R}$ and $\boldsymbol{\sigma} \in \mathbb{R}^d$ of our original covariance function $cg(\cdot, \cdot, \boldsymbol{\sigma})$. The *vsgp-basis* variant is identical to *vsgp-full* except that σ_n, c and $\boldsymbol{\sigma}$ are determined by optimising the marginal likelihood of a full g.p. trained on a subset of the training data, and then held fixed while the $\boldsymbol{\sigma}_i$ and $\boldsymbol{v}_i$ are optimised as before. Both v.s.g.p. variants use the $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$ probabilistic model of Section 4.2, where $k = cg(\cdot, \cdot, \boldsymbol{\sigma})$.

spgp-full and *spgp-basis* correspond to the work of Snelson and Ghahramani [5], and are identical to their v.s.g.p. counterparts except that — as with all sparse g.p. methods prior to the present work — they are forced to satisfy the constraints $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}, i = 1 \ldots m$. To initialise the marginal likelihood optimisation we take the $\boldsymbol{v}_i$ to be a random subset of the training data. The other parameters are always initialised to the same sensible starting values, which is reasonable due to the preprocessing we employ (which is identical to that of [4]) in order to standardise the data sets.

Figure 1 demonstrates the basic idea on a one dimensional toy problem. Using $m = 4$ basis functions is not enough for *spgp-full* to accurately infer a posterior similar to that of the full g.p. trained on the depicted $n = 200$ training points. The v.s.g.p. achieves a posterior closer to that of the full g.p. by employing — in comparison to the full g.p. — larger $\boldsymbol{\sigma}_i$'s and a smaller $\boldsymbol{\sigma}$, leading to an effective covariance function — that of $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$ as given by (16) — which better matches that of the full g.p. depicted in Figure 1 (c). In addition to merely observing the

similarity between Figures 1 (b) and (c), we verified this last statement directly by visualising $\mathbb{E}_{\tilde{\mathcal{G}}_{\mathcal{U}}(k)}\left[f(\boldsymbol{x})f(\boldsymbol{z})\right]$ of (19) as a function of $\boldsymbol{x}$ and $\boldsymbol{z}$, but we omit the plot due to space limitations.

Figure 2 shows our experiments which, following [4] and [5], were performed on the larger *pumadyn-32nm* and *kin-40k* data sets[5]. For the optimisation of the s.p.g.p. and v.s.g.p. methods we used a standard conjugate gradient type optimiser. Optimising the v.s.g.p. methods from a random initialisation tended to lead to inferior local optima, so we used the s.p.g.p. to find a starting point for the optimisation. This is possible because both methods optimise the same criteria, while the s.p.g.p. merely searches a subset of the space permitted by the v.s.g.p. framework. To ensure a fair comparison, we optimised the s.p.g.p. for 4000 iterations, whereas for the v.s.g.p. we optimised first the s.p.g.p. for 2000 iterations (*i.e.* fixing $\boldsymbol{\sigma}_i = \boldsymbol{\sigma}, i = 1 \dots m$), took the result as a starting point, and optimised the v.s.g.p. for a further 2000 iterations (with the $\boldsymbol{\sigma}_i$ unconstrained).

We have also reproduced with kind permission the results of Seeger *et al.* [4], and hence have used exactly the experimental methodology described therein. The results we reproduce are from the *info-gain* and *smo-bart* methods. *info-gain* is their own method which is extremely cheap to train for a given set of hyper parameters. The method uses greedy subset selection based on a criteria which can be evaluated efficiently. *smo-bart* is similar but is based on a criteria which is more expensive to compute [1]. We also show the result of training a full g.p. on a subset of the data of size 2000 and 1024 for *kin-40k* and *pumadyn-32nm*, respectively.

Neither *info-gain* nor *smo-bart* estimate the hyper-parameters, but rather fix them to the values determined by the optimisation of marginal likelihood for the full g.p. Hence they are most directly comparable to *spgp-basis* and *vsgp-basis*. However, *spgp-full* and *vsgp-full* correspond to the more difficult task of estimating the hyper parameters at the same time as the basis.

For *pumadyn-32nm* we do not plot *spgp-basis* and *vsgp-basis* as the results are practically identical to *spgp-full* and *vsgp-full*. This differs from [5], where local minima problems with *spgp-full* on the *pumadyn-32nm* data set are explicitly reported. It is unclear why our experiments did not suffer in this way — possible explanations are the choice of initial starting point, as well as the specific optimisation algorithm employed. The results of the s.p.g.p. and v.s.g.p. methods on the *pumadyn-32nm* data set very similar, but both outperform the *info-gain* and *smo-bart* approaches.

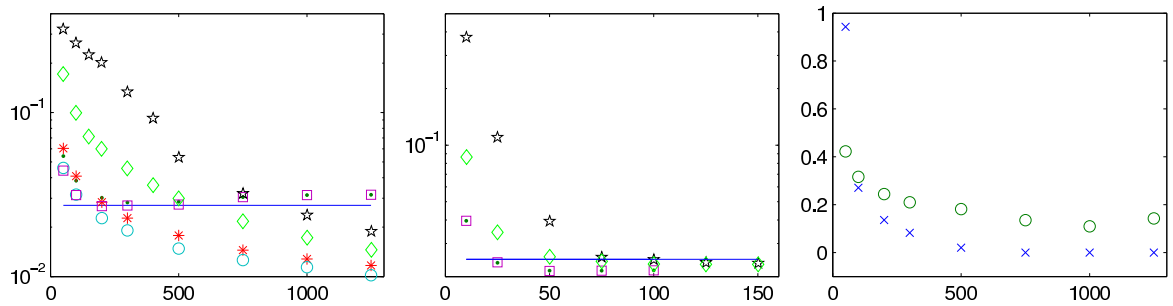
The *kin-40k* results are rather different. While the $\boldsymbol{\sigma}_i$ deviated little from $\boldsymbol{\sigma}$ on the *pumadyn-32nm* data set, this was not the case for *kin-40k*, particularly for small m, as seen in Figure 2 (c) where we plot $\frac{1}{md}\sum_{i=1}^{m}\sum_{j=1}^{d}([\boldsymbol{\sigma}_i - \boldsymbol{\sigma}]_j)^2$. Our results are in agreement with those of [5] — our *vsgp-full* outperforms *spgp-full* for small m, which in turn outperforms both *info-gain* and *smo-bart*. However for large m both *spgp-full* and *vsgp-full* tend to over fit. This is to be expected due to the use of marginal likelihood optimisation, as the choice of basis $\mathcal{U}$ is equivalent to the choice of the order of md hyper parameters for the covariance function of $\tilde{\mathcal{G}}_{\mathcal{U}}(k)$. Happily, and somewhat surprisingly, the *vsgp-full* method tends not to over fit more than the *spgp-full*, in spite of it's having roughly twice as many basis parameters. Neither *vsgp-basis* nor *spgp-basis* suffered from over fitting however, and while they both outperform *info-gain* and *smo-bart*, our *vsgp-basis* clearly demonstrates the advantage of our new s.p.g.p. framework by consistently outperforming *spgp-basis*.

6 Conclusions

Sparse g.p. regression is an important topic which has received a lot of attention in recent years. Previous methods have based their computations on subsets of the data or pseudo input points. To relate this to our method, this is analogous to basing the computations on a set of basis functions of the form $k(\boldsymbol{v}_i, \cdot)$ where k is the covariance function and the $\boldsymbol{v}_i$ are for example the pseudo input points. We have generalised this for the case of Gaussian covariance function, by basing our computations on a set of Gaussian basis functions whose bandwidth parameters may vary independently.

This provides a new avenue for approximations which is applicable to all kernel based algorithms, including for example the g.p. and the support vector machine. To demonstrate the utility of this new degree of freedom, we have constructed a sparse g.p. regression algorithm which outperforms previous methods, particularly for very sparse solutions.

[5] *kin-40k*: 10000 training, 30000 test, 9 attributes, see www.igi.tugraz.at/aschwaig/data.html.
pumadyn-32nm: 7168 training, 1024 test, 33 attributes, see www.cs.toronto/ delve.



(a) *kin-40k* data set (b) *pumadyn-32nm* data set (c) Deviation of $\boldsymbol{\sigma}_i$ on *kin-40k*

Figure 2: Plots (a) and (b) depict the test error as a function of basis size m for the full g.p. (blue line), *spgp-full* (green dot), *spgp-basis* (red asterisk), *vsgp-full* (cyan square), *vsgp-basis* (blue circle), *info-gain* (black pentagram) and *smo-bart* (green diamond). Plot (c) depicts m vs. the variation in the $\boldsymbol{\sigma}_i$ (see text) for *vsgp-full* (green circles) and *vsgp-basis* (blue crosses).

References

[1] Alex J. Smola and Peter L. Bartlett. Sparse greedy gaussian process regression. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 619–625. MIT Press, Cambridge, MA, 2000.

[2] Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Comp.*, 14(3):641–668, 2002.

[3] Neil Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse gaussian process methods: The informative vector machine. In Suzanna Becker, Sebastian Thrun, and Klaus Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 609–616, 2002.

[4] Matthias Seeger, Chris Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse gaussian process regression. In Christopher M. Bishop and Brendan J. Frey, editors, *Workshop on AI and Statistics 9*. Society for Artificial Intelligence and Statistics, 2003.

[5] Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1257–1264. MIT Press, Cambridge, MA, 2006.

[6] J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1935–1959, 12 2005.

[7] Christian Walder, Bernhard Schölkopf, and Olivier Chapelle. Implicit surface modelling with a globally regularised basis of compact support. *Proc. EUROGRAPHICS*, 25(3):635–644, 2006.

[8] M. O. Franz and P. V. Gehler. How to choose the covariance for gaussian process regression independently of the basis. In *Proc. Gaussian Processes in Practice Workshop*, Bletchley Park, UK, 2006.

[9] G. F. Roach. *Green's Functions*. Cambridge University Press, Cambridge, UK, 1970.

[10] Ingo Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93, 2002.

[11] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. Technical Report UCB/CSD-01-1166, EECS Department, University of California, Berkeley, Nov 2001.

[12] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950.

[13] C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, Cambridge, Massachusetts, 01 2006.

[14] Izrail S. Gradshtein and Iosif Ryzhik. *Table of Integrals, Series, and Products*. Academic Press, New York, corrected and enlarged edition, 2nd printing edition, 1965, 1980.

[15] C. T. Pan. A modification to the linpack downdating algorithm. *BIT Numerical Mathematics*, 30(4):707–722, 1990.

A Proof of Lemma 3.1

We need to a) prove which (dilated) Gaussian functions lie in the r.k.h.s. of a Gaussian, and b) compute the norm of those which do. The inner product then follows from the parallelogram identity. Note that the result corresponding to a) is mentioned in [11] (in the proof of their Theorem 2), based on a characterisation of the r.k.h.s. in terms of Fourier transforms — for a sketch of this simple approach see Appendix B. Presently we prove both a) and b) in a direct manner using the following

Theorem A.1 (Aronzajn [12])**.** *The function f belongs to the r.k.h.s. $\mathcal{H}$ with reproducing kernel (r.k.) k if and only if there exists an $\epsilon > 0$ such that*

$$R_\epsilon(\boldsymbol{x}, \boldsymbol{y}) = k(\boldsymbol{x}, \boldsymbol{y}) - \epsilon f(\boldsymbol{x}) f(\boldsymbol{y}),$$

is positive definite (p.d.), in which case

$$\|f\|_{\mathcal{H}}^2 = \inf \{1/\epsilon : R_\epsilon \textit{ is p.d.}\}.$$

Let

$$p(x) = \exp(-2ax^2)$$

and

$$k(x, y) = \exp(-b(x - y)^2).$$

It turns out that we can write ([13], *Chapter 4.3, Eigenfunction Analysis of Kernels*)

$$k(x, y) = \sum_{m=0}^{\infty} \lambda_m \phi_m(x) \phi_m(y),$$

where

$$\begin{aligned} \lambda_m &= \sqrt{\frac{2c}{A}} \frac{B^m}{2^m m!} \neq \sqrt{\frac{2a}{A}} B^m, \\ \phi_m(x) &= \exp(-(c-a)x^2) H_m(\sqrt{2c}\, x), \end{aligned}$$

and

$$c = \sqrt{a^2 + 2ab}, \quad A = a + b + c, \quad B = b/A.$$

Note that what follows the $\neq$ for λ_m was given in [13], but is incorrect. The functions $\{\phi_i\}$ are orthogonal with respect to the Lebesgue measure, with

$$\begin{aligned} \int_{\mathbb{R}} \phi_m(x) \phi_n(x) \, \mathrm{d}p(x) &= \int_{\mathbb{R}} \exp(-2cx^2) H_m(\sqrt{2c}\, x) H_n(\sqrt{2c}\, x) \, \mathrm{d}x \\ &= 1/\sqrt{2c} \int_{\mathbb{R}} \exp(-x^2) H_m(x) H_n(x) \, \mathrm{d}x \\ &= \sqrt{\frac{\pi}{2c}}\, 2^m m!\, \delta_{m,n}. \end{aligned}$$

Here we made a change of variables $\int_{\mathbb{R}} f(x) \, \mathrm{d}x = a \int_{\mathbb{R}} f(ax) \, \mathrm{d}x$, and used (7.374.1) from Gradshteyn and Ryzhik [14], namely

$$\int_{\mathbb{R}} \exp(-x^2) H_m(x) H_n(x) \, \mathrm{d}x = \sqrt{\pi}\, 2^m m!\, \delta_{ij}.$$

Since the basis functions are not normalised, we have the following relation

$$s = \sum_{m=0}^{\infty} \zeta_m \phi_m \Rightarrow \langle s | p | \phi_n \rangle = r_n^2 \zeta_n.$$

Hence can write $f(x) = \exp(-x^2) = \sum_{i=0}^{\infty} \gamma_i \phi_i(x)$ where

$$\begin{aligned} r_{2m}^2 \gamma_{2m} &= \int_{\mathbb{R}} f(x)\phi_{2m}(x)\,\mathrm{d}p(x) \\ &= \int_{\mathbb{R}} \exp(-x^2)\exp(-(c-a)x^2) H_{2m}(\sqrt{2c}\,x)\exp(-2ax^2)\,\mathrm{d}x \\ &= 1/\sqrt{F}\int_{\mathbb{R}} \exp(-x^2) H_{2m}\left(\sqrt{2c/F}\,x\right)\,\mathrm{d}x \\ &= \sqrt{\frac{\pi}{F}}\,\frac{(2m)!}{m!}\,(2c/F-1)^m\,. \end{aligned}$$

where $F = c + a + 1$. To evaluate this integral we used (7.373.2) from [14], namely

$$\int_{\mathbb{R}} \exp(-x^2) H_{2m}(xy)\,\mathrm{d}x = \sqrt{\pi}\,\frac{(2m)!}{m!}(y^2-1)^m,$$

from which we also have $\gamma_{2m+1} = 0, \forall m \in \mathbb{N}_0$. This leads to

$$\gamma_{2m} = \sqrt{\frac{\pi}{F}}\,\frac{(2m)!}{m!}\,(2c/F-1)^m \,/\left(\sqrt{\frac{\pi}{2c}}\,2^{2m}(2m)!\right) = \sqrt{\frac{2c}{F}}\,\frac{(2c/F-1)^m}{2^{2m}m!},$$

and so we have

$$\begin{aligned} R_\epsilon(x,y) &= k(x,y) - \epsilon f(x) f(y) \\ &= \sum_{i=0}^{\infty} \lambda_i \phi_i(x)\phi_i(y) - \epsilon \sum_{j=0}^{\infty} \gamma_j \phi_j(x) \sum_{k=0}^{\infty} \gamma_k \phi_k(y). \end{aligned}$$

Letting $g = \sum_{p=0}^{\infty} \xi_p \phi_p$ be arbitrary, to satisfy Mercer's condition we require that $\mathcal{I} \geq 0$ where

$$\begin{aligned} \mathcal{I} &\triangleq \int\int g(x)g(y)R_\epsilon(x,y)\,\mathrm{d}p(x)\,\mathrm{d}p(y) \\ &= \int\int \sum_{p=0}^{\infty} \xi_p \phi_p(x) \sum_{q=0}^{\infty} \xi_q \phi_q(y) \\ &\qquad \cdot \left(\sum_{i=0}^{\infty} \lambda_i \phi_i(x)\phi_i(y) - \epsilon \sum_{j=0}^{\infty} \gamma_j \phi_j(x) \sum_{k=0}^{\infty} \gamma_k \phi_k(y)\right)\,\mathrm{d}p(x)\,\mathrm{d}p(y) \\ &= \sum_{p=0}^{\infty} \xi_p^2 r_p^4 \lambda_p - \epsilon \sum_{p=0}^{\infty} \xi_p r_p^2 \gamma_p \sum_{q=0}^{\infty} \xi_q r_q^2 \gamma_q. \end{aligned}$$

This can be written $\boldsymbol{\xi}^\top M \boldsymbol{\xi}$ where $M = N - \epsilon \boldsymbol{p}^\top \boldsymbol{p}$ is an infinite dimensional diagonal matrix minus a rank one matrix. Since $\lambda_k > 0$ for all k, we can use the fact that (*Observation 1, [15]*)

$$\left(N - \epsilon \boldsymbol{p}\boldsymbol{p}^\top \succeq 0\right) \Leftrightarrow \left(\boldsymbol{p}^\top N^{-1} \boldsymbol{p} \leq 1/\epsilon\right),$$

and hence we equivalently require

$$1/\epsilon \geq \sum_{k=0}^{\infty} \gamma_k^2/\lambda_k = \frac{b}{\sqrt{2b-1}}.$$

The above equality can be shown using (in addition to various tedious algebraic manipulations) formula (0.241.3) from [14], namely $\sum_{m=1}^{\infty} p^m (2m)!/(m!)^2 = 1/\sqrt{1-4p}$. Note that the final sentence in Lemma 3.1 follows from Theorem A.1 and the fact that for $b \leq 1/2$ there is no $\epsilon > 0$ satisfying the above inequality.

It also follows from Theorem A.1 that $\|f\|^2_{\mathcal{H}(k)} = b/\sqrt{2b-1}$, where we denote by $\mathcal{H}(k)$ the r.k.h.s. with r.k. k. To compare this to Lemma 3.1, note that

$$f = \sqrt{\pi}\,g(0,\cdot,1/2), \quad \text{and} \quad k(x,y) = \sqrt{\frac{\pi}{b}}\,g(x,y,1/(2b)).$$

Using Lemma 3.1 and $\langle f, g\rangle_{\mathcal{H}(k)} = c\,\langle f, g\rangle_{\mathcal{H}(ck)}$, one can verify with a little algebra that

$$\left\langle \sqrt{\pi}\, g(0,\cdot,1/2), \sqrt{\pi}\, g(0,\cdot,1/2)\right\rangle_{\mathcal{H}(\sqrt{\frac{\pi}{b}}\, g(x,y,1/(2b)))} = \frac{b}{\sqrt{2b-1}}.$$

Finally, it is straightforward to generalise the result to the Gaussian kernel on $\mathbb{R}^d$, since this is merely the product of d univariate Gaussian kernels.

B A Sketch of the Fourier Transform Based Proof of Lemma 3.1

Assuming $k : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, by letting

$$k(x,y) = \phi(x-y) = \int_{\mathbb{R}} \Phi(w) \exp(-\imath(x-y)w)\, \mathrm{d}w,$$

we can verify that

$$\left\langle \int_{\mathbb{R}} F(w) \exp(-\imath x w)\, \mathrm{d}w, \int_{\mathbb{R}} G(w) \exp(-\imath x w)\, \mathrm{d}w \right\rangle_{\mathcal{H}(k)} = \int_{\mathbb{R}} \frac{F(w)G(w)}{\Phi(w)}\, \mathrm{d}w,$$

because the reproducing property holds, *i.e.*

$$\begin{aligned}
\langle f(x), k(x,y)\rangle_{\mathcal{H}(k)} &= \left\langle \int_{\mathbb{R}} F(w) \exp(-\imath x w)\, \mathrm{d}w, \int_{\mathbb{R}} \Phi(w) \exp(-\imath x w) \exp(\imath y w)\, \mathrm{d}w \right\rangle_{\mathcal{H}(k)} \\
&= \int_{\mathbb{R}} \frac{F(w)\Phi(w)\exp(\imath y w)}{\Phi(w)}\, \mathrm{d}w \\
&= f(y).
\end{aligned}$$

The result follows by substituting the known Fourier transforms for the Gaussian, and requiring a finite norm.

C Implementation Details

As we employ standard gradient based optimisation tools, we need only derive the partial derivatives of the objective function (the marginal likelihood) with respect to the various parameters. Here we provide all of the key expressions.

Derivatives of g

Recall that

$$g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) \triangleq \frac{1}{\sqrt{(2\pi)^d \left|\mathrm{diag}\left(\boldsymbol{\sigma}\right)\right|}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{y}\right)^\top \mathrm{diag}\left(\boldsymbol{\sigma}\right)^{-1}\left(\boldsymbol{x}-\boldsymbol{y}\right)\right).$$

The partial derivatives are then

$$\begin{aligned}
\frac{\partial}{\partial \left[\boldsymbol{x}\right]_i} g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) &= \frac{\left(\left[\boldsymbol{y}\right]_i - \left[\boldsymbol{x}\right]_i\right)}{\left[\boldsymbol{\sigma}\right]_i} g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) \\
\frac{\partial}{\partial \left[\boldsymbol{\sigma}\right]_i} g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}) &= \frac{1}{2}\left(\frac{\left(\left[\boldsymbol{y}\right]_i - \left[\boldsymbol{x}\right]_i\right)^2}{\left[\boldsymbol{\sigma}\right]_i^2} - \frac{1}{\left[\boldsymbol{\sigma}\right]_i}\right) g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\sigma}).
\end{aligned}$$

Marginal Likelihood

The marginal likelihood is

$$P(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{0}, U_{vx}^\top U_\Psi^{-1} U_{vx} + \Lambda + \sigma_n^2 I),$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ and

$$\lambda_i = \left(U_{xx}\right)_{i,i} - \left(U_{vx}\right)_{i,:}^\top U_\Psi^{-1} \left(U_{vx}\right)_{i,:}.$$

Writing the negative log marginal likelihood as $\phi = -\log P(\boldsymbol{y})$, we then have

$$\begin{aligned}\phi &= \frac{1}{2}\left(d\log(2\pi) + \log|U_{vx}^\top U_\Psi^{-1}U_{vx} + \Lambda + \sigma_n^2 I)|\right. \\ &\left. + \boldsymbol{y}^\top\left(U_{vx}^\top U_\Psi^{-1}U_{vx} + \Lambda + \sigma_n^2 I\right)^{-1}\boldsymbol{y}\right)\end{aligned}$$

where d is the dimensionality of the input space. Removing the constant term $\frac{1}{2}d\log(2\pi)$ and defining $\Gamma = \sigma_n^{-2}\Lambda + I$ we obtain the following new cost functional

$$\phi = \frac{1}{2}\left(\log\left|\sigma_n^2(\Gamma + \sigma_n^{-2}U_{vx}^\top U_\Psi^{-1}U_{vx})\right| + \sigma_n^{-2}\boldsymbol{y}^\top\left(\Gamma + \sigma_n^{-2}U_{vx}^\top U_\Psi^{-1}U_{vx}\right)^{-1}\boldsymbol{y}\right).$$

Let L_{vv} be the Cholesky factorisation of U_ψ such that $U_\Psi = L_{vv}L_{vv}^\top$, $V_{vx} = L_{vv}^{-1}U_{vx}$, $M_{vv} = \sigma_n^2 I_{vv} + V_{vx}\Gamma^{-1}V_{vx}^\top = S_{vv}S_{vv}^\top$ where S_{vv} is the Cholesky factorisation of M_{vv}, and $\boldsymbol{\beta} = S_{vv}^{-1}V_{vx}\Gamma^{-1}\boldsymbol{y}$. By the matrix inversion lemma we have

$$\left(\Gamma + \sigma_n^{-1}V_{vx}^\top V_{vx}\sigma_n^{-1}\right)^{-1} = \Gamma^{-1} - \Gamma^{-1}V_{vx}^\top M_{vv}^{-1}V_{vx}\Gamma^{-1}$$

and

$$|\Gamma + \sigma_n^{-1}V_{vx}^\top V_{vx}\sigma_n^{-1}| = |\Gamma||\sigma_n^{-2}I_{vv}||M_{vv}|.$$

Hence we can write

$$\phi = \frac{1}{2}\left(\log|\Gamma| + \log|M_{vv}| + (p-q)\log(\sigma_n^2) + \sigma_n^{-2}\left(\boldsymbol{y}^\top\Gamma^{-1}\boldsymbol{y} - \boldsymbol{\beta}^\top\boldsymbol{\beta}\right)\right).$$

Derivatives of the Marginal Likelihood

Let's divide the cost functional into two parts $\phi = \phi_1 + \phi_2$ where

$$\begin{aligned}\phi_1 &= \frac{1}{2}\left(\log|\Gamma| + \log|M_{vv}| + (p-q)\log(\sigma_n^2)\right) \\ \phi_2 &= \frac{1}{2}\sigma_n^{-2}\left(\boldsymbol{y}^\top\Gamma^{-1}\boldsymbol{y} - \boldsymbol{\beta}^\top\boldsymbol{\beta}\right).\end{aligned}$$

Then we have

$$\mathrm{d}(\phi_1) = \frac{1}{2}(\mathrm{tr}[\Gamma^{-1}\mathrm{d}(\Gamma)]) + \frac{1}{2}\mathrm{d}(\log|M_{vv}|) + \frac{1}{2}(p-q)\sigma_n^{-2}\mathrm{d}(\sigma_n^2).$$

Let us define

$$A_{vv} = \sigma_n^2 U_\Psi + U_{vx}\Gamma^{-1}U_{vx}^\top.$$

Using $M_{vv} = L_{vv}^{-1}(\sigma_n^2 U_\Psi + U_{vx}\Gamma^{-1}U_{vx}^\top)L_{vv}^{-\top}$ we can show that

$$M_{vv} = L_{vv}^{-1}A_{vv}L_{vv}^{-\top}$$

as well as

$$\log|M_{vv}| = \log|A_{vv}| - \log|U_\Psi|.$$

Hence we can write

$$\mathrm{d}(A_{vv}) = \mathrm{d}(\sigma_n^2)U_\Psi + \sigma_n^2\mathrm{d}(U_\Psi) + \mathrm{d}(U_{vx})\Gamma^{-1}U_{vx}^\top + U_{vx}\mathrm{d}(\Gamma^{-1})U_{vx}^\top + U_{vx}\Gamma^{-1}\mathrm{d}(U_{vx}^\top),$$

and

$$\begin{aligned}\mathrm{d}(\log|M_{vv}|) &= \mathrm{tr}\left(A_{vv}^{-1}\left(\mathrm{d}(\sigma_n^2)U_\Psi + \sigma_n^2\mathrm{d}(U_\Psi) + U_{vx}\mathrm{d}(\Gamma^{-1})U_{vx}^\top + \right.\right. \\ &\left.\left. \mathrm{d}(U_{vx})\Gamma^{-1}U_{vx}^\top + U_{vx}\Gamma^{-1}\mathrm{d}(U_{vx}^\top)\right)\right) - \mathrm{tr}[U_\Psi^{-1}\mathrm{d}(U_\Psi)].\end{aligned}$$

If we define $(L_{vv}S_{vv})^{-\top}S_{vv}^{-1}V_{vx} = A_{vv}^{-1}U_{vx}.B_{vx} = (L_{vv}S_{vv})^{-\top}S_{vv}^{-1}V_{vx}\Gamma^{-\frac{1}{2}} = A_{vv}^{-1}U_{vx}\Gamma^{-\frac{1}{2}}$, then using $\mathrm{tr}[AB] = \mathrm{tr}[BA] = \mathrm{tr}[A^\top B^\top]$ we have

$$\begin{aligned}\mathrm{d}\log|M_{vv}| &= -\mathrm{tr}[B_{vx}\Gamma^{-\frac{1}{2}}\mathrm{d}(\Gamma)\Gamma^{-1}U_{vx}^\top] + 2\mathrm{tr}[B_{vx}\Gamma^{-1}\mathrm{d}(U_{vx})^\top] \\ &+ \mathrm{tr}[M_{vv}^{-1}\mathrm{d}(\sigma_n^2)] + \sigma_n^2\mathrm{tr}[A_{vv}^{-1}\mathrm{d}(U_\Psi)] - \mathrm{tr}[U_\Psi^{-1}\mathrm{d}(U_\Psi)]\end{aligned}$$

Note that

$$V_{vx}^\top M_{vv}^{-1} V_{vx} = V_{vx}^\top (L_{vv}^{-1} A_{vv} L_{vv}^{-\top})^{-1} V_{vx} = U_{vx}^\top A_{vv}^{-1} U_{vx}.$$

Accordingly we can derive

$$\begin{aligned}
\mathrm{d}(\Gamma^{-1} V_{vx}^\top M_{vv}^{-1} V_{vx} \Gamma^{-1}) \quad = \quad & \mathrm{d}(\Gamma^{-1}) U_{vx}^\top B_{vx} \Gamma^{-\frac{1}{2}} + \Gamma^{-\frac{1}{2}} B_{vx}^\top U_{vx} \mathrm{d}(\Gamma^{-1}) \\
& + \Gamma^{-1} \mathrm{d}(U_{vx}^\top) B_{vx} \Gamma^{-\frac{1}{2}} + \Gamma^{-\frac{1}{2}} B_{vx}^\top \mathrm{d}(U_{vx}) \Gamma^{-1} \\
& - \Gamma^{-\frac{1}{2}} B_{vx}^\top \mathrm{d}(A_{vv}) B_{vx} \Gamma^{-\frac{1}{2}}.
\end{aligned}$$

Recall that

$$\boldsymbol{\beta} = S_{vv}^{-1} V_{vx} \Gamma^{-1} \boldsymbol{y},$$

and define

$$\boldsymbol{v} = (L_{vv} S_{vv})^{-\top} \boldsymbol{\beta},$$

as well as

$$\boldsymbol{\mu} = \Gamma^{-\frac{1}{2}} V_{vx}^\top S_{vv}^{-\top} \boldsymbol{\beta}.$$

Then we have

$$\boldsymbol{v} = (L_{vv} S_{vv})^{-\top} S_{vv}^{-1} V_{vx} \Gamma^{-1} \boldsymbol{y} = B_{vx} \Gamma^{-\frac{1}{2}} \boldsymbol{y},$$

and by using $V_{vx}^\top S_{vv}^{-\top} \boldsymbol{\beta} = U_{vx}^\top \boldsymbol{v}$,

$$\boldsymbol{\mu} = \Gamma^{-\frac{1}{2}} U_{vx}^\top \boldsymbol{v}.$$

We can show that

$$\begin{aligned}
\boldsymbol{y}^\top \mathrm{d}(\Gamma^{-1} U_{vx}^\top A_{vv}^{-1} U_{vx} \Gamma^{-1}) \boldsymbol{y} \quad = \quad & (\boldsymbol{\mu}^\top \Gamma^{-\frac{1}{2}} - 2\boldsymbol{y}^\top \Gamma^{-1}) \mathrm{d}(\Gamma) \Gamma^{-\frac{1}{2}} \boldsymbol{\mu} \\
& + 2(\boldsymbol{y}^\top \Gamma^{-1} - \boldsymbol{\mu}^\top \Gamma^{-\frac{1}{2}}) \mathrm{d}(U_{vx})^\top \boldsymbol{v} \\
& - \boldsymbol{v}^\top \mathrm{d}(\sigma_n^2) \Gamma^{\frac{1}{2}} \boldsymbol{\mu} - \sigma_n^2 \boldsymbol{v}^\top \mathrm{d}(U_\Psi) \boldsymbol{v},
\end{aligned}$$

and finally that

$$\mathrm{d}(\phi_2) \quad = \quad -\frac{1}{2} \sigma_n^{-2} \left(\boldsymbol{y}^\top \Gamma^{-1} \mathrm{d}(\Gamma) \Gamma^{-1} \boldsymbol{y} + \mathrm{d}(\boldsymbol{y}^\top \Gamma^{-1} V_{vx}^\top M_{vv}^{-1} V_{vx} \Gamma^{-1} \boldsymbol{y}) \right).$$

Note that for reasons of efficiency $\mathrm{d}(\Gamma)$ should not be calculated explicitly. For example, the time complexity of calculating $\mathrm{d}(\Gamma)$ with respect to the basis is $O(m^2 nd)$. However, by noting that each occurrence of $\mathrm{d}(\Gamma)$ is accompanied by the pre-multiplication of a vector it is possible to reduce the complexity to $O(mnd)$ plus a single pre-calculation cost of $O(m^2 n)$.