Sparse Gaussian Processes with Spherical Harmonic Features

Vincent Dutordoir [1 2] **Nicolas Durrande** [1] **James Hensman** [3]

Abstract

We introduce a new class of inter-domain variational Gaussian processes (GP) where data is mapped onto the unit hypersphere in order to use spherical harmonic representations. Our inference scheme is comparable to variational Fourier features, but it does not suffer from the curse of dimensionality, and leads to diagonal covariance matrices between inducing variables. This enables a speed-up in inference, because it bypasses the need to invert large covariance matrices. Our experiments show that our model is able to fit a regression model for a dataset with 6 million entries two orders of magnitude faster compared to standard sparse GPs, while retaining state of the art accuracy. We also demonstrate competitive performance on classification with non-conjugate likelihoods.

1. Introduction

Gaussian processes (GPs) (Rasmussen & Williams, 2006) provide a flexible framework for modelling unknown functions: they are robust to overfitting, offer good predictive uncertainty estimates and allow us to incorporate prior assumptions into the model. Given a dataset with some inputs $\mathbf{X} \in \mathbb{R}^{N \times d}$ and outputs $\mathbf{y} \in \mathbb{R}^N$, a GP regression model assumes $y_i = f(\mathbf{x}_i) + \varepsilon_i$ where f is a GP over $\mathbb{R}^d$ and where the ε_i are normal random variables accounting for observation noise. The model predictions at $\mathbf{x} \in \mathbb{R}^d$ are then given by the posterior distribution $f \,|\, \mathbf{y}$. However, computing the posterior distribution usually scales $\mathcal{O}(N^3)$, because it requires solving a system involving an $N \times N$ matrix.

A range of sparse GP methods (see Quiñonero-Candela & Rasmussen (2005) for an overview) have been developed to improve on this $\mathcal{O}(N^3)$ scaling. Among them, variational inference is a practical approach allowing regression (Titsias, 2009), classification (Hensman et al., 2015), minibatching (Hensman et al., 2013) and structured models including latent variables, time-series and depth (Titsias & Lawrence, 2010; Frigola et al., 2014; Hensman & Lawrence, 2014; Salimbeni & Deisenroth, 2017). Variational inference in GPs works by approximating the exact (but intractable) posterior $f \,|\, \mathbf{y}$. This approximate posterior process is constructed from a conditional distribution based on M pseudo observations of $u_i = f(\mathbf{z}_i)$ at locations $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^M$. The approximation is optimised by minimising the Kullback-Leibler divergence between the approximate posterior and the exact one. The resulting complexity is $\mathcal{O}(M^3 + M^2 N)$, so choosing $M \ll N$ enables significant speedups compared to vanilla GP models. However, when using common stationary kernels such as Matérn, Squared Exponential (SE) or rational quadratic, the influence of pseudo-observations are only local and limited to the neighbourhoods of the inducing points $\mathbf{Z}$, so a large number of inducing points M may be required to cover the input space. This is especially problematic for higher dimensional data as a result of the curse of dimensionality.

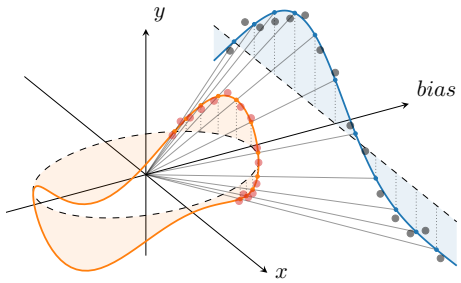

Figure 1. Illustration of the mapping between a 2D dataset (grey dots) embedded into a 3D space and its projection (orange dots) onto the unit half-circle using a linear mapping.

Variational Fourier Features (Hensman et al., 2017, VFF) inducing variables have been proposed to overcome this limitation. In VFF the inducing variables based on pseudo observations are replaced with inducing variables obtained by projecting the GP onto the Fourier basis. This results in inducing variables that have global influence on the predictions. For one-dimensional inputs, this construction leads to covariances matrices between inducing variables that are

[1]PROWLER.io, Cambridge, United Kingdom [2]Department of Engineering, University of Cambridge, Cambridge, United Kingdom [3]Amazon Research, Cambridge, United Kingdom (work done while JH was affiliated to PROWLER.io). Correspondence to: Vincent Dutordoir <vincent@prowler.io>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

almost diagonal and this can be exploited to reduce the complexity to $\mathcal{O}(N + M^2 N)$. However, for d-dimensional input spaces VFF requires construction of a new basis given by the outer product of the one-dimensional Fourier basis. This implies that the number of inducing variables grows exponentially with the dimensionality, which limits the use of VFF to just one dimension or two. Furthermore, VFF is restricted to kernels of the Matérn family.

In this work we improve upon VFF in multiple directions. Rather than using a sine and cosine basis, we use a basis of spherical harmonics to define a new interdomain sparse approximation. As we will show, spherical harmonics are the eigenfunctions of stationary kernels on the hypersphere, which allows us to exploit the Mercer representation of the kernel for defining the inducing variables. In arbitrary dimensions, our method leads to *diagonal* covariance matrices which makes it faster than VFF as we fully bypass the need to compute expensive matrix inverses. Compared to both sparse GPs and VFF, our approximation scheme suffers less from the curse of dimensionality. As VFF, each spherical harmonic inducing function has a global influence, but there is a natural ordering of the spherical harmonics that can guarantee that the best features are picked given an overall budget for the number M of inducing variables. Moreover, our method works for any stationary kernel on the sphere.

Following the illustration in fig. 1, we outline the different steps of our method. We start by concatenating the data with a constant input (bias) and project it linearly onto the unit hypersphere $\mathbb{S}^{d-1}$. We then learn a sparse GP on the sphere based on the projected data. We can do this extremely efficiently by making use of our spherical harmonic inducing variables, shown in fig. 2. Finally, the linear mapping between the sphere and the sub-space containing the data can then be used to map the predictions of the GP on the sphere back to the original $(\mathbf{x}, y)$ space.

This paper is organised as follows. In section 2 we give the necessary background on sparse GPs and VFF. In section 3 we highlight that every stage of the proposed method can be elegantly justified. For example, the linear mapping between the data and the sphere is a property of some specific covariance functions such as the arc-cosine kernel, from which we can expect good generalisation properties (section 3.1). Another example is that the spherical harmonics are coupled with the structure of the Reproducing Kernel Hilbert Space (RKHS) norm for stationary kernels on the sphere, which makes them very natural candidates for basis functions for sparse GPs (section 3.2). In section 3.3 we define our spherical harmonic inducing variables and compute the necessary components to do efficient GP modelling. Section 4 is dedicated to the experimental evaluation.

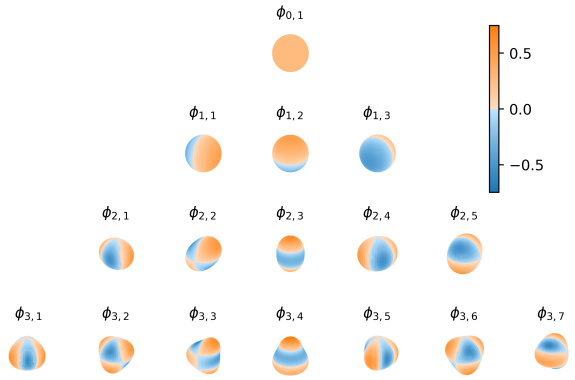

Figure 2. The first four levels of spherical harmonic functions in $\mathbb{R}^3$. The domain of the spherical harmonics is the surface of the unit sphere $\mathbb{S}^2$. The function value is given by the color and radius.

2. Background

2.1. GPs and Sparse Variational Inference

GPs are stochastic processes such that the distribution of any finite dimensional marginal is multivariate normal. The distribution of a GP is fully determined by its mean $\mu(\cdot)$ and covariance function (kernel) $k(\cdot, \cdot)$. GP models typically consist of combining a latent (unobserved) GP $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$ with a likelihood that factorises over observation points $p(\mathbf{y} \mid f) = \prod_n p(y_n \mid f(\mathbf{x}_n))$. When the observation model is $y_n \mid f(\mathbf{x}_n) \sim \mathcal{N}\big(f(\mathbf{x}_n), \tau^2\big)$, the posterior distributions of f and y given some data are still Gaussian and can be computed analytically. Let $\varepsilon_i \sim \mathcal{N}\big(0, \tau^2\big)$ represent the observation noise at $\mathbf{x}_i$, then the GP posterior distribution is $f \mid \mathbf{y} \sim \mathcal{GP}(m, v)$ with

$$m(\mathbf{x}) = \mathbf{k_f}(\mathbf{x})(\mathbf{K_{ff}} + \tau^2 I)^{-1}\mathbf{y},$$
$$v(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') - \mathbf{k_f}(\mathbf{x})^\top(\mathbf{K_{ff}} + \tau^2 I)^{-1}\mathbf{k_f}(\mathbf{x}'),$$

where $\mathbf{K_{ff}} = [k(\mathbf{x}_i, \mathbf{x}_j)]_{i,j=1}^N$ and $\mathbf{k_f}(\mathbf{x}) = [k(\mathbf{x}_n, \mathbf{x})]_{n=1}^N$.

Computing this exact posterior requires inverting the $N \times N$ matrix $\mathbf{K_{ff}}$, which has a $\mathcal{O}(N^3)$ computational complexity and a $\mathcal{O}(N^2)$ memory footprint. Given a typical current hardware specification, this limits the dataset size to the order of few thousand observations. Furthermore, there is no known analytical expression for posterior distribution when the likelihood is not conjugate, as encountered in classification for instance.

Sparse GPs combined with variational inference provide an elegant way to address these two shortcomings (Titsias, 2009; Hensman et al., 2013; 2015). It consists of introducing a distribution $q(f)$ that depends on some parameters, and finding the values of these parameters such that $q(f)$ gives the best possible approximation of the exact posterior $p(f \mid \mathbf{y})$. Sparse GPs introduce M pseudo

inputs $\mathbf{Z} \in \mathbb{R}^{M \times d}$ corresponding to M inducing variables $\mathbf{u} \sim \mathcal{N}(\mathbf{m}, \mathbf{S})$, and choose to write the approximating distribution as $q(f) = q(\mathbf{u})p(f \mid f(\mathbf{Z}) = \mathbf{u})$. This results in a distribution $q(f)$ that is parametrised by the variational parameters $\mathbf{m} \in \mathbb{R}^M$, and $\mathbf{S} \in \mathbb{R}^{M \times M}$, which are learned by minimising the Kullback–Leibler (KL) divergence $\mathrm{KL}[q(f) \,\|\, p(f \mid \mathbf{y})]$.

At prediction time, the conjugacy between $q(\mathbf{u})$ and the conditioned posterior $f \mid f(\mathbf{Z}) = \mathbf{u}$ implies that $q(f)$, where $\mathbf{u}$ is marginalised out, is a GP with a mean $\mu(\mathbf{x})$ and a covariance function $\nu(\mathbf{x}, \mathbf{x}')$ that can be computed in closed:

$$\mu(\mathbf{x}) = \mathbf{k_u}(\mathbf{x})\mathbf{K_{uu}^{-1}}\mathbf{m} \tag{1}$$
$$\nu(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') + \mathbf{k_u}(\mathbf{x})^\top \mathbf{K_{uu}^{-1}}(\mathbf{S} - \mathbf{K_{uu}})\mathbf{K_{uu}^{-1}}\mathbf{k_u}(\mathbf{x}'),$$

where $[\mathbf{K_{uu}}]_{m,m'} = \mathbb{E}[f(\mathbf{z}_m)\, f(\mathbf{z}_{m'})]$, and $[\mathbf{k_u}(\mathbf{x})]_m = \mathbb{E}[f(\mathbf{z}_m)\, f(\mathbf{x})]$.

Sparse GPs result in $\mathcal{O}(M^2N + M^3)$ and $\mathcal{O}(M^3)$ computational complexity at training (minimising the KL) and prediction respectively. Picking $M \ll N$ can thus result in drastic improvement, but the lower M is, the less accurate the approximation, as recently shown by Shi et al. (2020). Typical kernels—such as Matérn or Squared Exponential (SE)—depend on a lengthscale parameter that controls how quickly the correlation between two evaluations of f drops for two inputs that move away from another. For short lengthscales, this correlation drops quickly and two observations can be almost independent even for two input points that are close by in the input space. For a sparse GP model, this implies that $\mu(\mathbf{x})$ and $\nu(\mathbf{x}, \mathbf{x})$ will rapidly revert to the prior mean and variance when $\mathbf{x}$ is not in the immediate neighbourhood of an inducing point $\mathbf{z}_i$. A similar effect can be observed when the input space is high-dimensional: because inducing variables only have a local influence, the number of inducing points required to cover the space grows exponentially with the input space dimensionality. In practice, it may thus be required to pick large values for M to obtain accurate approximate posteriors but this defeats the original intent of sparse methods.

This behaviour where the vector of features $\mathbf{k_u}(\cdot)$ of the approximate distribution are given by kernel function $k(\mathbf{z}_i, \cdot)$ can be addressed using interdomain inducing variables.

2.2. Inter-domain GPs and Variational Fourier Features

Inter-domain Gaussian processes (see Lázaro-Gredilla & Figueiras-Vidal (2009) and van der Wilk et al. (2020) for a modern exposition) use alternative forms of inducing variables such that the resulting sparse GP models result in more informative features. In interdomain GPs the inducing variables are obtained by integrating the GP f with an inducing function:

$$u_m = \int f(\mathbf{x})\, g_m(\mathbf{x})\, \mathrm{d}\mathbf{x}\,.$$

This redefinition of $\mathbf{u}$ implies that the expressions of $\mathbf{K_{uu}}$ and $\mathbf{k_u}$ change, but the inference scheme of interdomain GPs and the mathematical expressions for the posterior mean and variance are exactly the same as classic sparse GPs. Depending on the choice of $g_m(\cdot)$, interdomain GPs can result in various feature vector $\mathbf{k_u}(\cdot)$. These feature vectors can alleviate the classic sparse GP limitation of inducing variables having only a local influence.

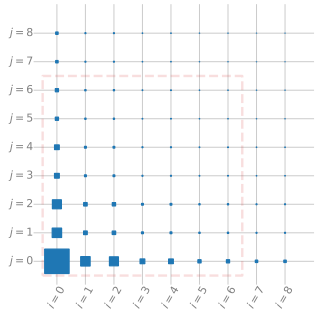

Figure 3. Illustration of the variance of the inducing variables when using VFF with a two dimensional input space. Each pair (i, j) corresponds to an inducing function $\psi_i(x_1)\psi_j(x_2)$, and the area of each square is proportional to the variance of the associated inducing variable. For a given number of inducing variables, say $M = 49$ as highlighted with the red dashed line, VFF selects features that do not carry signal (north-east quarter of the red dashed square) whereas it ignores features that are expected to carry signal (along the axes $i = 0$ and $j = 0$).

VFF (Hensman et al., 2017) is an interdomain method where the inducing variables are given by a Matérn RKHS inner product between the GP and elements of the Fourier basis:

$$u_m = \langle f, \psi_m \rangle_{\mathcal{H}},$$

where $\psi_0 = 1$, $\psi_{2i} = \cos(ix)$ and $\psi_{2i+1} = \sin(ix)$ if the input space is $[0, 2\pi]$. This leads to

$$\mathbf{K_{uu}} = [\langle \psi_i, \psi_j \rangle_{\mathcal{H}}]_{i,j=0}^{M-1} \text{ and } \mathbf{k_u}(x) = [\psi_i(x)]_{i=0}^{M-1}\,.$$

This results in several advantages. First, the features $\mathbf{k_u}(x)$ are exactly the elements of the Fourier basis, which are independent of the kernel parameters and can be precomputed. Second, the matrix $\mathbf{K_{uu}}$ is the sum of a diagonal matrix plus low rank matrices. This structure can be used to drastically reduce the computational complexity, and the experiments showed one or two orders of magnitude speed-ups compared to classic sparse GPs. Finally, the variance of the inducing variables typically decays quickly with increasing frequencies, which means that by selecting the first M elements of the Fourier basis we pick the features that carry the most signal.

The main flaw of VFF comes from the way it generalises to multidimensional input spaces. The approach in Hensman et al. (2017) for getting a set of d-dimensional inducing functions consists of taking the outer product of d univariate basis and to consider separable kernels so that the elements of $\mathbf{K_{uu}}$ are given by the product of the inner products in each dimension. For example, in dimension 2, a set of M^2 inducing functions is given by $\{(x_1, x_2) \mapsto \psi_i(x_1)\psi_j(x_2)\}_{0\leq i,j\leq M-1}$, and entries on $\mathbf{K_{uu}}$ are $\langle \psi_i\psi_j, \psi_k\psi_l\rangle_{\mathcal{H}_{k_1k_2}} = \langle \psi_i, \psi_k\rangle_{\mathcal{H}_{k_1}} \langle \psi_j, \psi_l\rangle_{\mathcal{H}_{k_2}}$. This construction scales poorly with the dimension: for example choosing a univariate basis as simple as $\{1, \cos, \sin\}$ for an eight-dimensional problem already results in more that 6,500 inducing functions. Additionally, this construction is very inefficient in terms of captured variance, as we illustrate in Figure 3 for a 2D input space. The figure shows that the prior variance associated with the inducing function $\psi_i(x_1)\psi_j(x_2)$ vanishes quickly when both i and j increase. This means that most of the inducing functions on which the variational posterior is built are irrelevant, whereas some important ones such as $\psi_i(x_1)\psi_0(x_2)$ or $\psi_0(x_1)\psi_j(x_2)$ for $i, j \geq \sqrt{M}$ are important but ignored. Although we used a 2D example to illustrate this poor behaviour, it is important to bear in mind that the issue gets exacerbated for higher dimensional input spaces. As detailed in fig. 4 and discussed later, our proposed approach does not suffer from such behaviour.

3. Variational Inference with Spherical Harmonics

In this section we describe the three key ingredients of the proposed approach: the mapping of dataset to the hypersphere, the definition of GPs on this sphere, and the use of spherical harmonics as inducing functions for such GPs.

3.1. Linear Mapping to the Hypersphere

As illustrated in fig. 1, for a 1D dataset, the first step of our approach involves appending the dataset with a dummy input value b that we will refer to as the *bias* (grey dots in fig. 1). For convenience we will overload the notation, and refer to the augmented inputs $(\mathbf{x}, b)$ as $\mathbf{x}$ from now on, denote its dimension by d, and assume $x_d = b$. The next step is to project the augmented data onto the unit hypersphere $\mathbb{S}^{d-1}$ according to a linear mapping: $(\mathbf{x}, y) \mapsto (\mathbf{x}/||\mathbf{x}||, y/||\mathbf{x}||)$ as depicted by the orange dots. Given the projected data, we learn a GP model on the sphere (orange function), and the model predictions can be mapped back to the hyperplane (blue curve) using the inverse linear mapping (i.e. following the fine grey lines starting from the origin).

Arc-Cosine kernel. Although this setup may seem very arbitrary, it is inspired by the important works on the limits of neural networks as Gaussian processes, especially the *arc-sine* kernel (Williams, 1998) and the *arc-cosine* kernels (Cho & Saul, 2009). An arc-cosine kernel corresponds to the infinitely-wide limit of a single-layer ReLU-activated network with Gaussian weights. Let $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$, such that $x_d = x'_d = b$, be an augmented input vector whose last entry corresponds to the bias. Then the arc-cosine kernel can be written as

$$k_{ac}(\mathbf{x}, \mathbf{x}') = ||\mathbf{x}||\,||\mathbf{x}'||\,\underbrace{\frac{1}{\pi}\big(\sin\theta + (\pi - \theta)\cos\theta\big)}_{=J(\theta)},$$
$$\text{where} \quad \theta = \cos^{-1}\left(\frac{\mathbf{x}^\top \mathbf{x}'}{||\mathbf{x}||||\mathbf{x}'||}\right).$$

The kernel is a function of the norm of the inputs $||\mathbf{x}||\,||\mathbf{x}'||$ and a factor $J(\theta)$ that only depends on the geodestic distance θ (or the great-circle distance) between the projection of $\mathbf{x}$ and $\mathbf{x}'$ on the unit hypersphere.

Deep Arc-Cosine kernel. Cho & Saul (2009) also introduced the kernel that corresponds to ℓ successive applications of infinitely-wide fully-connected neural network layers with ReLU activations. It can be written as

$$k^{(\ell)}_{ac}(\mathbf{x}, \mathbf{x}') = ||\mathbf{x}||\,||\mathbf{x}'||\,J(\theta^{(\ell)}), \quad \text{for } \ell \in \mathbb{N}_{>0}$$

with $\theta^{(\ell)} = \cos^{-1}(J(\theta^{(\ell-1)}))$ and $\theta^{(1)} = \theta$. The resulting kernel mimics the computations in large multi-layer ReLU-activated networks.

We observe that k_{ac} and $k^{(\ell)}_{ac}$ are separable in two parts: a part that only depends on the norm of the inputs (radial part) and a part that depends on the geodestic distance θ between the input vectors (angular part). We will later refer to this type of kernels as *zonal* kernels. Furthermore, the dependence in the radius is just the linear kernel. Since the linear kernel is degenerate, it means that there is a one-to-one mapping between the GP samples restricted to the unit sphere $\mathbb{S}^{d-1}$ and the samples over $\mathbb{R}^d$, and that this mapping is exactly the linear transformation we introduced previously.

One insight provided by the link between our settings and the (deep) arc-cosine kernel is that once mapped back to the original space, our GP samples will exhibit linear asymptotic behaviour, which can be compared to the way ReLU-activated neural networks operate. Although a proper demonstration would require further work, this correspondence suggests that our models may inherit the desirable generalisation properties of neural networks.

Having mapped the data to the sphere, we must now introduce GPs on the sphere and their covariance. The following section provides some theory that allows us to work with various kernels on the sphere, including the arc-cosine kernel as a special case.

3.2. Mercer's Theorem for Zonal Kernels on the Sphere

Stationary kernels, i.e. translation invariant covariances, are ubiquitous in machine learning when the input space is Euclidean. When working on the hypersphere, their spherical counterpart are zonal kernels, which are invariant to rotations. More precisely, a kernel $k : \mathbb{S}^{d-1} \times \mathbb{S}^{d-1} \mapsto \mathbb{R}$ is called zonal if there exists a shape function s such that $k(\mathbf{x}, \mathbf{x}') = s(\mathbf{x}^\top \mathbf{x}')$. From hereon, we focus on zonal kernels. Since stationary kernels are functions of $\mathbf{x} - \mathbf{x}'$ they have the property that $\Delta_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') = \Delta_{\mathbf{x}'} k(\mathbf{x}, \mathbf{x}')$, where $\Delta_{\mathbf{x}} = \sum_{i=1}^{d} \partial^2 / \partial^2 x_i$ is the Laplacian operator. Such property is also verified by zonal kernels: $\Delta_{\mathbf{x}}^{\mathbb{S}^{d-1}} k(\mathbf{x}, \mathbf{x}') = \Delta_{\mathbf{x}'}^{\mathbb{S}^{d-1}} k(\mathbf{x}, \mathbf{x}')$, where we denote by $\Delta_{\mathbf{x}}^{\mathbb{S}^{d-1}}$ the Laplace-Beltrami operator with respect to the variable $\mathbf{x}$. Combined with an intregration by part, this property can be used to show that the kernel operator $\mathcal{K}$ of a zonal covariance and the Laplace-Beltrami operator commute:

$$\begin{aligned}
\mathcal{K}\left[\Delta^{\mathbb{S}^{d-1}} g\right] &= \int_{\mathbb{S}^{d-1}} k(\mathbf{x}, \cdot)\left[\Delta_{\mathbf{x}}^{\mathbb{S}^{d-1}} g(\mathbf{x})\right] \mathrm{d}\mathbf{x} \\
&= \int_{\mathbb{S}^{d-1}} g(\mathbf{x}) \Delta_{\mathbf{x}}^{\mathbb{S}^{d-1}} k(\mathbf{x}, \cdot) \mathrm{d}\mathbf{x} \\
&= \int_{\mathbb{S}^{d-1}} g(\mathbf{x}) \Delta^{\mathbb{S}^{d-1}} k(\mathbf{x}, \cdot) \mathrm{d}\mathbf{x} = \Delta^{\mathbb{S}^{d-1}} \mathcal{K} g
\end{aligned}$$

which in turn implies that these two operators share the same eigenfunctions. This result is of particular relevance to us since there is a huge body of literature on diagonalisation of the Laplace-Beltrami operator on $\mathbb{S}^{d-1}$, and that it is well known that its eigenfunctions are given by the spherical harmonics. This reasoning can be summarised by the following theorem:

Theorem 1 (Mercer representation)**.** *Any zonal kernel k on the hypersphere can be decomposed as*

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N_\ell^d} \widehat{a}_{\ell,k} \phi_{\ell,k}(\mathbf{x}) \phi_{\ell,k}(\mathbf{x}'), \tag{2}$$

where $\mathbf{x}, \mathbf{x}' \in \mathbb{S}^{d-1}$ and $\widehat{a}_{\ell,k}$ are positive coefficients, $\phi_{\ell,k}$ denote the elements of the spherical harmonic basis in $\mathbb{S}^{d-1}$, and N_ℓ^d corresponds to the number of spherical harmonics for a given level ℓ.

Although it is typically stated without a proof, this theorem is already known in some communities (see Wendland (2005) for a functional analysis exposition, or Peacock (1999) for its use in cosmology).

Given the Mercer representation of a zonal kernel, its RKHS can be characterised by

$$\mathcal{H} = \left\{ g = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N_\ell^d} \widehat{g}_{\ell,k} \phi_{\ell,k} : \sum_{\ell=0}^{\infty} \sum_{k=1}^{N_\ell^d} \frac{|\widehat{g}_{\ell,k}|^2}{\widehat{a}_{\ell,k}} < \infty \right\}$$

with the inner product between two functions $g(\mathbf{x}) = \sum_{\ell,k} \widehat{g}_{\ell,k} \phi_{\ell,k}(\mathbf{x})$ and $h(\mathbf{x}) = \sum_{\ell,k} \widehat{h}_{\ell,k} \phi_{\ell,k}(\mathbf{x})$ defined as

$$\langle g, h \rangle_{\mathcal{H}} = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N_\ell^d} \frac{\widehat{g}_{\ell,k} \widehat{h}_{\ell,k}}{\widehat{a}_{\ell,k}}. \tag{3}$$

It is straightforward to show that this inner product satisfies the reproducing property (see Appendix section B.2).

In order to make these results practical, we need to compute the coefficients $\widehat{a}_{\ell,k}$ for some kernels of interest. For a given value of $\mathbf{x}' \in \mathbb{S}^{d-1}$ we can see $\mathbf{x} \mapsto k(\mathbf{x}, \mathbf{x}')$ as a function from $\mathbb{S}^{d-1}$ to $\mathbb{R}$, and represent it in the basis of spherical harmonics:

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\ell=0}^{\infty} \sum_{k=1}^{N_\ell^d} \langle k(\mathbf{x}', \cdot), \phi_{\ell,k} \rangle_{L^2(\mathbb{S}^{d-1})} \phi_{\ell,k}(\mathbf{x}). \tag{4}$$

Combining equations (2) and (4) gives the following expression for the coefficients we are interested in: $\widehat{a}_{\ell,k} = \langle k(\mathbf{x}', \cdot), \phi_{\ell,k} \rangle_{L^2(\mathbb{S}^{d-1})} / \phi_{\ell,k}(\mathbf{x}')$. Although this expression involves a $(d-1)$ dimensional integral on the hypersphere, our hypothesis that k is zonal means we can make use of the Funk-Hecke formula (see theorem 4 in the supplementary) and rewrite it as a simpler 1D integral over the shape function of k. Following this procedure finally leads to

$$\widehat{a}_{\ell,k} = \frac{\omega_d}{C_\ell^{(\alpha)}(1)} \int_{-1}^{1} s(t)\, C_\ell^{(\alpha)}(t) (1 - t^2)^{\frac{d-3}{2}} \mathrm{d}t, \tag{5}$$

where $\alpha = \frac{d-2}{2}$, $C_\ell^{(\alpha)}$ is the Gegenbauer polynomial of degree ℓ and $s(\cdot)$ is the kernel's shape function. The constants ω_d and $C_\ell^{(\alpha)}(1)$ are given in theorem 4 in the supplementary.

Using eq. (5) we are able to compute the Fourier coefficients of any zonal kernel. The details of the calculations for the arc-cosine kernel restricted to the sphere is given in section B.3 in the supplementary material.

Alternatively, a key result from Solin & Särkkä (2014, eq. 20) is to show that the coefficients $\widehat{a}_{\ell,k}$ have a simple expression that depends on the kernel spectral density S and the eigenvalues of the Laplace-Beltrami operator. For GPs on $\mathbb{S}^{d-1}$, the coefficients boil down to $\widehat{a}_{\ell,k} = S(\sqrt{\ell(\ell + d - 2)})$. This is the expression we used in our experiments to define Matérn and SE covariances on the hypersphere. More details on this method are given in section B.4 in the supplementary.

As one may have noticed, the values of $\widehat{a}_{l,k}$ do not depend on the second index k (i.e. the eigenvalues only depend on the degree of the spherical harmonic, but not on its orientation). This is a remarkable property of zonal kernels which allows us to use the *addition theorem* (see supplementary material

theorem 3) for spherical harmonics to simplify eq. (2):

$$k(\mathbf{x}, \mathbf{x}') = \sum_{\ell=0}^{\infty} \widehat{a}_\ell \, \frac{\ell + \alpha}{\alpha} \, C_\ell^{(\alpha)}\big(\mathbf{x}^\top \mathbf{x}'\big).$$

This representation is cheaper to evaluate than eq. (2) but it still requires truncation for practical use.

3.3. Spherical Harmonics as Inducing Features

We can now build on the results from the previous section to propose powerful and efficient sparse GPs models. We want features $\mathbf{k_u}(\mathbf{x})$ that exhibit non-local influence for expressiveness, and inducing variables that induce sparse structure in $\mathbf{K_{uu}}$ for efficiency. We achieve this by defining the inducing variables u_m to be the inner product between the GP[1] and spherical harmonics:[2]

$$u_m = \langle f, \phi_m \rangle_{\mathcal{H}}. \tag{6}$$

To leverage these new inducing variables we need to compute two quantities: 1) the covariance between u_m and f for $\mathbf{k_u}(\mathbf{x})$, and 2) the covariance between the inducing variables themselves for the $\mathbf{K_{uu}}$ matrix. See van der Wilk et al. (2020) for an in-depth discussion of these concepts.

The covariance of the inducing variables with f are

$$[\mathbf{k_u}(\mathbf{x})]_m = \mathbb{E}[f(\mathbf{x})\, u_m] = \langle k(\mathbf{x}, \cdot), \phi_m \rangle_{\mathcal{H}} = \phi_m(\mathbf{x}),$$

where we relied on the linearity of both expectation and inner product and where we used the reproducing property of $\mathcal{H}$.

The covariance between the inducing variables is given by

$$[\mathbf{K_{uu}}]_{m,m'} = \mathbb{E}[u_m\, u_{m'}] = \langle \phi_m, \phi_{m'} \rangle_{\mathcal{H}} = \frac{\delta_{mm'}}{\widehat{a}_m},$$

where $\delta_{mm'}$ is the Kronecker delta. Crucially, this means that $\mathbf{K_{uu}}$ is a diagonal matrix with elements $1/(\widehat{a}_m)$.

Substituting $\mathbf{K_{uu}}$ and $\mathbf{k_u}(\mathbf{x})$ into the sparse variational approximation (eq. (1)), leads to the following form for $q(f)$

$$\mathcal{GP}\Big(\tilde{\mathbf{\Phi}}^\top(\mathbf{x})\mathbf{m};\; k(\mathbf{x}, \mathbf{x}') + \tilde{\mathbf{\Phi}}^\top(\mathbf{x})(\mathbf{S} - \mathbf{K_{uu}})\tilde{\mathbf{\Phi}}(\mathbf{x}')\Big),$$

with $\tilde{\mathbf{\Phi}}(\mathbf{x}) = [\widehat{a}_m \phi_m(\mathbf{x})]_{m=1}^M$.

This sparse approximation has two main differences compared to a SVGP model with standard inducing points.

[1] Although f does not belong to $\mathcal{H}$ (Kanagawa et al., 2018), such expression is well defined since the regularity of ϕ_m can be used to extend the domain of definition of the first argument of the inner product to a larger class of functions. See Hensman et al. (2017) for a detailed discussion.

[2] Note that in the context of inducing variables, we switch to a single integer m to index the spherical harmonics and order them first by increasing level ℓ, and then by increasing k within a level.

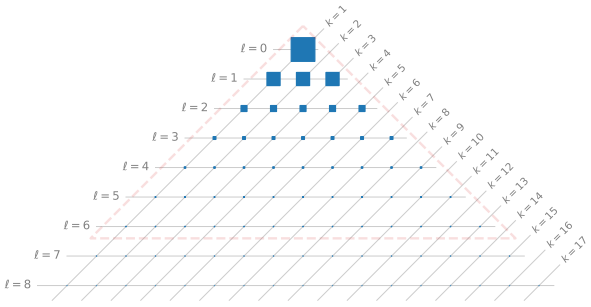

Figure 4. Illustration of the variance of the VISH inducing variables for a 2D input space. Settings are the same as in fig. 3 but a spherical harmonic feature $\phi_{\ell,k}$ is associated to each pair (ℓ, k). For a given number M of inducing variables, the truncation pattern (see red dashed triangle for $M = 49$) is optimal since it selects the most influential features.

First, the spherical harmonic inducing variables lead to features $k_{\mathbf{u}}(\mathbf{x})$ with non-local structure. Second, the approximation $q(f)$ does not require any inverses anymore. The computational bottleneck of this model is now simply the matrix multiplication in the variance calculation, which has a $\mathcal{O}(N_{\text{batchsize}} M^2)$ cost. Compared to the $\mathcal{O}(M^3 + N_{\text{batchsize}} M^2)$ cost of inducing point SVGPs, this gives a significant speedup – as we show in the experiments.

As is usual in sparse GP methods, the number of inducing variables M is constrained by the computational budget available to the user. Given that we ordered the spherical harmonic by increasing ℓ, choosing the first M elements means we will select first features with low angular frequency. Provided that the kernel spectral density is a decreasing function (this will be true for classic covariances, but not for quasi-periodic ones), this means that the selected features correspond to the ones carrying the most signal according to the prior. In other words, the decomposition of the kernel can be compared to an infinite dimensional principal component analysis, and our choice of the inducing function is optimal since we pick the ones with the largest variance. This is illustrated in fig. 4, which shows the analogue of fig. 3 for spherical harmonic inducing functions.

4. Experiments

We evaluate our method Variational Inference with Spherical Harmonics (VISH) on regression and classification problems and show the following properties of our method: 1) VISH performs competitively in terms of accuracy and uncertainty quantification on a range of problems from the UCI dataset repository. 2) VISH is extremely fast and accurate on large-scale conjugate problems (approximately 6 million 8D entries in less than 2 minutes). 3) Compared to VFF, VISH can be applied to multi-dimensional datasets and preserve its computational efficiency. 4) On problems

with non-conjugate likelihood our method does not suffer from some of the issues encountered by VFF.

We begin with a toy experiment in which we show that the approximation becomes more accurate as we increase the number of basis functions.

4.1. Toy Experiment: Banana Classification

The banana dataset is a 2D binary classification problem (Hensman et al., 2015). In fig. 5 we show three different fits of VISH with $M \in \{9,\ 225,\ 784\}$ spherical harmonics, which correspond respectively to maximum levels of 2, 14, and 27 for our inducing functions. Since the variational framework provides a guarantee that more inducing variables must be monotonically better (Titsias, 2009), we expect that increasing the number of inducing functions will provide improved approximations. This is indeed the case as we show in the rightmost panel: with increasing M the ELBO converges and the fit becomes tighter.

While this is expected behaviour for SVGP methods, it is not guaranteed by VFF. Given the Kronecker-structure used by VFF for this 2D experiment Hensman et al. (2017) report that using a full rank covariance matrix for the variational distribution was intolerably slow. They also show that enforcing the Kronecker structure on the posterior results in an ELBO that *decreased* as frequencies were added, and they finally propose a sum-of-two-Kroneckers structure, but provide no guarantee that this would converge to the exact posterior in the limit of larger M. In VISH we do not need to impose any structure on the approximate covariance matrix $\mathbf{S}$, so we retain the guarantee that adding more basis functions will move us closer to the posterior process. The method remains fast despite optimising over full covariance matrices: fitting the models displayed in fig. 5 only takes a few seconds on a standard desktop.

4.2. Regression on UCI Benchmarks

We use five UCI regression datasets to compare the performance of our method against other GP approaches. We measure accuracy of the predictive mean with Mean Squared Error (MSE) and uncertainty quantification with mean Negative Log Predictive Density (NLPD). For each dataset we randomly select 90% of the data for training and 10% for testing and repeat this 5 times to get error bars. We normalise the inputs and the targets to be centered unit Gaussian. We report the MSE and NLPD of the normalised data. In Table 1 we report the performance of VISH, Additive-VFF (A-VFF) (Hensman et al., 2017), SVGP (Hensman et al., 2013), Additive-GRP (A-GPR) and GPR.

We start by comparing VISH against SVGP, and notice that for Energy, Concrete and Power the change in inductive bias and expressiveness of spherical harmonic inducing variables opposed to standard inducing points improves performance. Also, while the sparse methods (A-VFF, VISH and SVGP) are necessary for scalability (as highlighted by the next experiment), they remain inferior to the exact GPR model – which should be seen as the optimal baseline.

For VFF we have to resort to an additive model (A-VFF) in order to deal with the dimensionality of the data, as a vanilla VFF model can only be used in one or two dimensions. Following (Hensman et al., 2017, eq. 78), we assume a different function for each input dimension

$$f(x) = \sum_{d=1}^{D} f_d(x_d), \text{ with } f_d \sim \mathcal{GP}(0, k_d), \qquad (7)$$

and approximate this process by a mean-field approximate posterior over the processes $q(f_1, \ldots, f_D) = \prod_d q(f_d)$, where each process is a SVGP $q(f_d) = \int p(f_d \,|\, \mathbf{u}_d) q(\mathbf{u}_d) \mathrm{d}\mathbf{u}_d$. We used $M = 30$ frequencies per input dimension. As extra baseline we added an exact GPR model which makes the same additive assumption (A-GPR). As expected, not having to impose this structure improves performance; we see that VISH beats A-VFF on every dataset.

Limitations Our current implementation of VISH only supports datasets up to 8 dimensions (9 dimensions when the bias is concatenated). This is not caused by a theoretical limitation because our approach leads to a diagonal $\mathbf{K_{uu}}$ matrix in any dimension. The problem stems from the fact that there are no libraries available providing stable spherical harmonic implementations in high dimensions (needed for $k_{\mathbf{u}}(\cdot)$). Our implementation based on Dai & Xu (2013, Theorem 5.1) is stable up to 9 dimensions and future work will focus on scaling this up. Furthermore, VISH does not solve the curse of dimensionality for GP models but does drastically improve over the scaling of VFF.

4.3. Large-Scale Regression on Airline Delay

This experiment illustrates three core capabilities of VISH: 1) it can deal with large datasets and 2) it is computationally and time efficient 3) the model improves performance in terms of NLPD.

We use the 2008 U.S. airline delay dataset to asses these capabilities. The goal of this problem is to predict the amount of delay y given eight characteristics $\mathbf{x}$ of a flight, such as the age of the aircraft (number of years since deployment), route distance, airtime, etc. We follow the exact same experiment setup as Hensman et al. (2017)[3] and evaluate the performance on 4 datasets of size 10,000, 100,000, 1,000,000, and 5,929,413 (complete dataset), created by subsampling the original one. For each dataset we use two thirds of the

[3] https://github.com/jameshensman/VFF

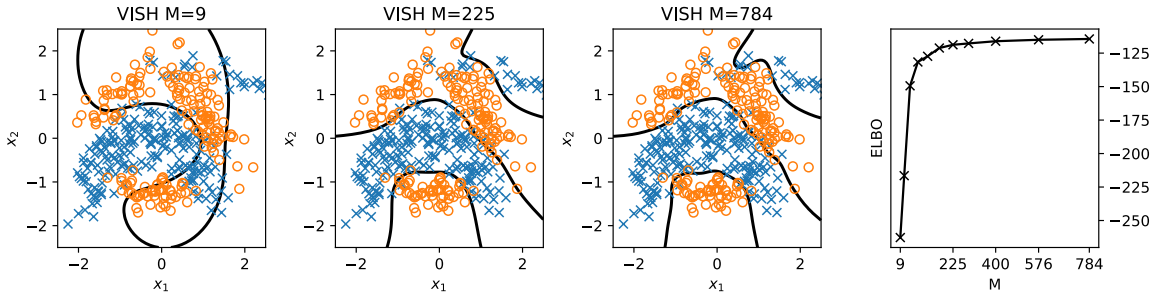

Figure 5. Classification of the 2D banana dataset with growing number of spherical harmonic basis functions. The right plot shows the convergence of the ELBO with respect to increasing numbers of basis functions.

				MSE					NLPD				
Dataset	N↓	D	M	VISH	A-VFF	SVGP	A-GPR	GPR	VISH	A-VFF	SVGP	A-GPR	GPR
Yacht	308	6	294	0.004 ± 0.00	0.010 ± 0.01	0.002 ± 0.00	0.010 ± 0.00	$\mathbf{0.001 \pm 0.00}$	-1.698 ± 0.21	-0.861 ± 0.36	-1.72 ± 0.12	-0.903 ± 0.16	$\mathbf{-2.420 \pm 0.15}$
Energy	768	8	210	$\mathbf{0.003 \pm 0.00}$	0.011 ± 0.00	0.013 ± 0.00	0.012 ± 0.00	$\mathbf{0.003 \pm 0.00}$	$\mathbf{-1.575 \pm 0.13}$	-0.824 ± 0.11	-0.61 ± 0.04	-0.797 ± 0.11	-1.461 ± 0.12
Concrete	1030	8	210	0.122 ± 0.02	0.123 ± 0.01	0.128 ± 0.01	0.099 ± 0.01	$\mathbf{0.096 \pm 0.01}$	0.336 ± 0.07	0.371 ± 0.06	0.374 ± 0.03	0.268 ± 0.06	$\mathbf{0.228 \pm 0.12}$
Kin8mn	8192	8	210	0.219 ± 0.01	0.561 ± 0.02	0.116 ± 0.01	0.556 ± 0.02	$\mathbf{0.076 + 0.00}$	0.612 ± 0.02	1.129 ± 0.01	0.381 ± 0.03	1.125 ± 0.02	$\mathbf{0.116 \pm 0.02}$
Power	9568	4	336	0.054 ± 0.00	0.058 ± 0.00	0.055 ± 0.00	0.032 ± 0.00	$\mathbf{0.046 \pm 0.00}$	-0.005 ± 0.03	-0.002 ± 0.04	-0.029 ± 0.03	-0.306 ± 0.07	$\mathbf{-0.114 \pm 0.04}$

Table 1. Predictive mean squared errors (MSEs) and negative log predictive densities (NLPDs) with one standard deviation based on 5 splits on 5 UCI regression datasets. Lower is better. All models assume a Gaussian noise model, use a Matérn-3/2 kernel and use the L-BFGS optimiser for the hyper- and variational parameters. VISH and SVGP are configured with the same number of inducing points M. A-VFF and A-GPR assume an Additive structure over the inputs (see text). For A-VFF and VISH the optimal posterior distribution for the inducing variables is set following Titsias (2009).

data for training and one third for testing. Every split is repeated 10 times and we report the mean and one standard deviation of the MSE and NLPD. For every run the outputs are normalized to be a centered unit Gaussian. The inputs are normalized to [0, 1] for VFF and SVGP. For VISH we normalize the inputs so that each column falls within $[-v_d, v_d]$. The hyperparameter v_d corresponds to the prior variance of the weights of an infinite-width fully-connected neural net layer (see Cho & Saul (2009)). We can optimise for this weight-variance by back-propagation through $k_u(x)$ w.r.t. the ELBO. This is similar to the lengthscale hyperparameters of stationary kernels.

Table 2 shows the outcome of the experiment. The results for VFF and SVGP are from Hensman et al. (2017). We observe that VISH improves on the other methods in terms of NLPD and is within error bars in terms of MSE. Given the variability in the data the GP models improve when more data is available during training.

Given the dimensionality of the dataset, a full-VFF model is completely infeasible. As an example, using just four frequencies per dimension would already lead to $M = 4^8 = 65,536$ inducing variables. So VFF has to resort to an additive model with a prior covariance structure given as a sum of Matérn-3/2 kernels for each input dimension, as in eq. (7). Each of the functions f_d is approximated using 30 frequencies. We report two variants of VISH: one using all spherical harmonics up to degree 3 (M=210) and another up to degree 4 (M=660). As expected, the more inducing variables, the better the fit.

We also report the wall clock time for the experiments (training and evaluation) for $N = 10,000$ and $N = 5,929,413$. All these experiments were ran on a single consumer-grade GPU (Nvidia GTX 1070). On the complete dataset of almost 6 million records, VISH took 41 ± 0.81 seconds on average. A-VFF required 75.61 ± 0.75 seconds and the SVGP method needed approximately 15 minutes to fit and predict. This shows that VISH is roughly two orders of magnitude faster than SVGP. A-VFF comes close to VISH but has to impose additive structure to keep its computational advantage.

4.4. SUSY Classification

In the last experiment we tackle a large-scale classification problem. We are tasked with distinguishing between a signal process which produces super-symmetric (SUSY) particles and a background process which does not. The inputs consist of eight kinematic properties measured by the particle detectors in the accelerator. The dataset contains 5 million records of which we use the last 10% for testing. We are interested in obtaining a calibrated classifier and measure the AuC of the ROC curve.

For SVGP and VISH we first used a subset of 20,000 points to train the variational and hyper-parameters of the model with L-BFGS. We then applied Adam to the whole dataset. A similar approach was used to fine-tune the NN baselines by Baldi et al. (2014).

Table 3 lists the performance of VISH and compares it to a

		$N = 10,000$			$N = 100,000$		$N = 1,000,000$		$N = 5,929,413$		
Method	M	MSE	NLPD	Time	MSE	NLPD	MSE	NLPD	MSE	NLPD	Time
VISH	210	0.91 ± 0.16	1.328 ± 0.09	1.86 ± 0.38	0.826 ± 0.052	1.28 ± 0.03	0.84 ± 0.01	1.29 ± 0.01	0.833 ± 0.004	1.29 ± 0.002	41.32 ± 0.81
VISH	660	0.90 ± 0.16	$\mathbf{1.326 \pm 0.09}$	4.76 ± 1.25	0.808 ± 0.052	$\mathbf{1.27 \pm 0.03}$	0.83 ± 0.03	$\mathbf{1.28 \pm 0.01}$	0.834 ± 0.055	$\mathbf{1.27 \pm 0.002}$	160.8 ± 3.80
A-VFF	30/dim.	$\mathbf{0.89 \pm 0.15}$	1.362 ± 0.09	6.78 ± 0.85	0.819 ± 0.05	1.32 ± 0.03	0.83 ± 0.01	1.33 ± 0.03	0.827 ± 0.004	1.32 ± 0.007	75.61 ± 0.75
SVGP	500	0.90 ± 0.16	1.358 ± 0.09	836.54 ± 0.78	$\mathbf{0.808 \pm 0.05}$	1.31 ± 0.03	$\mathbf{0.82 \pm 0.01}$	1.32 ± 0.002	$\mathbf{0.814 \pm 0.004}$	1.31 ± 0.002	918.77 ± 1.21

Table 2. Predictive mean squared errors (MSEs), negative log predictive densities (NLPDs) and wall-clock time in seconds with one standard deviation based on 10 random splits on the airline arrival delays experiment. Total dataset size is given by N and in each split we randomly select 2/3 and 1/3 for training and testing.

Method	AuC
BDT	0.850 ± 0.003
NN	$\mathbf{0.867 \pm 0.002}$
$\text{NN}_{\text{dropout}}$	0.856 ± 0.001
SVGP (SE)	0.852 ± 0.002
VISH	0.859 ± 0.001

Table 3. Performance comparison for the SUSY benchmark. The mean AuC is reported with one standard deviation, computed by training five models with different initialisations. Larger is better. Results for BDT and NN are from Baldi et al. (2014).

boosted decision tree (BDT), 5-layer neural network (NN), and a SVGP. We observe the competitive performance of VISH, which is a single-layer GP method, compared to a 5-layer neural net with 300 hidden units per layer and extensive hyper-parameter optimisation (Baldi et al., 2014). We also note the improvement over a SVGP with SE kernel.

5. Conclusion

We introduced a framework for performing variational inference in Gaussian processes using spherical harmonics. Our general setup is closely related to VFF, and we inherit several of its advantages such as a considerable speed-up compared to classic sparse GP models, and having features with a global influence on the approximation. By projecting the data onto the hypersphere and using dedicated GP models on this manifold our approach succeeds where other sparse GPs methods fail. First, VISH provides good scaling properties as the dimension of the input space increases. Second, we showed that under some relatively weak hypothesis we are able to select the optimal features to include in the approximation. This is due to the intricate link between "stationary" covariances on the sphere and the Laplace-Beltrami operator. Third, the Mercer representation of the kernel means that the matrices to be inverted at training time are exactly diagonal – resulting in a very cheap-to-compute sparse approximate GP.

We showed on a wide range of regression and classification problems that our method performs at or close to the state of the art while being extremely fast. The good predictive performance may be inherited from the connection between infinitely wide neural network and the way we map the predictions on the sphere back to the original space. Future work will explore this hypothesis.

Acknowledgements

Thanks to Fergus Simpson for pointing us in the direction of spherical harmonics in the early days of this work. Also many thanks to Arno Solin for sharing his implementation of the Matérn kernels' power spectrum.

References

Baldi, P., Sadowski, P., and Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5:4308, 2014.

Cho, Y. and Saul, L. K. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems*, pp. 342–350, 2009.

Dai, F. and Xu, Y. *Approximation theory and harmonic analysis on spheres and balls*. Springer, 2013.

Frigola, R., Chen, Y., and Rasmussen, C. E. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems*, pp. 3680–3688, 2014.

Hensman, J. and Lawrence, N. D. Nested variational compression in deep Gaussian processes. *arXiv preprint arXiv:1412.1370*, 2014.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Advances in Uncertainty in Artificial Intelligence*, 2013.

Hensman, J., Matthews, A. G. d. G., and Ghahramani, Z. Scalable variational Gaussian process classification. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2015.

Hensman, J., Durrande, N., and Solin, A. Variational fourier features for Gaussian processes. *Journal of Machine Learning Research*, 18:151–1, 2017.

Kanagawa, M., Hennig, P., Sejdinovic, D., and Sriperumbudur, B. K. Gaussian processes and kernel methods: A

review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

Lázaro-Gredilla, M. and Figueiras-Vidal, A. Inter-domain Gaussian processes for sparse inference using inducing features. In *Advances in Neural Information Processing Systems*, 2009.

Peacock, J. A. *Cosmological physics*. Cambridge University Press, 1999.

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

Rasmussen, C. E. and Williams, C. K. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Salimbeni, H. and Deisenroth, M. P. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, 2017.

Shi, J., Titsias, M., and Mnih, A. Sparse orthogonal variational inference for Gaussian processes. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, pp. 1932–1942, 2020.

Solin, A. and Särkkä, S. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, pp. 1–28, 2014.

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2009.

Titsias, M. and Lawrence, N. D. Bayesian Gaussian process latent variable model. In *Proceedings of International Conference on Artificial Intelligence and Statistics*, 2010.

van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. A framework for interdomain and multioutput Gaussian processes. *arXiv:2003.01115*, 2020. URL https://arxiv.org/abs/2003.01115.

Wendland, H. *Scattered data approximation*. Cambridge University Press, 2005.

Williams, C. K. Computation with infinite neural networks. *Neural Computation*, 10(5):1203–1216, 1998.