Spherical Gaussian Processes

Vincent Dutordoir

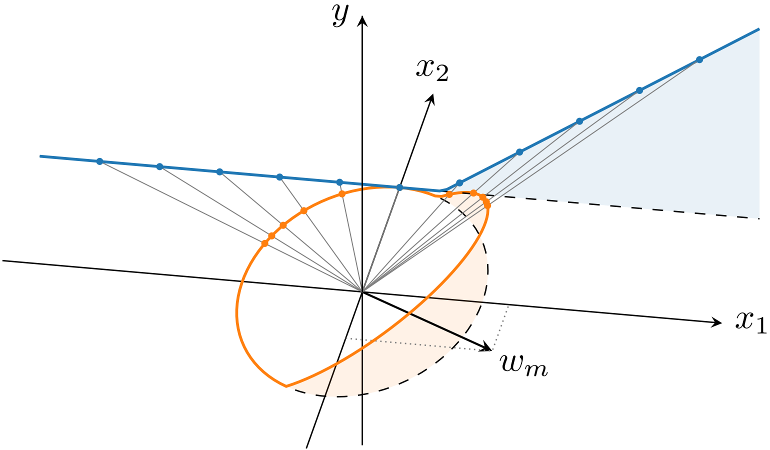
CBL - Tea Talk

Why do we care about spherical Gaussian processes?

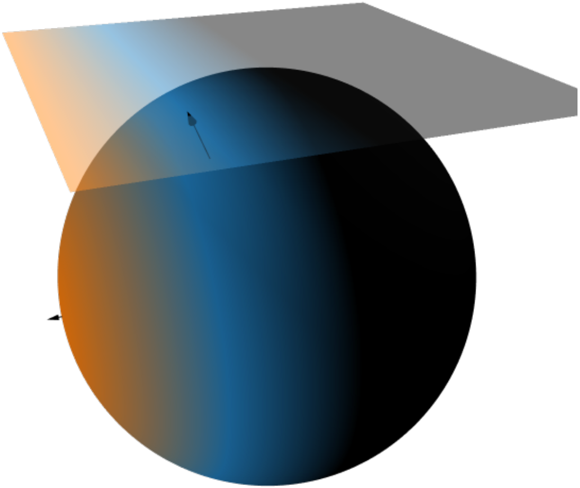
$$
\begin{aligned}
\operatorname{ReLU}(\boldsymbol{w}^\top \boldsymbol{x} + \mathrm{b}) &= \operatorname{ReLU}\Big(\begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{x} \\ \mathrm{b} \end{bmatrix} \Big) \\
&= \operatorname{ReLU}\big(\|\tilde{\boldsymbol{w}}\| \, \|\tilde{\boldsymbol{x}}\| \cos(\theta) \big) \\
&= \|\tilde{\boldsymbol{w}}\| \, \|\tilde{\boldsymbol{x}}\| \operatorname{ReLU}\big(\cos(\theta) \big),
\end{aligned}
$$

where $\tilde{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w} \\ 1 \end{bmatrix}$, $\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{x} \\ \mathrm{b} \end{bmatrix}$, and θ is the angle between $\tilde{\boldsymbol{w}}$ and $\tilde{\boldsymbol{x}}$.

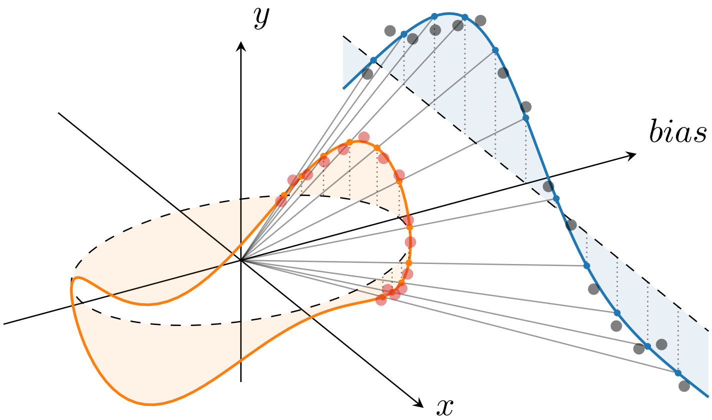
ReLU on sphere and in data plane in 2D



ReLU on sphere and in data plane in 3D

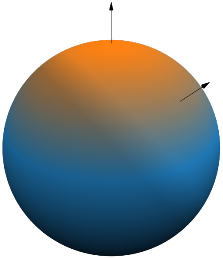


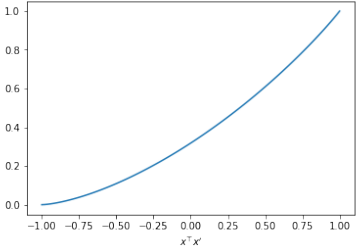
Model on the sphere (circle) and linearly project



Mercer Decomposition of Zonal Kernels

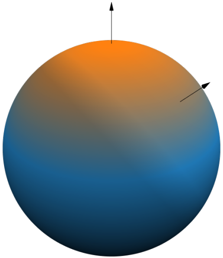
$$\kappa(\cos(\theta)) = \sin(\theta) + (\pi - \theta)\sin(\theta)$$



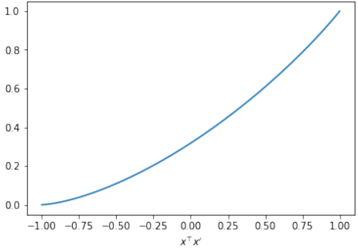


- Spherical counterpart of stationary kernels: $k(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}^\top \mathbf{x}')$.

Mercer Decomposition of Zonal Kernels

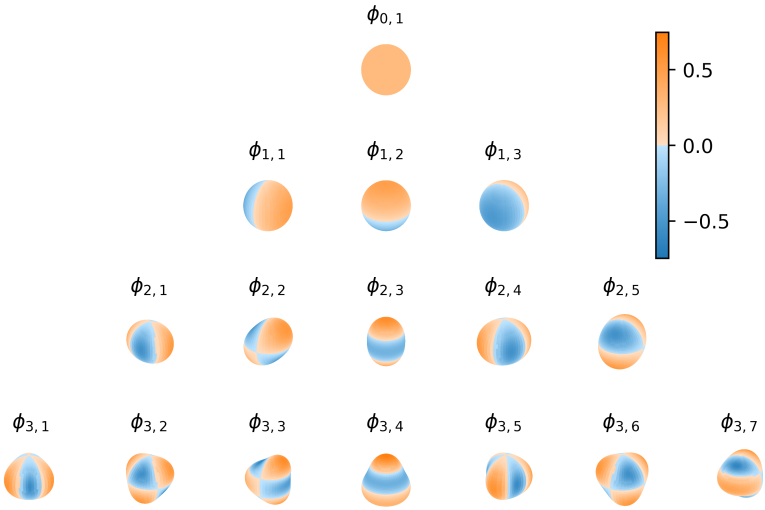


$$\kappa(\cos(\theta)) = \sin(\theta) + (\pi - \theta)\sin(\theta)$$



- Spherical counterpart of stationary kernels: $k(\mathbf{x}, \mathbf{x}') = \kappa(\mathbf{x}^\top \mathbf{x}')$.
- Mercer's decomposition: $\kappa(\mathbf{x}^\top \mathbf{x}') = \sum_{n=0}^{\infty} \sum_{k=1}^{N_n} \lambda_n \, \phi_{n,k}(\mathbf{x}) \, \phi_{n,k}(\mathbf{x}')$.

Spherical Harmonics



Reproducing Kernel Hilbert Space (RKHS)

Define the RKHS through the Mercer decomposition

$$\mathcal{H}_k = \left\{ f(\cdot) = \sum_{n,k} \hat{f}_{n,k} \phi_{n,k}(\cdot) : \|f\|_{\mathcal{H}_k} < \infty \right\}$$

Reproducing Kernel Hilbert Space (RKHS)

Define the RKHS through the Mercer decomposition

$$\mathcal{H}_k = \left\{ f(\cdot) = \sum_{n,k} \hat{f}_{n,k} \phi_{n,k}(\cdot) : \|f\|_{\mathcal{H}_k} < \infty \right\}$$

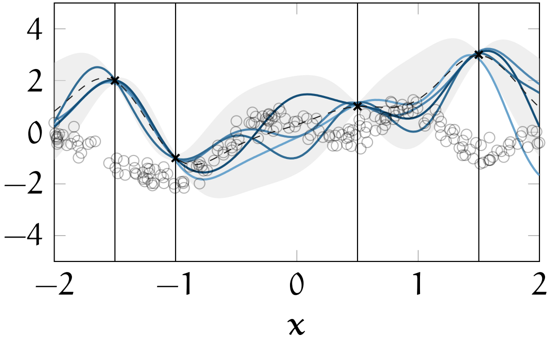
with inner-product:

$$\langle g(\cdot), h(\cdot) \rangle_{\mathcal{H}_k} = \sum_{n,k} \frac{\hat{g}_{n,k} \hat{h}_{n,k}}{\lambda_n},$$

such that (reproducing property):

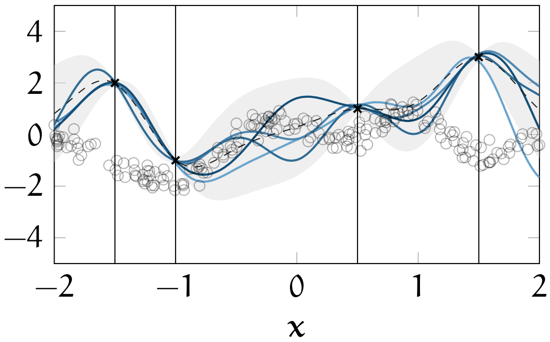
$$\langle g(\cdot), k(\boldsymbol{x}, \cdot) \rangle_{\mathcal{H}_k} = g(\boldsymbol{x}).$$

Sparse GPs with interdomain Inducing Variables



$$f(\cdot) \mid f(\mathbf{Z}) = \mathbf{u} \tag{1}$$

Sparse GPs with interdomain Inducing Variables



$$f(\cdot) \mid f(\mathbf{Z}) = \mathbf{u} \tag{1}$$

Linear transformation of the GP

$$\mathbf{u} = f(z) \quad \rightarrow \quad \mathbf{u} = \int f(x) g(x) dx$$

Spherical harmonics Inducing Features

Approximate posterior constructed out of inducing features

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

Inducing points	**Inducing Features**
$u_m = f(z_m)$	$u_m = \langle f, \phi_m \rangle_{\mathcal{H}_k}$
$\mathrm{Cov}(u_m, f(\cdot)) = k(\cdot, z_m)$	$\mathrm{Cov}(u_m, f(\cdot)) = \phi_m(\cdot)$
$\mathrm{Cov}(u_m, u_{m'}) = k(z_m, z_{m'})$	$\mathrm{Cov}(u_m, u_{m'}) = \lambda_m \delta_{mm'}$

Approximate Posterior

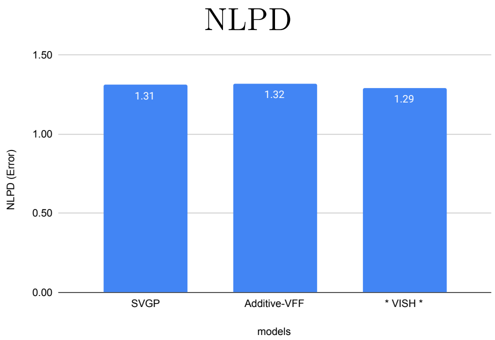
$$q_f = \mathcal{GP}\left(\sum \alpha_m k(z_m, \cdot); \quad k(\cdot, \cdot) - \mathbf{k}_{\cdot\mathbf{u}} \mathbf{K}_{\mathbf{uu}}^{-1} (\mathbf{K}_{\mathbf{uu}} - \boldsymbol{\Sigma}) \mathbf{K}_{\mathbf{uu}}^{-1} \mathbf{k}_{\mathbf{u}\cdot}\right)$$

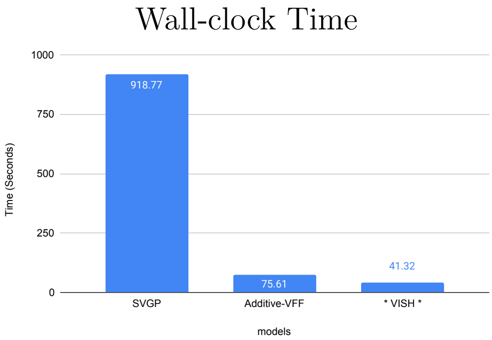
$$\Downarrow$$

$$q_f = \mathcal{GP}\left(\sum \alpha_m \phi_m(\cdot); \quad k(\cdot, \cdot') - \Phi^\top(\cdot)(\Lambda - \boldsymbol{\Sigma})\Phi(\cdot')\right),$$

Experiment

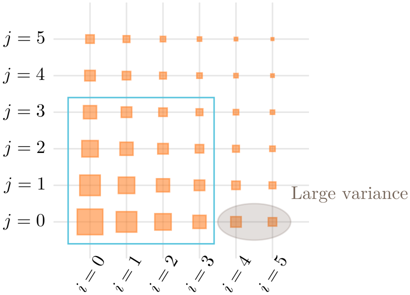
Airline dataset: 6,000,000 datapoints regression task fitted in 40 seconds on a single ‘cheap’ GTX 1070 GPU



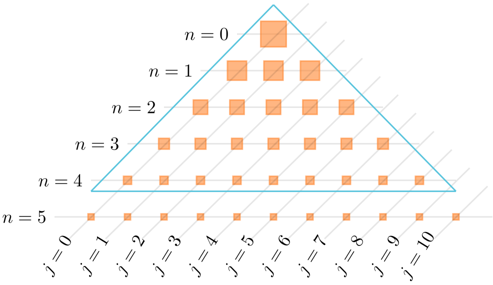


Optimal inducing features

VFF



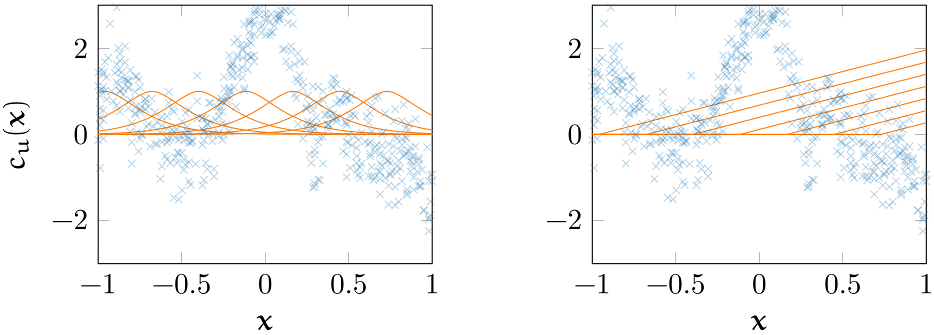
VISH



Basis functions

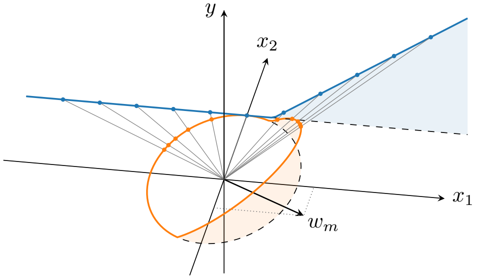
Traditionally, the inducing variables are defined as $\mathrm{u} = \mathrm{f}(\boldsymbol{z})$, which leads to the basis functions:

$$\mathrm{k}_{\mathrm{uf}} = \mathrm{Cov}(\mathrm{f}(\boldsymbol{z}), \mathrm{f}(\boldsymbol{x})) = \mathrm{k}(\boldsymbol{z}, \boldsymbol{x}) \tag{2}$$



One step further

From spherical harmonic basis function to "Activated" features



$$g_m(\boldsymbol{x}) = \|\boldsymbol{x}\| \, \|\boldsymbol{w}_m\| \, \mathrm{ReLU}\left(\frac{\boldsymbol{w}_m^\top \boldsymbol{x}}{\|\boldsymbol{w}_m\| \, \|\boldsymbol{x}\|}\right)$$

Before:

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

$$K_{uf} = \mathrm{Cov}(u_m, f(\cdot)) = \phi_m(\cdot)$$

Now:

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

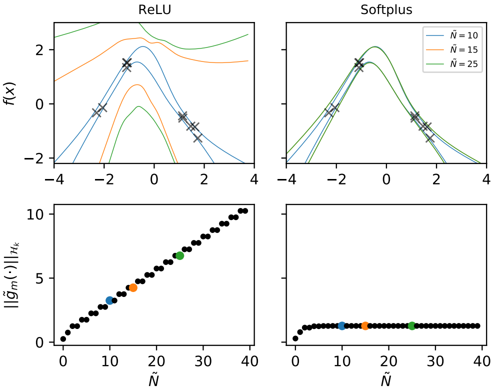
$$K_{uf} = \mathrm{Cov}(u_m, f(\cdot)) = g_m(\cdot)$$

Caveat

We require inducing variables $u = \langle f(\cdot), g(\cdot)\rangle_{\mathcal{H}_k}$ with finite variance

$$\mathrm{Var}(u) = \|g(\cdot)\|_{\mathcal{H}_k} = \sum_i \frac{\hat{g}_i^2}{\lambda_i} < \infty$$

where $\hat{g}_i$ are the coefficients of g when projected on the kernel eigen function basis of the kernel. **This series may diverge!**

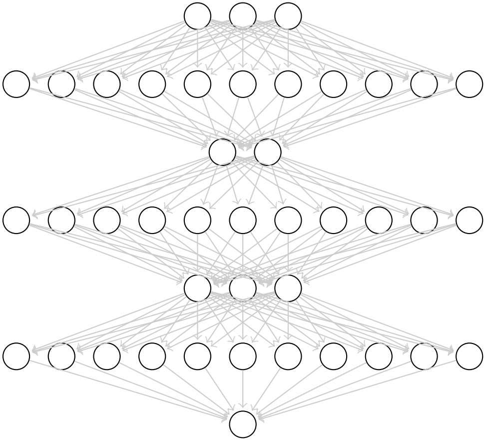


Deep Gaussian Processes

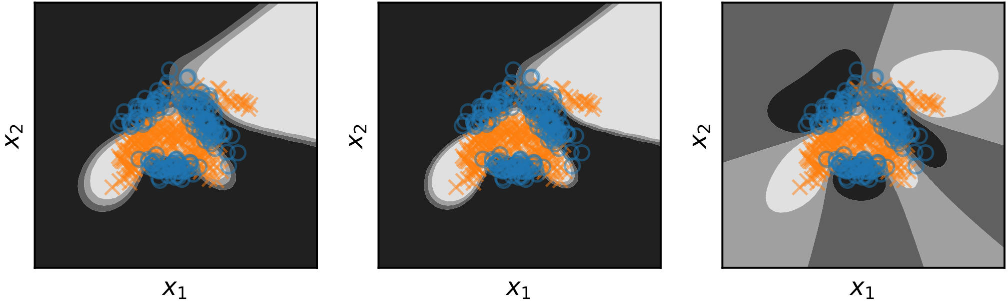
- GP inducing variables for which the basis functions behave like NN activations.

- For such construction, propagating through the mean of each GP layer is exactly a forward pass in a NN!

- Benefits:
    1. Initialise a DGP with a NN point estimate
    2. Add uncertainty to a DNN



Toy experiment: 3 layer DGP



Take-Away Messages

1. Interdomain Gaussian processes are a great framework for building powerful models.
2. Gaussian processes <3 Spherical Harmonics: leads to diagonal covariance matrices!
3. ReLU-like basis functions need to be handled with care, but give rise to approximate Deep GPs for which propagating the mean is equivalent to a DNN.

Thank you for your attention

References

- Mark van der Wilk, Vincent Dutordoir, ST John, Artem Artemev, Vincent Adam, and James Hensman. A Framework for Interdomain and Multioutput Gaussian Processes. Arxiv.

- Vincent Dutordoir, Nicolas Durrande, and James Hensman. "Sparse Gaussian Processes with Spherical Harmonic Features". ICML 2020.

- Vincent Dutordoir, James Hensman, Mark van der Wilk, Carl Henrik Ek, Zoubin Ghahramani, and Nicolas Durrande. "Deep Neural Networks as Point Estimates for Deep Gaussian Processes". NeurIPS 2021.

- Code: `https://github.com/vdutor/SphericalHarmonics`