An Overview of Sparse Variational Gaussian Processes

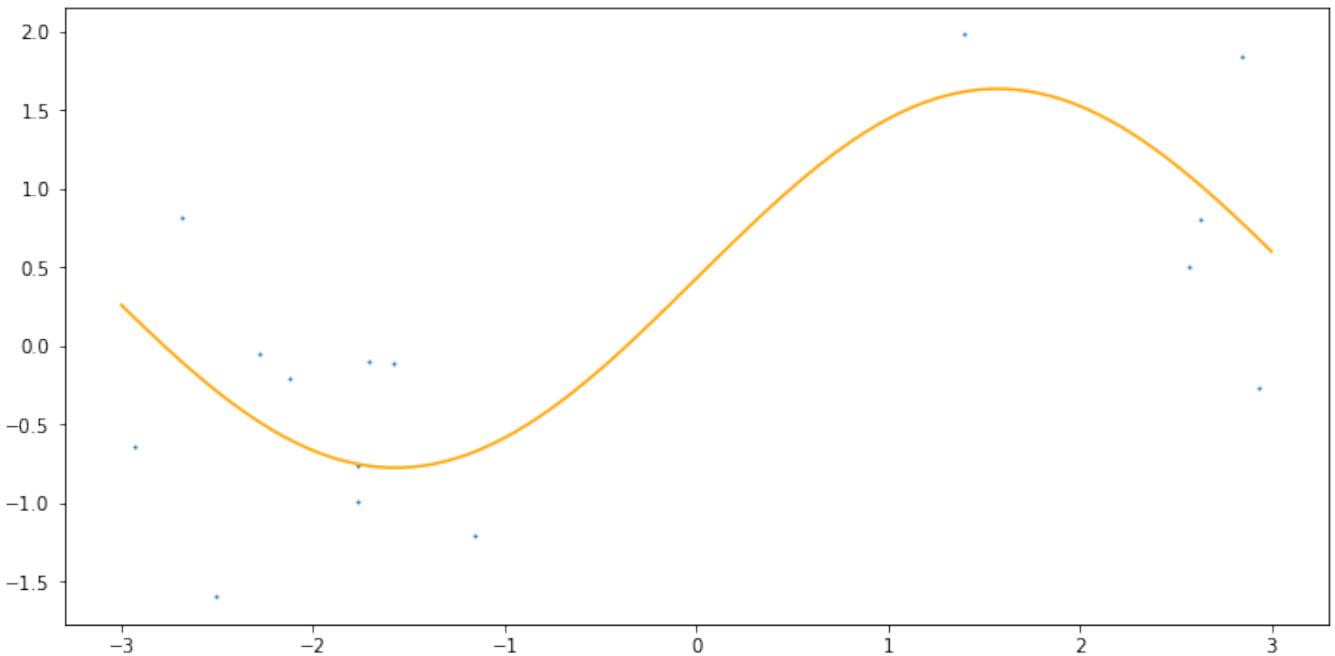
Harrison Zhu[1]

[1]Imperial College London

November 12, 2021

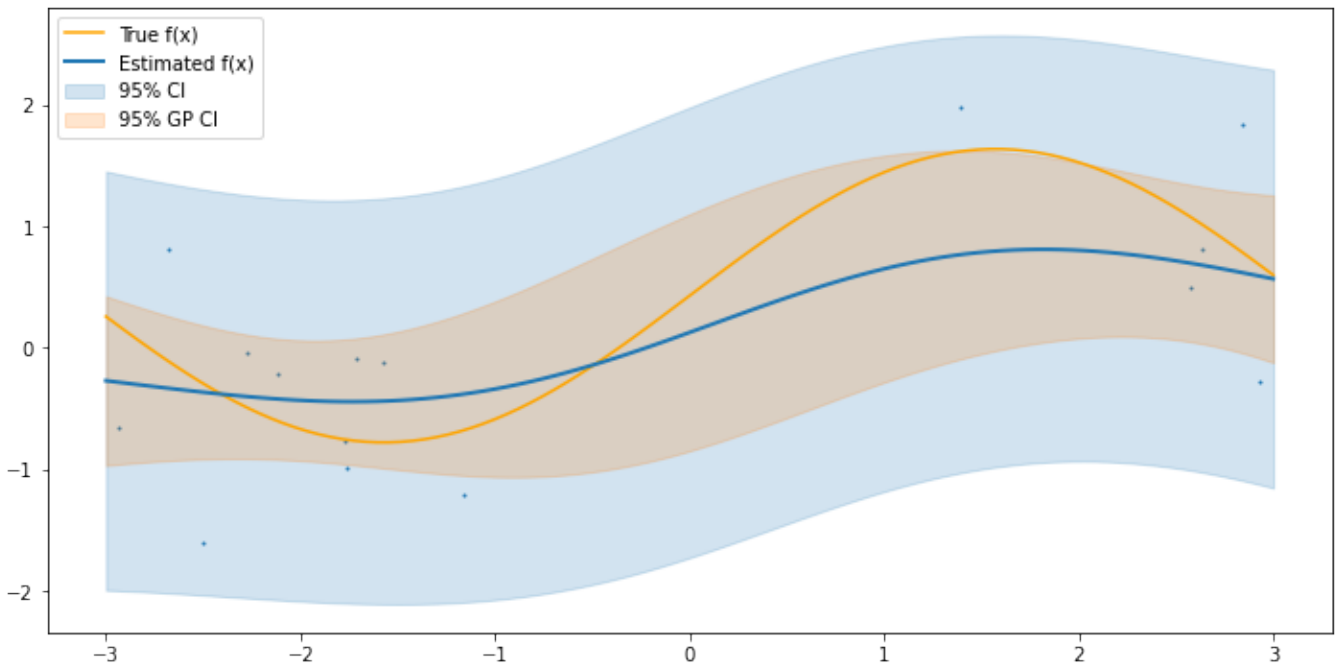
Motivation

Suppose there is some unknown function $f(x, z)$, where x is a feature and z is a source of randomness e.g.
$f(x, z) = f_x(x) + f_z(z) = 2x + \epsilon(z)$ with $\epsilon(z) \sim \mathcal{N}(0, \sigma^2)$ for all z.
Task: Approximate/emulate f, given some signal about f e.g. direct queries of or derivatives of f. Usually focus on f_x and use something simple for f_z.



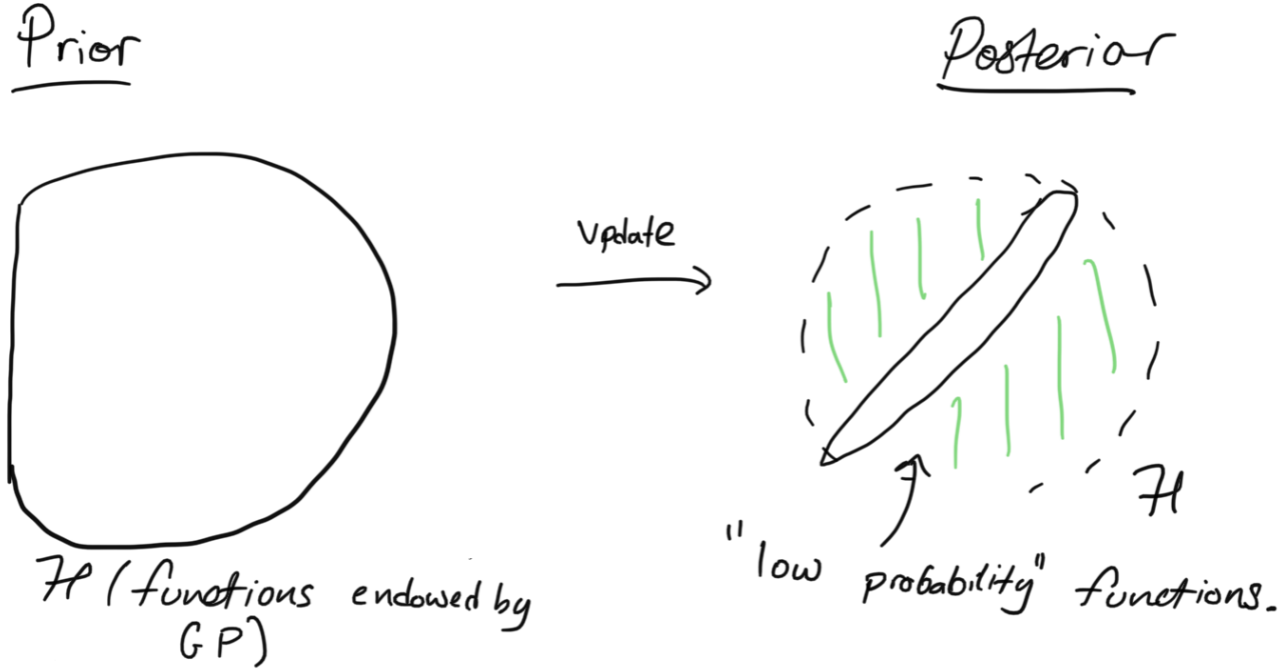
Motivation

Task: Approximate/emulate f, given some signal about f e.g. direct queries of or derivatives of f.
We can use various classes of function approximators: linear models, neural networks etc... or Gaussian Processes



Motivation

GPs can cover a wide range of function classes and flexibly identify, probabilistically, which ones are "good approximations" - it represents a sequence of random variables - in a Bayesian updating procedure.



Gaussian Processes

A GP is $f \sim \mathcal{GP}(\mu, k)$, with mean function μ and covariance/kernel function k such that

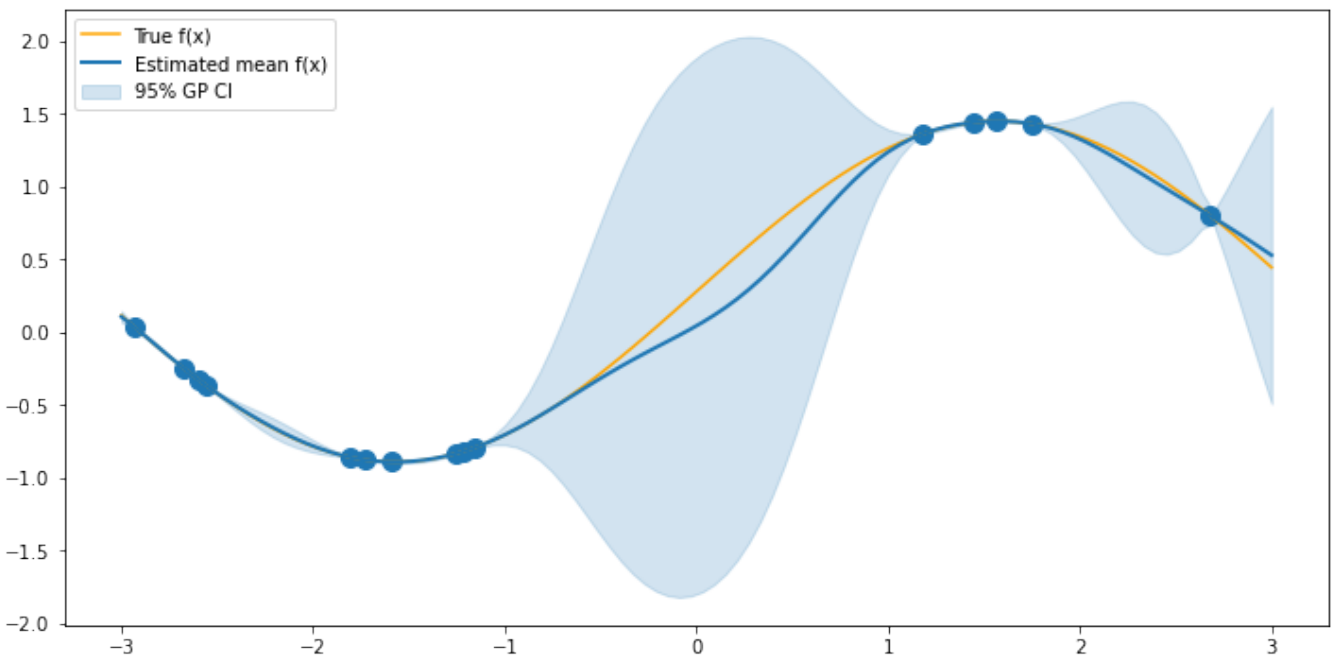
1. $\mu(x) := \mathbb{E}[f(x)]$ for all $x \in \mathcal{X}$,
2. $k(x, x') := \mathrm{Cov}(f(x), f(x')) = \mathbb{E}[(f(x) - \mu(x))(f(x') - \mu(x'))]$ for all $x, x' \in \mathcal{X}$.
3. Given a finite subset $X := (x_1, \ldots, x_n)^\intercal \in \mathbb{R}^{n \times d}$, $(f(x_1), \ldots, f(x_n))^\intercal \sim \mathcal{N}(\mu_X, k_{XX})$, where $\mu_X \equiv \mu(X) := (\mu(x_1), \ldots, \mu(x_n))^\intercal$ and $k_{XX} = k(X, X) \in \mathbb{R}^{n \times n}$ such that $(k_{XX})_{ij} = k(x_i, x_j)$. We will also use the shorthand $k_{x,X} := k_{X,x}^\intercal \in \mathbb{R}^{n \times 1}$ with $(k_{X,x})_i = k(x_i, x)$.

The expectation $\mathbb{E}$ is over f!

Gaussian Processes Regression: Noise-free

$f|X, y \sim \mathcal{GP}(\tilde{\mu}, \tilde{k})$ with

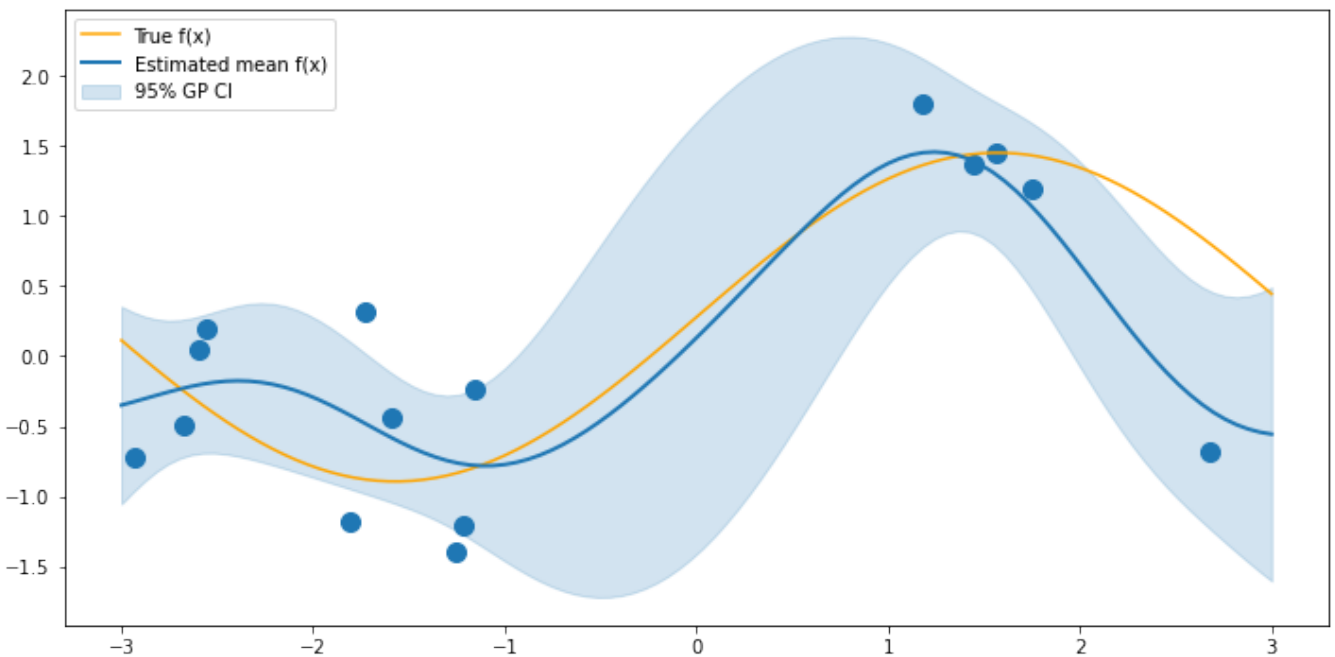
$$\tilde{\mu}(x) := \mu(x) + k_{xX} k_{XX}^{-1}(y - \mu_X),$$
$$\tilde{k}(x, x') := k(x, x') - k_{xX} k_{XX}^{-1} k_{Xx'},$$



Gaussian Processes Regression: Noisy

$f|X, y \sim \mathcal{GP}(\tilde{\mu}, \tilde{k})$ with

$$\tilde{\mu}(x) := \mu(x) + k_{xX}(k_{XX}^{-1} + \sigma^2 I_n)(y - \mu_X),$$
$$\tilde{k}(x, x') := k(x, x') - k_{xX}(k_{XX}^{-1} + \sigma^2 I_n)k_{Xx'},$$



Maximum Likelihood

MLE estimator:

$$\theta_* = \text{argmax}_\theta N(y; \mu(X), k(X,X) + \hat{\sigma}^2 I_n)$$

where θ represents the kernel hyperparameters and $\hat{\sigma}$. But this involves $\mathcal{O}(n^3)$ complexity for inversion of $k(X,X)$ and $\mathcal{O}(n^2)$ storage for $k(X,X)$. In addition, the non-convex optimisation problem may be difficult to solve.

Markov Chain Monte Carlo

Similarly, use a prior $p(\theta)$ for θ, and obtain a samples of the posterior distribution $p(\theta|\text{data}) \propto p(y|X,\theta)p(\theta)$. However, note that to compute $p(y|X,\theta)$ we still need deal with the expensive operations involving $k(X,X)$.

Variational Inference

Given a target distribution p, VI seeks to construct an approximate distribution q_* such that

$$q_* := \text{argmin}_{q \in \mathcal{Q}} \text{KL}(q||p) = \text{argmin}_{q \in \mathcal{Q}} \int_{\mathcal{X}} \log \frac{q(x)}{p(x)} q(x) \text{d}x,$$

where $\mathcal{Q}$ is a family of variational distributions that is user-defined.

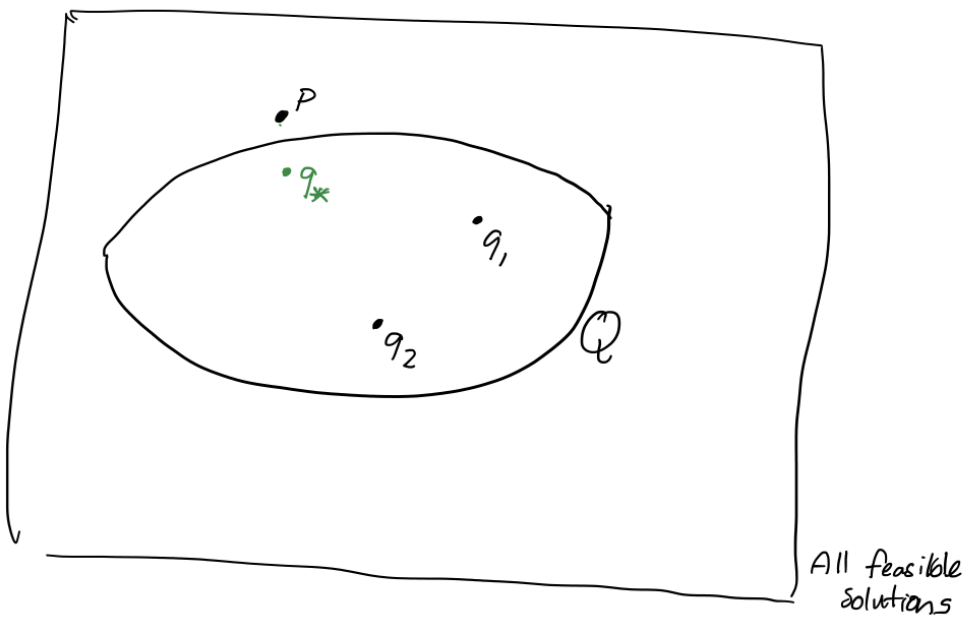


Figure: Illustration of the variational approximation from $\mathcal{Q}$.

Sparse Variational Gaussian Processes

Compress all information onto $u = (f(z_1), \ldots, f(z_m))$, where $\{z_1, \ldots, z_m\} \subset \mathcal{X}$ and $m << n$. Approximate $p(f, u|y)$ using a variational approximation $q(f, u) := p(f|u)q(u)$. We construct $q(u)$ such that

$$\begin{aligned} q(u) &= \text{argmin}_{q \in \mathcal{Q}} \text{KL}(p(f, u|y)||q(f, u)) \\ &= \text{argmin}_{q \in \mathcal{Q}} \text{KL}(q(u)||p(u)) - \sum_{i=1}^{n} \mathbb{E}_{p(f_i|u)q(u)}[\log p(y_i|f_i)], \\ &=: \text{argmin}_{q \in \mathcal{Q}} - \text{ELBO}(q) \\ &= \text{argmax}_{q \in \mathcal{Q}} \text{ELBO}(q) \end{aligned}$$

SGPR [Titsias, 2009]

This algorithm has per-iteration complexity $\mathcal{O}(nm^2 + m^3)$. Assuming that $q(u) = N(m_u, S_u)$, the optimal posterior could be computed analytically [Titsias, 2009], giving

$$\begin{aligned}
\text{ELBO}(q) &= \log N(y; 0, Q_{ff} + \sigma^2 I) - \tfrac{1}{2\sigma^2}\text{Tr}(k_{XX} - Q_{ff}), \\
q(u) &= N(m_u, S_u), \\
S_u &= k_{ZZ}^{-1} + k_{ZZ}^{-1} k_{ZX} k_{XZ} k_{ZZ}^{-1} \sigma^2, \\
m_u &= \sigma^2 S_u^{-1} k_{ZZ}^{-1} k_{ZX} y.
\end{aligned}$$

Therefore it only remains to optimise over the kernel hyperparameters and σ^2 using gradient-based optimisation.

SGPR [Titsias, 2009]

To perform prediction at a new point $x \in \mathcal{X}$, we can obtain a posterior distribution

$$
\begin{aligned}
q(f(x)) := \int p(f(x)|u)q(u)\mathrm{d}u = N(f(x); \tilde{\mu}(x), \tilde{v}(x)),\\
\tilde{\mu}(x) =& k_{xZ}k_{ZZ}^{-1}m_u,\\
\tilde{v}(x) =& k_{xx} - k_{xZ}(k_{ZZ}^{-1} + k_{ZZ}^{-1}S_u k_{ZZ}^{-1})k_{xZ},
\end{aligned}
$$

Caution:

- SGPR is only possible for iid Gaussian noise regression problems
- May not be computationally feasible when n (e.g. label-rich datasets) or m (e.g. spatiotemporal datasets where the number of inducing points explode) are very large.

SVGP [Hensman et al., 2015]

This algorithm has per-iteration complexity $\mathcal{O}(n_b m^2 + m^3)$. We only take a minibatch n_b of the training points and do not use the optimal posterior $q(u)$.

$$\begin{aligned}
&\sum_{i=1}^{n} \mathbb{E}_{p(f_i|u)q(u)}[\log p(y_i|f_i)] - \mathrm{KL}(q(u)||p(u)) \\
&= \mathbb{E}_B[\tfrac{n}{n_b} \sum_{B\in B} \mathbb{E}_{p(f_i|u)q(u)}[\log p(y_i|f_i)]] - \mathrm{KL}(q(u)||p(u)), \\
&\approx \tfrac{1}{L} \sum_{b=1}^{L} \tfrac{n}{n_b} \sum_{i\in B_b} \mathbb{E}_{p(f_i|u)q(u)}[\log p(y_i|f_i)] - \mathrm{KL}(q(u)||p(u)),
\end{aligned}$$

where $|B| = |B_b| = n_b << n$ with minibatches B or B_b.

Caution:

- May overestimate likelihood variance in practice [Bauer et al., 2016].
- Need to estimate θ and $q(u)$ in coupled diffusion-like optimisation procedure.

Natural Gradients [Adam et al., 2021, Salimbeni et al., 2018]

Naively, one can simply perform SGD over the Euclidean space for the mean and variance of $q(u)$: $\xi = (m_u, S_u)$. But one can also perform optimisation using the geometry of $\mathcal{Q}$:

$$\eta_t^{k+1} \leftarrow \eta_t^k + \rho_k \tilde{\nabla}_\xi \text{ELBO}(\eta_t^k, \theta_t),$$

where $\tilde{\nabla}_\xi := F(\xi)^{-1} \nabla_\xi$, where $F(\xi)$ is the Fisher information matrix of $q(u; \xi)$. One can identify a statistical manifold (Riemannian) with metric tensor being $F(\xi)$ and it can be shown that $\tilde{\nabla}_\xi \text{ELBO}(\xi, \theta)$ is the Riemannian gradient if we pose ξ as lying on the statistical manifold and perform Riemannian gradient descent.

Demo

https://colab.research.google.com/drive/14yNYE06xTE2hb5Y9npB2QmD9YTH5EUx5?usp=sharing

Interdomain Inducing Points [Leibfried et al., 2021, van der Wilk et al., 2020]

We previously set the inducing points as $u = f(z)$, we could also encoded more generally

$$\mathcal{L}f(\cdot) = \int_{\mathcal{X}} f(x)\phi(x)\mathrm{d}x,$$

where $\phi(x)$ are "inducing features". Since f is random with sample distribution $\mathbb{P}$, we have the prior $u \sim \mathcal{N}(\mu_u, k_{uu})$ with

$$(\mu_u)_i = \mathbb{E}_{\mathbb{P}}[u] = \int_{\mathcal{X}} \mathbb{E}_{\mathbb{P}}[f(x)]\phi_i(x)\mathrm{d}x = \int_{\mathcal{X}} \mu(x)\phi_i(x)\mathrm{d}x,$$

$$k(u_i, u_j) = \mathbb{E}_{\mathbb{P}}[(u_i - (\mu_u)_i)(u_j - (\mu_u)_j)] = \int_{\mathcal{X}} \int_{\mathcal{X}} k(x, x')\phi_i(x)\phi_j(x')\mathrm{d}x\mathrm{d}x'$$

In addition, given x or $f(x)$, we have

$$k(f(x), u_i) = \mathbb{E}_{\mathbb{P}}[(f(x) - \mu(x))(u_j - (\mu_u)_j)] = \int k(x, x')\phi_i(x')\mathrm{d}x'.$$

Given these 3 terms, we can now fully specify the posterior distribution $q(f(x))$.

Deep Gaussian Processes (DGP; Salimbeni and Deisenroth [2017])

A deep Gaussian process is $f(x) = f_L \circ \cdot \circ f_1(x)$ such that each layer is a GP. We can see that this notation is not very well-defined, but the idea is to propagate samples of $f_1(x), \ldots, f_{L-1}(x)$ for each layer. Inducing points are used to define the DGP prior for each layer so that they are "input-dependent". Due to the multi-layered and intractable nature of the DGP prior, sparse variational GP techniques are often used to perform inference.

Latent Variable Gaussian Processes [Dutordoir et al., 2018]

Let $w_i \sim \mathcal{N}(0, 1)$ be independent for $i = 1, \ldots, n$. Then define a latent variable GP (LVGP) as $f(x_i, w_i) \sim \mathcal{GP}(\mu, k)$, where μ and k operate over the tuple (x_i, w_i), with w_i being a sample here. Then the marginal likelihood of this model is

$$p(y|X) = \int N(y; \mu(X, W), k((X, W), (X, W)))\mathrm{d}W,$$

where W a stacked version of the w_i's. Due to the intractable nature of this integral and the fact that we require kernel matrix operations, sparse variational Gaussian process methods have to be used to perform inference.

Many more applications

- State space models [Wilkinson et al., 2021]
- Variational Gaussian Processes (VGP; [Opper and Archambeau, 2009])
- Stein Gaussian Processes [Pinder et al., 2020]
- etc...

Conclusion

- Sparse variational approximations to GPs allow for computational tractability and flexibility for intractable posteriors
- Much work still in progress for inducing points formulation and inference techniques
- Scalable software available already, but more work still to come

Spontaneously updated notes here: `https://harrisonzhu508.github.io/pdfs/intro_to_gps.pdf`
Thank you!

Vincent Adam, Paul E. Chang, Mohammad Emtiyaz Khan, and Arno Solin. Dual Parameterization of Sparse Variational Gaussian Processes, 2021. _eprint: 2111.03412.

Matthias Bauer, Mark van der Wilk, and Carl Edward Rasmussen. Understanding probabilistic sparse Gaussian process approximations. In *Advances in neural information processing systems*, pages 1533–1541, 2016.

Vincent Dutordoir, Hugh Salimbeni, Marc Deisenroth, and James Hensman. Gaussian Process Conditional Density Estimation. *arXiv:1810.12750 [cs, stat]*, October 2018. URL `http://arxiv.org/abs/1810.12750`. arXiv: 1810.12750.

James Hensman, Alexander Matthews, and Zoubin Ghahramani. Scalable variational Gaussian process classification. In *Artificial Intelligence and Statistics*, pages 351–360. PMLR, 2015.

Felix Leibfried, Vincent Dutordoir, S. T. John, and Nicolas Durrande. A Tutorial on Sparse Gaussian Processes and Variational Inference. *arXiv:2012.13962 [cs, stat]*, July 2021. URL `http://arxiv.org/abs/2012.13962`. arXiv: 2012.13962.

Manfred Opper and Cédric Archambeau. The variational Gaussian approximation revisited. *Neural computation*, 21(3):786–792, 2009. Publisher: MIT Press.

Thomas Pinder, Christopher Nemeth, and David Leslie. Stein Variational Gaussian Processes. *arXiv preprint arXiv:2009.12141*, 2020.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. *arXiv preprint arXiv:1705.08933*, 2017.

Hugh Salimbeni, Stefanos Eleftheriadis, and James Hensman. Natural gradients in practice: Non-conjugate variational inference in Gaussian process models. In *International Conference on Artificial Intelligence and Statistics*, pages 689–697. PMLR, 2018.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

Mark van der Wilk, Vincent Dutordoir, S. T. John, Artem Artemev, Vincent Adam, and James Hensman. A Framework for Interdomain and Multioutput Gaussian Processes. *arXiv:2003.01115 [cs, stat]*, March 2020. URL http://arxiv.org/abs/2003.01115. arXiv: 2003.01115.

William J. Wilkinson, Arno Solin, and Vincent Adam. Sparse Algorithms for Markovian Gaussian Processes. *arXiv:2103.10710 [cs, stat]*, June 2021. URL http://arxiv.org/abs/2103.10710. arXiv: 2103.10710.