Deep Gaussian Processes

Adam Buckley

February 2026

Presentation Outline

1. **Gaussian Processes (GPs)**
   - Deriving GPs
   - Defining Gaussian Processes
   - Advantages and limitations of GPs
2. **Deep GPs (DGPs)**
   - Why GP limitations motivate DGPs
   - Architecture & composition
   - Advantages and limitations of DGPs
3. **Training GPs & DGPs**
   - Variational inference
   - Inducing points
4. **Credit Risk Modelling**
   - What is credit risk?
   - Why GPs/DGPs?
   - Results & comparisons
   - Key takeaways

Towards Gaussian Processes

Goal: For $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, we aim to find $f : \mathbb{R}^D \to \mathbb{R}$ such that $f(\mathbf{x}_i) \approx y_i$.

Linear regression: $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$. Too restrictive. Assumes linearity.

Extend with basis functions [1]:

$$f(\mathbf{x}) = \sum_{j=1}^{M} w_j \phi_j(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w} \tag{1}$$

- More flexible (non-linear through basis choice)
- Still need to choose fixed number and form of $\phi_j(\cdot)$

From Basis Functions to Gaussian Processes

If we take a baysian approach, and place a Gaussian prior over weights [5, 8]

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \Sigma_w) \tag{2}$$

Then the function values at any finite set of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ are jointly Gaussian:

$$\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \quad \text{where } K_{ij} = \phi(\mathbf{x}_i)^\top \Sigma_w \phi(\mathbf{x}_j) \tag{3}$$

Importantly, this only depends on inner products of $\phi(\mathbf{x})^\top \Sigma_w \phi(\mathbf{x}')$, not explicitly on $\phi(\mathbf{x})$,

Define Kernel Function: $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \Sigma_w \phi(\mathbf{x}')$. This lets us work directly with kernels without needing to specify $\phi(\cdot)$.

- Avoids explicit choice the shape and number of basis functions
- This is now non-parametric, complexity grows with data not parameters.

Defining Gaussian Processes

Definition: A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution [5]. Formally, a GP is defined by a mean function $m(\mathbf{x})$ and a covariance function $k(\mathbf{x}, \mathbf{x}')$:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')). \tag{4}$$

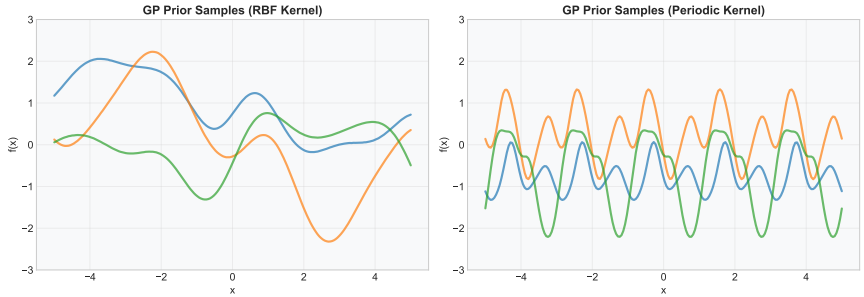
This means that for any finite set of inputs $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]$, the corresponding function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ are jointly Gaussian:

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}), \tag{5}$$

where $\mathbf{K}$ is the covariance matrix defined by a kernel function $k(\cdot, \cdot)$ evaluated at the inputs.

GP Prior: Distribution Over Functions

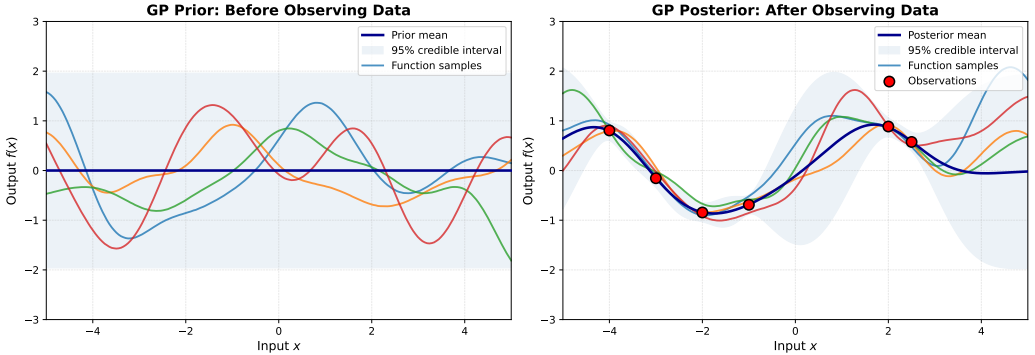
The GP prior $f \sim \mathcal{GP}(0, k(\cdot, \cdot))$ captures assumptions about properties of functions, such as smoothness or periodicity, encoded by the kernel.



- Each line is a random function drawn from the GP prior
- All functions are plausible before seeing any data
- Kernel choice controls function properties: smoothness, periodicity, etc.

GP Posterior: Learning From Data

Using Bayes rule, we condition the GP prior on observations to get the posterior: $p(f|\mathcal{D}) \propto p(\mathcal{D}|f)p(f)$.



- GP learns from noisy observations to predict the underlying function
- Uncertainty (shaded region) is small near data, large far from data

Gaussian Processes: Summary

Strengths

- Non-parametric flexibility
- Automatic uncertainty quantification

Limitations

- $O(N^3)$ growth [5]
- Struggles with non-stationarity [4]
- Struggles with hierarchical representations [2]
- Manual kernel selection

$\Rightarrow$ **Motivates Deep GPs**

Deep Gaussian Processes: Motivation

- To solve the issues mentioned for GPs, we need a more flexible model.

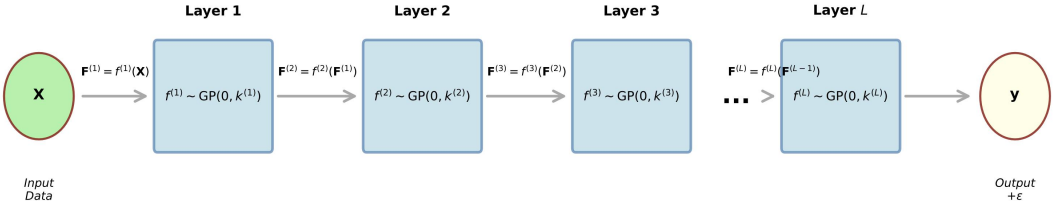
Idea: Similar to deep neural networks [8], why not stack GPs on top of each other, where the output of one GP becomes the input to the next?

$$\mathbf{X} \xrightarrow{f^{(1)}} \mathbf{F}^{(1)} \xrightarrow{f^{(2)}} \mathbf{F}^{(2)} \xrightarrow{\cdots} \mathbf{F}^{(L)} \rightarrow \mathbf{y} \tag{6}$$

DGP: Definition

DGP definition [2, 6]: A DGP with L layers is a hierarchical composition where each layer ℓ is a GP conditioned on the previous layer's output:

$$f^{(\ell)} \mid \mathbf{F}^{(\ell-1)} \sim \mathcal{GP}\left(0, k^{(\ell)}(\cdot,\cdot)\right), \quad \ell = 1, \ldots, L. \tag{7}$$



Each $f^{(\ell)}$ learns features from $\mathbf{F}^{(\ell-1)}$, with $\mathbf{F}^{(0)} = \mathbf{X}$.

Intractability of Exact Inference

Issue: The DGP posterior is:

$$p(\mathbf{F}^{(1:L)} \mid \mathbf{X}, \mathbf{y}) = \frac{p(\mathbf{y} \mid \mathbf{F}^{(L)}) \prod_{\ell=1}^{L} p(\mathbf{F}^{(\ell)} \mid \mathbf{F}^{(\ell-1)})}{\int p(\mathbf{y} \mid \mathbf{F}^{(L)}) \prod_{\ell=1}^{L} p(\mathbf{F}^{(\ell)} \mid \mathbf{F}^{(\ell-1)}) \, d\mathbf{F}^{(1)} \dots d\mathbf{F}^{(L)}}. \tag{8}$$

This is intractable for $L \geq 2$. Both the numerator and denominator involve integrating over nested GPs. [2, 7]

Therefore how do we train DGPs?

Sparse Variational Inference for DGPs

Solution: Variational inference with inducing points [3, 7]

Core idea:

- Approximate the intractable posterior $p(\mathbf{f}|\mathbf{y})$ with a simpler variational distribution $q(\mathbf{f})$
- Instead of working with all N training points, introduce $M \ll N$ **inducing points** $\mathbf{u}_\ell$ per layer

What are inducing points?

- Pseudo inputs and their function values at a small set of locations.
- Act as a summary of the full function, capturing key characteristics
- Their locations and values are optimised during training
- Allow computational cost to scale with M (typically 100–1000) instead of N (which could be millions)

The Evidence Lower Bound (ELBO)

We maximize the ELBO to approximate the true posterior [7]:

$$\mathcal{L} = \underbrace{\mathbb{E}_{q(\mathbf{f}^{(1:L)})}\left[\log p(\mathbf{y}|\mathbf{f}^{(L)})\right]}_{\text{Data fit}} - \underbrace{\sum_{\ell=1}^{L} \mathrm{KL}(q(\mathbf{u}_\ell)||p(\mathbf{u}_\ell))}_{\text{Regularisation per layer}}$$

- **Data fit term:** Measures how well the model predicts observed data
- **KL divergence term:** Prevents overfitting by keeping the variational distribution close to the prior
- Optimization performed via stochastic gradient descent with minibatches

Computational gain: $O(N^3) \to O(NM^2)$ with $M \approx 100$–1000 inducing points

Deep Gaussian Processes: Summary

Why use DGPs

- Model complex, non-stationary functions GPs struggle with [4]
- Flexible kernel composition without manual design

Training-focused challenges

- Exact inference is intractable. Need approximations
- Increased cost and tuning
- Sensitive to getting stuck in local minima
- Uncertainty quantification no longer directly computable, needs approximations.

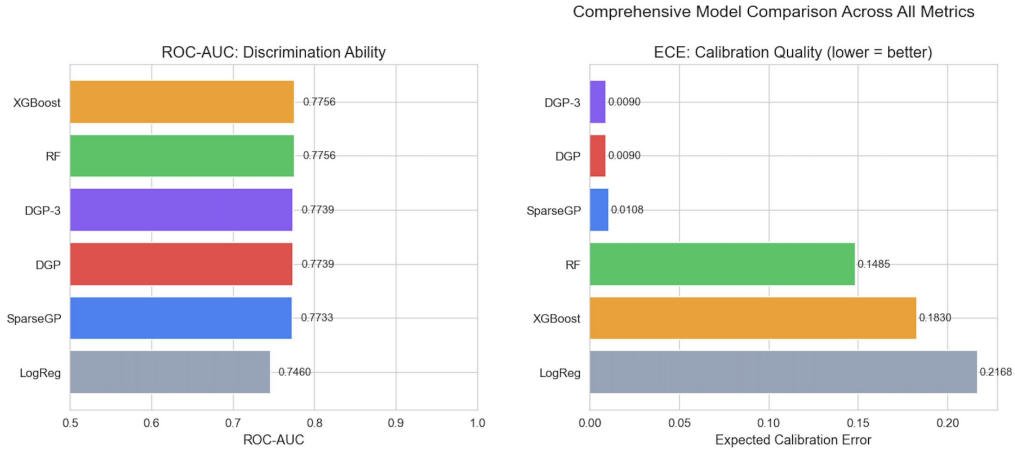
Credit Risk Modelling: Initial Results

So what can we do with GPs and DGPs?
Credit Risk Modelling: Probability of Default Prediction.

- UCI Credit Card Dataset: 30,000 transactions with 23 features. (80% train, 20% test split).
- Single-layer GP vs 2-layer and 3-layer DGPs, trained using doubly stochastic variational inference [7].
- Compared against logistic regression, random forest and XGBoost baselines.
- Goal is to predict the probability that a customer will default on their next payment.

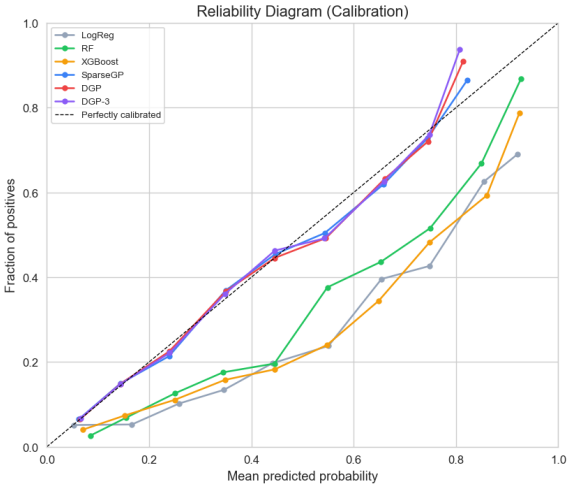
Full results, code and analysis will be included in final dissertation. Here we present initial results.

Initial Results: Overall Performance Comparison



Calibration Quality: A Critical Advantage

- Calibration measures how well predicted probabilities match observed frequencies.
- This means GP probabilities are trustworthy, critical for interpretability and decision-making in regulated environments [5, 7].



Key Results

1. **GPs/DGPs deliver similar discriminative performance to other ML models**
   - AUC around 0.77 for all models, no significant difference between GPs/DGPs or other tested models.
   - So all models are equally good at ranking customers by risk.
2. **No significant improvement from DGPs over GPs in this dataset**
3. **Calibration massively better in GPs and DGPs**
   - GPs also provide full uncertainty distributions over predictions. Not explored in our initial results. Will be discussed in more depth in the final dissertation.

Questions?

References I

[1] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[2] Andreas C. Damianou and Neil D. Lawrence. Deep Gaussian Processes, March 2013. URL `http://arxiv.org/abs/1211.0358`. arXiv:1211.0358 [stat].

[3] James Hensman, Nicolo Fusi, and Neil D. Lawrence. Gaussian Processes for Big Data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290. AUAI Press, 2013.

[4] Christopher J. Paciorek and Mark J. Schervish. Nonstationary Covariance Functions for Gaussian Process Regression. In *Advances in Neural Information Processing Systems*, volume 16, pages 273–280. MIT Press, 2004.

[5] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. ISBN 978-0-262-18253-9.

References II

[6] Hugh Salimbeni. *Deep Gaussian Processes: Advances in Models and Inference*. PhD thesis, Imperial College London, 2019.

[7] Hugh Salimbeni and Marc Deisenroth. Doubly Stochastic Variational Inference for Deep Gaussian Processes, November 2017. URL `http://arxiv.org/abs/1705.08933`. arXiv:1705.08933 [stat].

[8] Christopher Williams. Computing with Infinite Networks. In M. C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996. URL `https://proceedings.neurips.cc/paper_files/paper/1996/file/ae5e3ce40e0404a45ecacaaf05e5f735-Paper.pdf`.