Gaussian Processes

Definition, applications and deep extension

Luis Antonio Ortega Andrés

March 26, 2021

Table of contents

Definitions

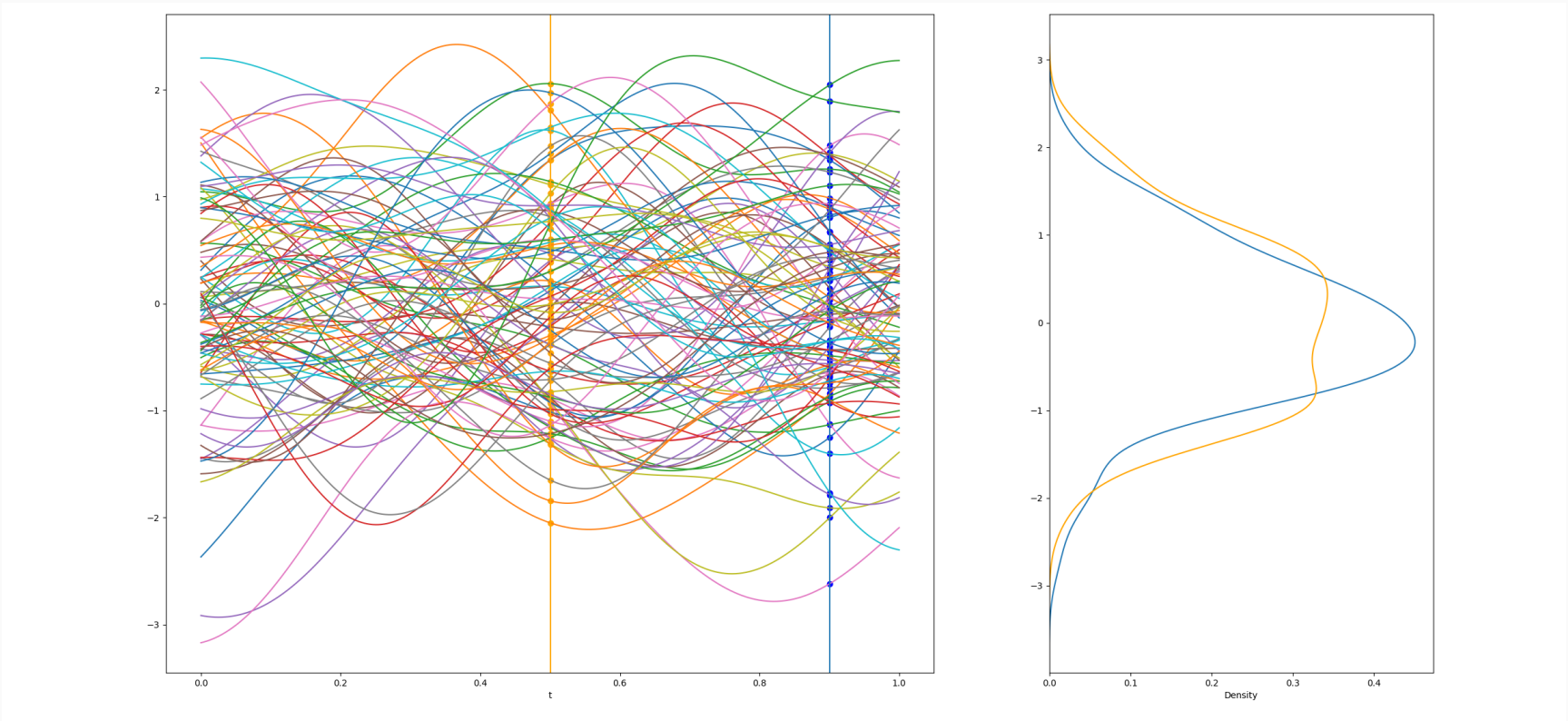
Gaussian process

Gaussian process

A **Gaussian process** is a **stochastic process**, i.e, a collection of random variables $\{X_t\}_{t \in \mathcal{T}}$ indexed by a set $\mathcal{T}$, such that any finite subset is Gaussian.

$$\{X_{t_1}, \ldots, X_{t_N}\} \sim \mathcal{N}(\cdot, \cdot)$$

For example, $X_{t_i} \sim \mathcal{N}(\cdot, \cdot)$.



Characterization

Gaussian processes are completely determined by their first and second order moments[1].

Given $\boldsymbol{t} = (t_1, \ldots, t_N)$, $N \in \mathbb{N}$:

$$X(\boldsymbol{t}) = (X_{t_1}, \ldots, X_{t_N}) \sim \mathcal{N}\left(m(\boldsymbol{t}), K(\boldsymbol{t}, \boldsymbol{t})\right)$$

$$\text{where } \begin{cases} m(\boldsymbol{t}) & = \mathbb{E}\left[X(\boldsymbol{t})\right] \\ K(\boldsymbol{t}, \boldsymbol{t}) & = Cov\left(X(\boldsymbol{t}), X(\boldsymbol{t})\right) \end{cases}$$

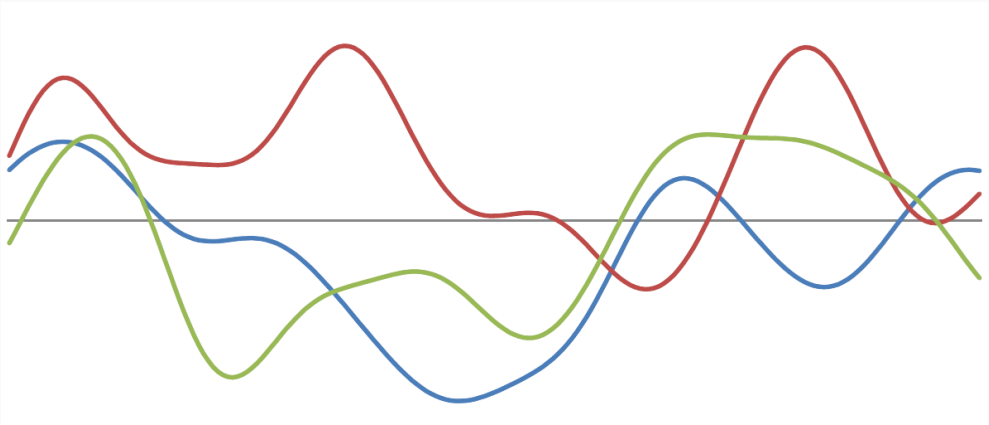
Defining m and K we get a Gaussian process.

[1] Bishop, Christopher M. Pattern recognition and machine learning. Springer, 2006.

Examples

It is usual to take a **zero mean function**, $m(\boldsymbol{t}) = 0$ and **kernel functions**:

- *RBF*: $K(\boldsymbol{t}, \boldsymbol{t}') = \exp\left(-\frac{\|\boldsymbol{t}-\boldsymbol{t}'\|^2}{2\sigma^2}\right)$.
- *Matérn*: Family of kernels, parameterized by ν. Generalize several kernels.



Define a **probability distribution over functions**: *Given a function, which is its probability when interpreted as a sample of a Gaussian process?*

Regression Problem

Problem statement

Given dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ and an unknown function f such that

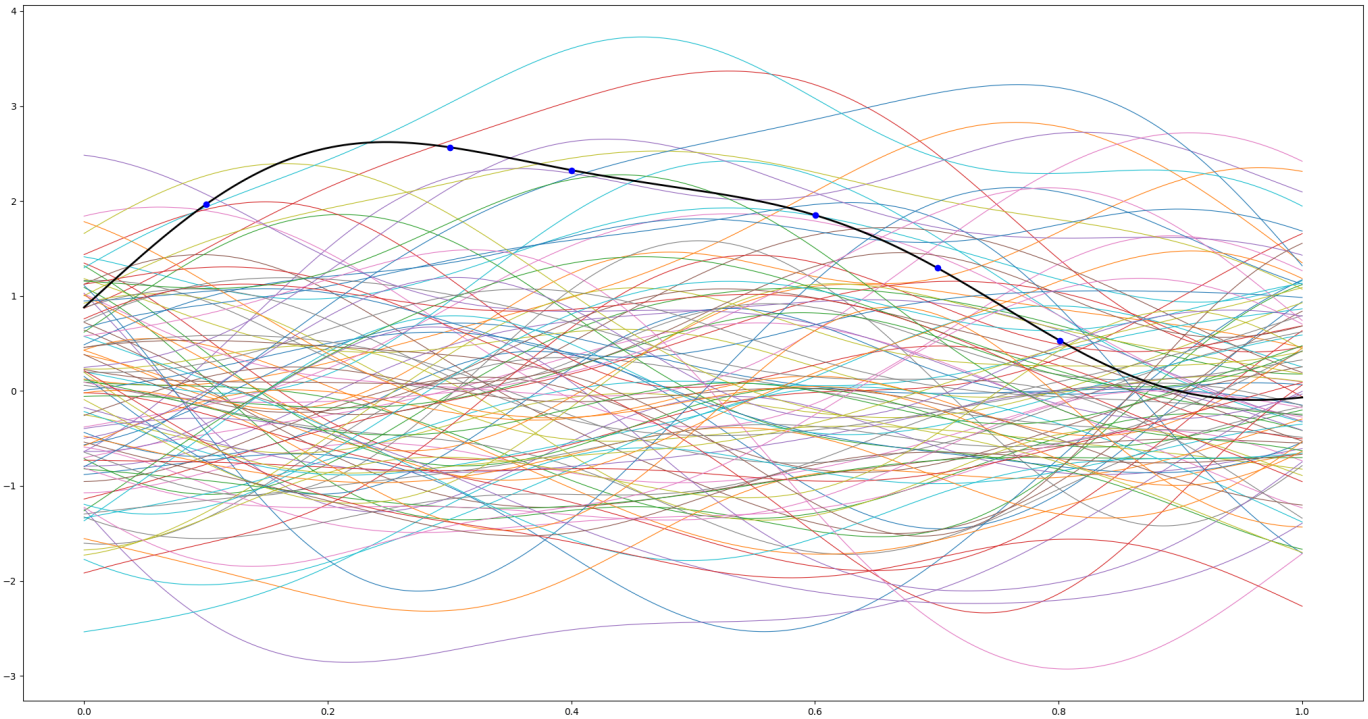
$$y_n = f(\boldsymbol{x}_n) + \epsilon \quad \forall n = 1, \ldots, N, \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$$

Assumption

Function f is a Gaussian process of unknown mean function m and kernel function K

Assumption

Function f is a Gaussian process of unknown mean function m and kernel function K



Naming $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and $\boldsymbol{y} = (y_1, \ldots, y_N)$:

$$\boldsymbol{y} = f(\boldsymbol{X}) + \mathcal{N}(0, \sigma^2 \boldsymbol{I}) \implies \boldsymbol{y} \sim \mathcal{N}\left(m(\boldsymbol{X}), K(\boldsymbol{X}, \boldsymbol{X}) + \sigma^2 \boldsymbol{I}\right).$$

Remark: *A distribution is assumed over function points* **but not over** x.

Let $\boldsymbol{X}^*$ be a test case where $\boldsymbol{y}^* = f(\boldsymbol{X}^*) + \mathcal{N}(0, \sigma^2 \boldsymbol{I})$:

$$\begin{pmatrix} f(\boldsymbol{X}) \\ f(\boldsymbol{X}^*) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(\boldsymbol{X}) \\ m(\boldsymbol{X}^*) \end{pmatrix}, \begin{pmatrix} K(\boldsymbol{X}, \boldsymbol{X}) & K(\boldsymbol{X}, \boldsymbol{X}^*) \\ K(\boldsymbol{X}^*, \boldsymbol{X}) & K(\boldsymbol{X}^*, \boldsymbol{X}^*) \end{pmatrix}\right)$$

An **usual assumption** is that $m = 0$.

Remark

Typically, $\boldsymbol{x}$ is erased from the notation.

Naming $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$ and $\boldsymbol{y} = (y_1, \ldots, y_N)$:

$$\boldsymbol{y} = \boldsymbol{f} + \mathcal{N}(0, \sigma^2 \boldsymbol{I}) \implies \boldsymbol{y} \sim \mathcal{N}\left(0, \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2 \boldsymbol{I}\right).$$

Remark: *A distribution is assumed over* $\boldsymbol{f}$ *points* **but not over** $\boldsymbol{X}$.

Let $\boldsymbol{X}^*$ be a test case where $\boldsymbol{y}^* = \boldsymbol{f}^* + \mathcal{N}(0, \sigma^2 \boldsymbol{I})$:

$$\begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}^*} \\ \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} \end{pmatrix}\right)$$

Predictive posterior

$$\begin{pmatrix} \boldsymbol{f} \\ \boldsymbol{f}^* \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}^*} \\ \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}} & \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} \end{pmatrix}\right) \quad \boldsymbol{y} \mid \boldsymbol{f} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I})$$

$$\Downarrow$$

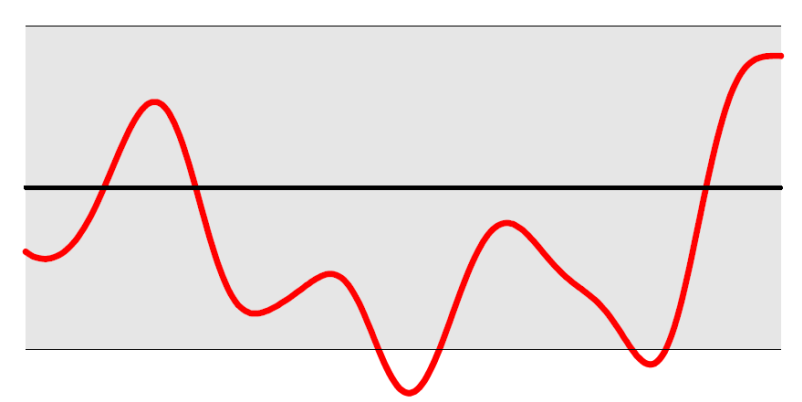
$$P(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}^*) = P(\boldsymbol{y} \mid \boldsymbol{f}) \implies \boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{f}^* \sim \mathcal{N}(\boldsymbol{f}, \sigma^2 \boldsymbol{I})$$

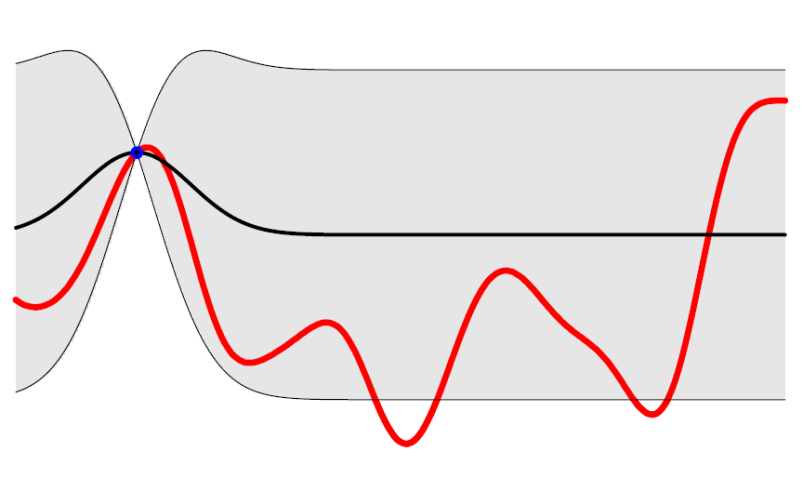
$$\Downarrow$$

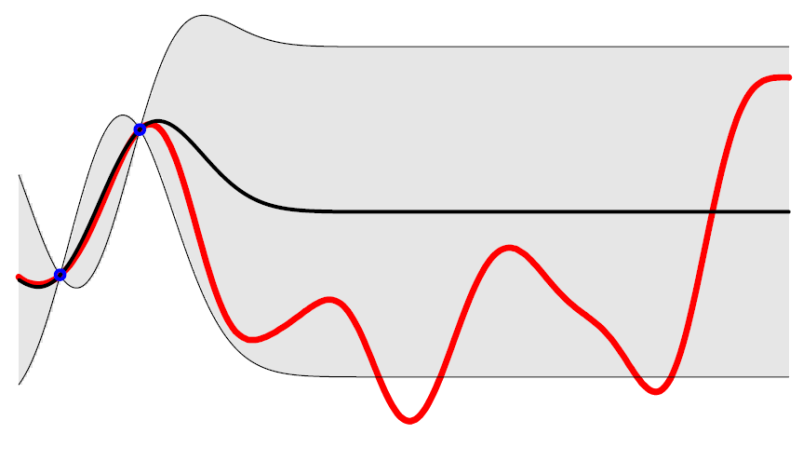
$$\boldsymbol{f}, \boldsymbol{f}^* \mid \boldsymbol{y} \sim \mathcal{N}(\cdot, \cdot) \implies \boldsymbol{f}^* \mid \boldsymbol{y} \sim \mathcal{N}(\cdot, \cdot)$$

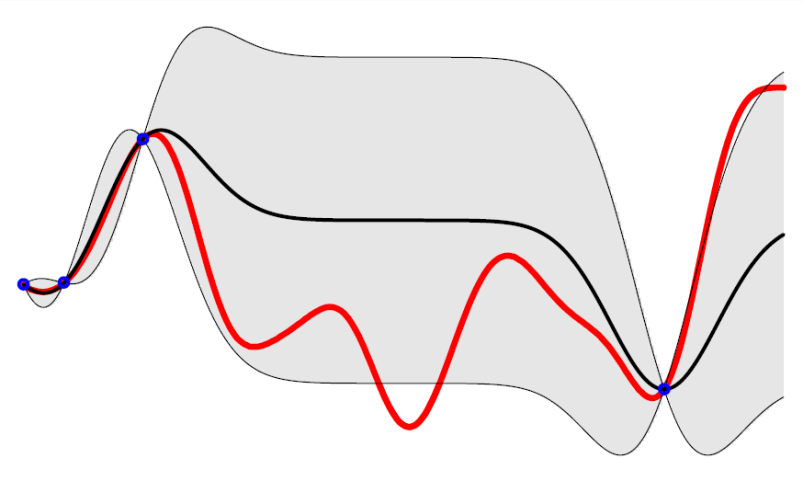
$$\Downarrow$$

$$\boldsymbol{f}^* \mid \boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \begin{cases} \boldsymbol{\mu} = \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{y} \\ \boldsymbol{\Sigma} = \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} - \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}^*} \end{cases}$$

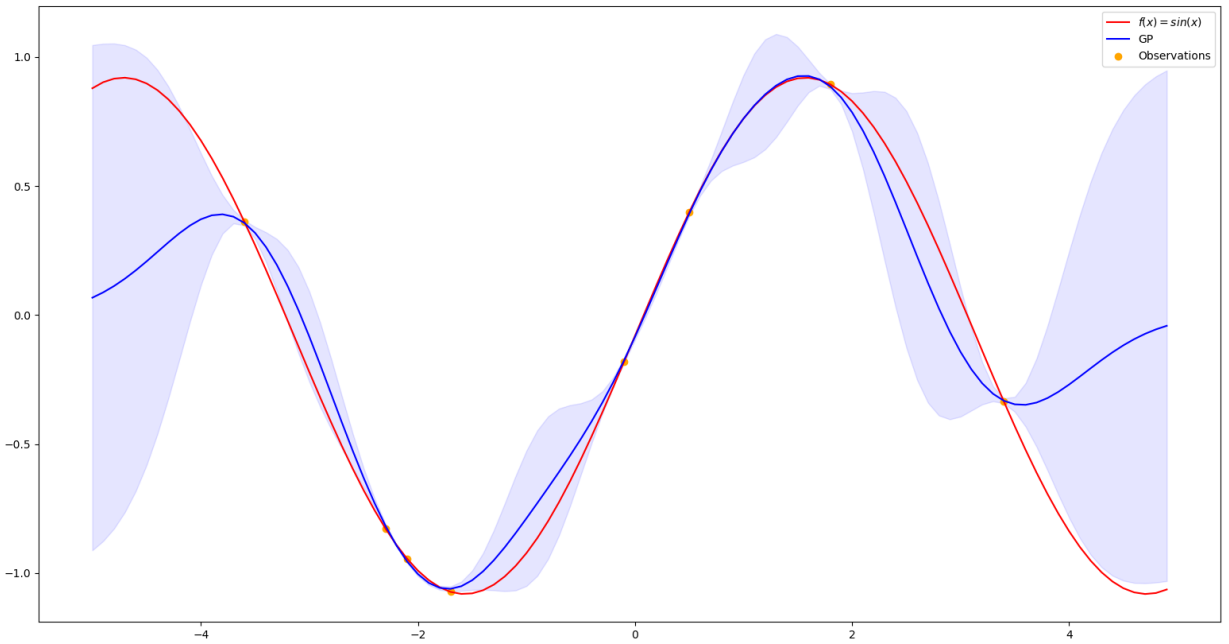








Unknown function $f(x) = \sin(x)$, $\mathcal{D}$ is a sample of 8 points in $(-5, 5)$, RBF kernel.



Computational complexity

Several computations are done, assuming $\boldsymbol{X}$ has N points and $\boldsymbol{X}^*$ has M:

$$\begin{aligned} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} &\implies \mathcal{O}(N^2) \\ \left(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2 \boldsymbol{I}\right)^{-1} &\implies \mathcal{O}(N^3) \\ \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}^*} = \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}}^T &\implies \mathcal{O}(NM) \\ \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} &\implies \mathcal{O}(M^2) \end{aligned}$$

Training: $\mathcal{O}(N^3)$ and Test: $\mathcal{O}(NM^2)$. They are **computationally inefficient!!**

This can be slightly reduced using the *Cholesky decomposition* for the matrix inversion.

Advantages

They give a prediction $\boldsymbol{\mu} = \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}$. **Equals the kernel ridge regression estimator!!**.

Implicit **confidence interval**, $(\boldsymbol{\mu} - 3\boldsymbol{\Sigma}, \boldsymbol{\mu} + 3\boldsymbol{\Sigma})$.

Full **probabilistic approach**.

Inducing points

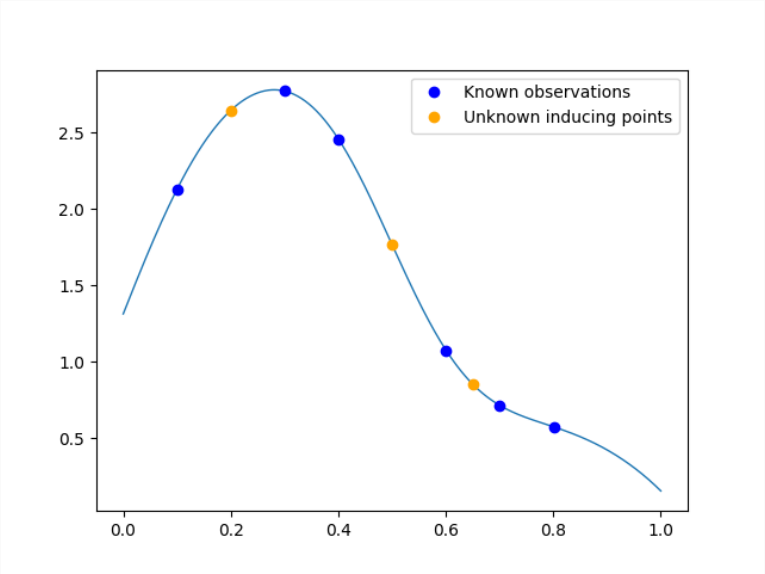
Inducing points

Main idea: Use a *smaller* and *hidden* set of points.

Let $\boldsymbol{X}_u \subset \mathbb{R}^D$ be a set of *known* points (commonly computed from $\boldsymbol{X}$).
We make the assumption that $\boldsymbol{u} = f(\boldsymbol{X}_u)$ is representative of $\boldsymbol{y}$.

Remark. The *inducing points* $\boldsymbol{u}$ are *unknown* and must be *marginalized*.

$$
\begin{aligned}
P(\boldsymbol{f}, \boldsymbol{f}^*) &= \int P(\boldsymbol{f}, \boldsymbol{f}^*, \boldsymbol{u}) \, d\boldsymbol{u} \\
&= \underbrace{\int P(\boldsymbol{f}, \boldsymbol{f}^* \mid \boldsymbol{u}) P(\boldsymbol{u}) \, d\boldsymbol{u}}_{\textit{Intractable}}
\end{aligned}
$$



Where $\boldsymbol{u}$ are taken from a Gaussian process:

$$\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{K}(\boldsymbol{X}_u, \boldsymbol{X}_u)).$$

Approaches

- Exact inference in approximated model:

$$P(\boldsymbol{f}, \boldsymbol{f}^* \mid \boldsymbol{u}) = P(\boldsymbol{f} \mid \boldsymbol{u})\ P(\boldsymbol{f}^* \mid \boldsymbol{u})$$

$$\Downarrow$$

$$P(\boldsymbol{f}, \boldsymbol{f}^*) = \int P(\boldsymbol{f} \mid \boldsymbol{u})\ P(\boldsymbol{f}^* \mid \boldsymbol{u})\ P(\boldsymbol{u})\ d\boldsymbol{u}$$

  And further approximate $P(\boldsymbol{f} \mid \boldsymbol{u})$ and $P(\boldsymbol{f}^* \mid \boldsymbol{u})$

- Variational inference:

$$Q(\boldsymbol{u}) \approx P(\boldsymbol{u} \mid \boldsymbol{y}).$$

Approximated models

Exact conditionals

$$P(\boldsymbol{f} \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q}_{\boldsymbol{f},\boldsymbol{f}})$$
$$P(\boldsymbol{f}^* \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} - \boldsymbol{Q}_{\boldsymbol{f}^*,\boldsymbol{f}^*})$$
$$\boldsymbol{Q}_{\boldsymbol{a},\boldsymbol{b}} = \boldsymbol{K}_{\boldsymbol{a},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{b}}$$

- The Subset of Regressors approximation

$$Q_{SOR}(\boldsymbol{f} \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, 0)$$
$$Q_{SOR}(\boldsymbol{f}^* \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, 0)$$

- The Deterministic Training Conditional approximation

$$Q_{DTC}(\boldsymbol{f} \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, 0)$$
$$Q_{DTC}(\boldsymbol{f}^* \mid \boldsymbol{u}) = P(\boldsymbol{f}^* \mid \boldsymbol{u})$$

Exact conditionals

$$P(\boldsymbol{f} \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q}_{\boldsymbol{f},\boldsymbol{f}})$$
$$P(\boldsymbol{f}^* \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, \boldsymbol{K}_{\boldsymbol{f}^*,\boldsymbol{f}^*} - \boldsymbol{Q}_{\boldsymbol{f}^*,\boldsymbol{f}^*})$$
$$\boldsymbol{Q}_{\boldsymbol{a},\boldsymbol{b}} = \boldsymbol{K}_{\boldsymbol{a},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{b}}$$

- The Fully Independent Training Conditional approximation

$$Q_{FITC}(\boldsymbol{f} \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, \mathit{diag}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q}_{\boldsymbol{f},\boldsymbol{f}}))$$
$$Q_{FITC}(\boldsymbol{f}^* \mid \boldsymbol{u}) = P(\boldsymbol{f}^* \mid \boldsymbol{u})$$

- The Partially Independent Training Conditional approximation

$$Q_{PITC}(\boldsymbol{f} \mid \boldsymbol{u}) = \mathcal{N}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{u}}\boldsymbol{K}_{\boldsymbol{u},\boldsymbol{u}}^{-1}\boldsymbol{u}, \mathit{blockdiag}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} - \boldsymbol{Q}_{\boldsymbol{f},\boldsymbol{f}}))$$
$$Q_{PITC}(\boldsymbol{f}^* \mid \boldsymbol{u}) = P(\boldsymbol{f}^* \mid \boldsymbol{u})$$

Variational bounds

Using Jensen's inequality:

$$\log P(\boldsymbol{y} \mid \boldsymbol{u}) = \log \mathbb{E}_{P(\boldsymbol{f}|\boldsymbol{u})}[P(\boldsymbol{u} \mid \boldsymbol{f})] \geq \mathbb{E}_{\log P(\boldsymbol{f}|\boldsymbol{u})}[P(\boldsymbol{u} \mid \boldsymbol{f})] \equiv \mathcal{L}_1,$$

raises Titsias' bound [2]

$$\log P(\boldsymbol{y}) = \log \int P(\boldsymbol{y} \mid \boldsymbol{u}) P(\boldsymbol{u}) d\boldsymbol{u} \geq \log \int \exp \mathcal{L}_1 P(\boldsymbol{u}) d\boldsymbol{u} \equiv \mathcal{L}_2.$$

But it is not suitable for **stochastic optimization**. Appears a new bound[3]

$$\log P(\boldsymbol{y}) \geq \mathbb{E}_{Q(\boldsymbol{u})} \left[\mathcal{L}_1 + \log P(\boldsymbol{u}) - \log Q(\boldsymbol{u})\right] \equiv \mathcal{L}_3.$$

---

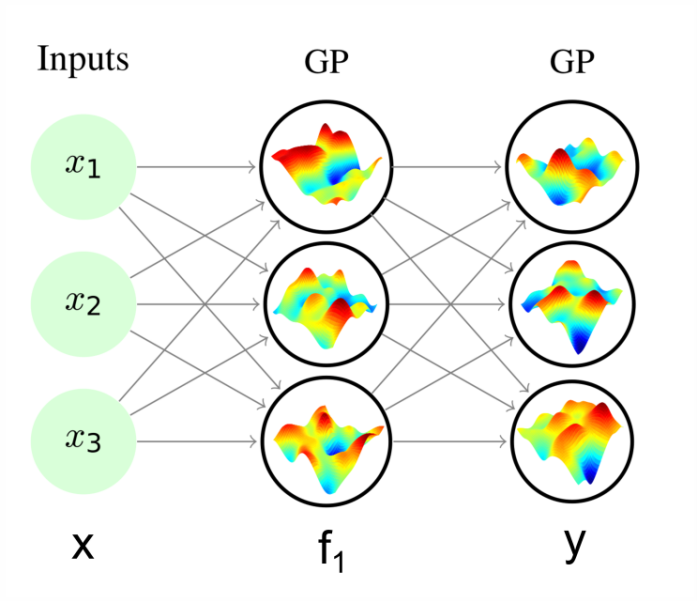
[2] Titsias, Michalis. "Variational learning of inducing variables in sparse Gaussian processes." In Artificial intelligence and statistics, 2009.

[3] Hensman, James, Nicolo Fusi, and Neil D. Lawrence. "Gaussian processes for big data." 2013.

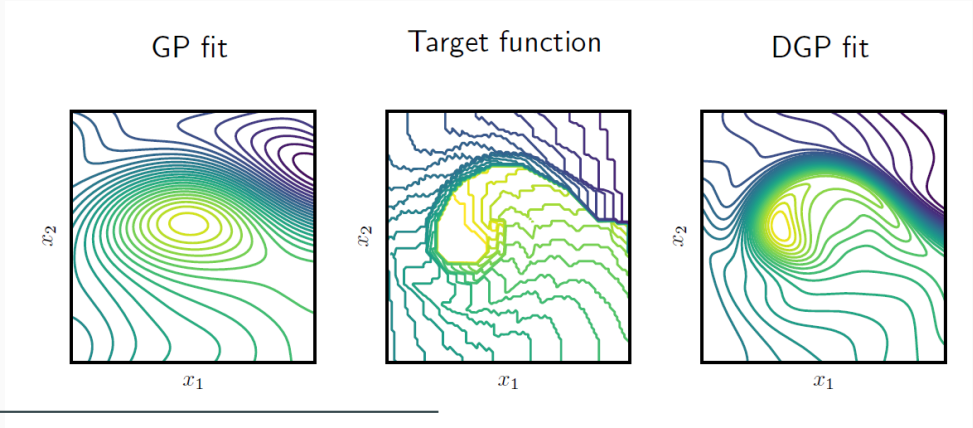
Deep Gaussian processes

General idea

Connect Gaussian processes in a chain.



[4]



[4] Image reference: https://www.groundai.com/project/inference-in-deep-gaussian-processes-using-stochastic-gradient-hamiltonian-monte-carlo/1

Definition

Let $\mathcal{D} = (\boldsymbol{X}, \boldsymbol{y})$ be a dataset of an unknown function f.

Deep Gaussian process

A **deep Gaussian process** of length L considers L independent Gaussian processes $f^1, \ldots, f^L$ such that the input of a Gaussian process is the output of the previous one.

$$\boldsymbol{X}^1 = f^1(\boldsymbol{X}) \implies \boldsymbol{X}^2 = f^2(\boldsymbol{X}^2) \implies \cdots \implies \boldsymbol{X}^L = f^L(\boldsymbol{X}^{L-1}) \approx \boldsymbol{Y}$$

Problem

In the Gaussian process, the input **did not** follow any distribution.

$$\boldsymbol{X} \text{ no distribution} \implies \boldsymbol{X}^1 \sim \mathcal{N}(\cdot, \cdot) \implies \boldsymbol{X}^2 \text{ no longer Gaussian}$$

Distribution in inner layers cannot be computed in closed form.

Distribution in inner layers cannot be computed in closed form.

- Variational inference is used to train the model.
- Different evidence lower bounds depending on the assumptions made (inducing points might be considered in each inner layer).
- Distribution is intractable but samples can be taken easily $\implies$ Monte Carlo.
- Expectation propagation algorithm is used.

Questions?

References i


C. M. Bishop.
Pattern recognition and machine learning.


T. D. Bui, J. M. Hernández-Lobato, D. Hernández-Lobato, Y. Li, and R. E. Turner.
Deep Gaussian Processes for Regression using Approximate Expectation Propagation.


J. Hensman, N. Fusi, and N. D. Lawrence.
Gaussian Processes for Big Data.


J. Quiñonero-Candela and C. E. Rasmussen.
A Unifying View of Sparse Approximate Gaussian Process Regression.


R. Ranganath, S. Gerrish, and D. M. Blei.
Black Box Variational Inference.

References ii


H. Salimbeni and M. Deisenroth.
Doubly Stochastic Variational Inference for Deep Gaussian Processes.


E. Snelson and Z. Ghahramani.
Sparse Gaussian Processes using Pseudo-inputs.


M. K. Titsias.
Variational Learning of Inducing Variables in Sparse Gaussian Processes.