

Tutorial: Gaussian Process Regression

Jonas El Gammal

Abstract

This tutorial provides an introduction to Gaussian Process (GP) regression, a powerful non-parametric Bayesian method for machine learning. We introduce the concept of stochastic processes and GPs, explain how to condition GPs on observed data, discuss kernel functions and their properties, and present the GP regression algorithm. The material is based on research from my master thesis and PhD thesis on Bayesian inference using Gaussian processes.

1 Introduction

Gaussian Processes (GPs) have become a widely used tool for regression and classification tasks over the past few decades. They are a mathematically simple, yet powerful, non-parametric Bayesian method leveraging the simple structure of multivariate Gaussian distributions such as analytical marginalization and conditioning [5].

The key idea behind GPs is to assume that any finite set of function values of an arbitrary function $f(x)$ is jointly Gaussian distributed. This allows us to define a distribution over functions, making GPs particularly useful for interpolation and regression tasks where we want to predict function values at new locations based on observed data.

2 Concept of Gaussian Processes

2.1 Stochastic Processes

To understand GPs, it is useful to first define a stochastic process. A stochastic process can be thought of as a function $\{Y(t) : t \in T\}$, where Y is a random variable drawn from some probability measure P . T is often referred to as the *index set* [3].

With this, we can define a Gaussian Process:

Definition 1. *A Gaussian Process is a collection of random variables, any finite number of which have a joint Gaussian distribution [5].*

This means that a GP is a stochastic process defined on any set $T = \{t_1, \dots, t_n\}$ where the n values $\{y_1, \dots, y_n\}$ are drawn from a joint Gaussian distribution:

$$\mathcal{N}(\mathbf{t}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{t} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{t} - \boldsymbol{\mu})\right). \quad (1)$$

Here $\mathbf{t} = (t_1, \dots, t_n)^\top$ is the vector of indices, $\boldsymbol{\mu}$ the *mean* vector (representing the expected values), and $\boldsymbol{\Sigma}$ the *covariance* matrix (capturing the relationships between the indices).

2.2 Gaussian Processes as Distributions over Functions

For a GP, we make $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ functions of the index set X such that for any two points $x, x' \in X$, the mean and covariance can be expressed as

$$m(x) = \mathbb{E}[f(x)], \quad (2)$$

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]. \quad (3)$$

This allows defining a GP as a distribution over functions $f(x)$, which can be expressed as

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

where $m(x)$ and $k(x, x')$ are the mean and covariance functions, respectively. The covariance function $k(x, x')$ is often called the *kernel*.

The definition as a stochastic process furthermore implies a consistency requirement (also called Kolmogorov's extension theorem [6]), which demands that any GP that specifies $(y_1, \dots, y_n) \sim \mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ on any set X must equally specify $(y'_1, \dots, y'_n) \sim \mathcal{N}(x'|\boldsymbol{\mu}', \boldsymbol{\Sigma}')$ for any subset $X' \subset X$ by taking the relevant parts of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. In other words, this means that predictions about a finite subset of indices can be made without knowing the full infinite-dimensional distribution.

Typically, this *unconditioned* version of the GP is called the prior GP as it reflects the distribution of functions that one would draw if one had no knowledge about the function. This GP prior is fully specified by the mean and kernel functions.

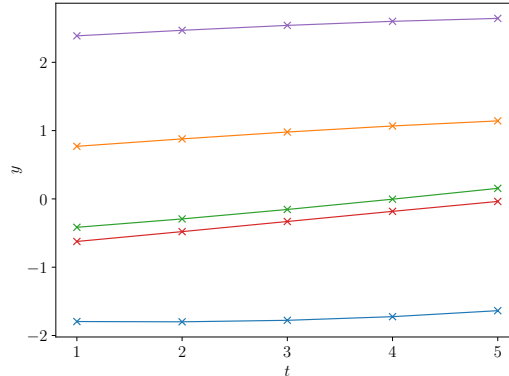


Figure 1: Samples drawn from a multivariate Gaussian distribution with $m(t) = 0$ and $k(t, t') = \exp\left(-\frac{(t-t')^2}{200}\right)$. The samples are indexed by the set $\{1, 2, 3, 4, 5\}$ and demonstrate how discrete samples from a GP can represent smooth functions.

3 Conditioning Gaussian Processes

To incorporate knowledge from a set of training points $\{(x_i, y_i = f_i) | i = 1, \dots, n\}$, we condition the GP. The joint distribution of the training points $\mathbf{f}(\mathbf{x}) \equiv \mathbf{y}$ and test points $\mathbf{f}_*(\mathbf{x}_*) \equiv \mathbf{y}_*$ is given by:

$$\begin{bmatrix} \mathbf{f}(\mathbf{x}) \\ \mathbf{f}_*(\mathbf{x}_*) \end{bmatrix} \equiv \begin{bmatrix} \mathbf{y} \\ \mathbf{y}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{m}(\mathbf{x}) \\ \mathbf{m}(\mathbf{x}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}(\mathbf{x}, \mathbf{x}) & \mathbf{K}(\mathbf{x}, \mathbf{x}_*) \\ \mathbf{K}(\mathbf{x}_*, \mathbf{x}) & \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) \end{bmatrix}\right), \quad (5)$$

where $\mathbf{m}_i = m(x_i)$ is the vector of the mean function and $\mathbf{K}_{i,j} = k(x_i, x_j)$ is called the *Gram matrix* of the training points.

Conditioning on the observed values is straightforward for multivariate Gaussians:

$$\mathbf{f}_*(\mathbf{x}_*) | \mathbf{x}, \mathbf{f}(\mathbf{x}) \sim \mathcal{N}(\bar{\mathbf{f}}_*, \boldsymbol{\Sigma}_{\mathbf{f}_*}), \quad (6)$$

with

$$\boldsymbol{\mu}(\mathbf{x}_*) \equiv \bar{\mathbf{f}}_* = \mathbf{m}(\mathbf{x}_*) + \mathbf{K}(\mathbf{x}_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}(\mathbf{f}(\mathbf{x}) - \mathbf{m}(\mathbf{x})), \quad (7)$$

and

$$\text{cov}(\mathbf{f}_*(\mathbf{x}_*)) \equiv \Sigma_{\mathbf{f}_*} = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}(\mathbf{x}_*, \mathbf{x})\mathbf{K}(\mathbf{x}, \mathbf{x})^{-1}\mathbf{K}(\mathbf{x}, \mathbf{x}_*). \quad (8)$$

This conditioned GP is called the *GP-posterior*. For brevity, the explicit dependence on the inducing points \mathbf{x}_* is often dropped in practice.

Even with a prior zero-mean function $m(x) = 0$, the GP-posterior mean can be non-zero. This motivates the choice of $m(x) = 0$, simplifying the GP construction to the selection of an appropriate kernel function.

Figure 2 illustrates this conditioning process. The left side shows sample functions from a GP prior with $m(x) = 0$ and $k(x, x') = \exp(-(x - x')^2/2)$, while the right side shows sample functions from a GP conditioned on training points.

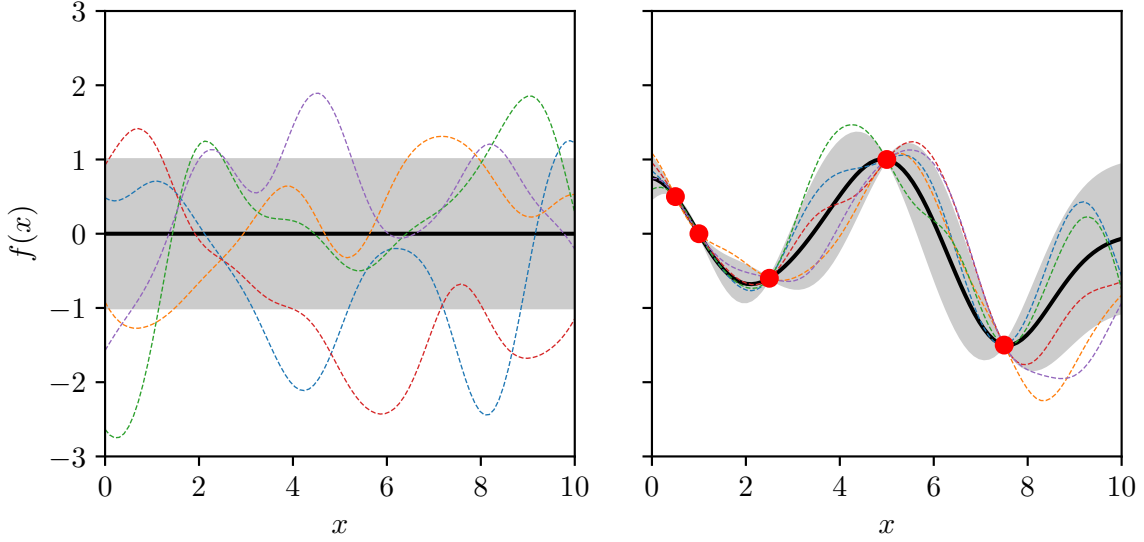


Figure 2: **Left:** Sample functions drawn from a GP with $m(x) = 0$ and $k(x, x') = \exp(-1/2(x - x')^2)$ (dashed lines) as well as the value of the prior GP mean function and the standard deviation $\sqrt{k(x, x)} = 1$ (solid black line and gray band respectively). **Right:** Sample functions drawn from the same GP (dashed lines), values of the posterior GP mean (solid black line), and standard deviation (gray band) after conditioning on five observations (red dots). Note how even with a zero prior mean function $m(x) = 0$, one obtains a non-zero posterior mean. Furthermore, after conditioning, only functions that pass through the training points are allowed.

3.1 Including Noise in Observations

In practice, training data often has associated noise $y = f(x) + \epsilon$, where ϵ is a random variable with variance σ_n^2 . This is incorporated by adding a noise term to the kernel function

$$\tilde{k}(x, x') = k(x, x') + \sigma_n^2 \delta_{x, x'}, \quad (9)$$

where $\delta_{x, x'}$ is the Kronecker delta. In the conditioning step, this term is only applied to the Gram matrix between training points $\tilde{\mathbf{K}}(\mathbf{x}, \mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{x}) + \sigma_n^2 \mathbb{I}$, changing Equations 7 and 8 to

$$\boldsymbol{\mu}_* = \mathbf{m}(\mathbf{x}_*) + \mathbf{K}(\mathbf{x}_*, \mathbf{x})\tilde{\mathbf{K}}(\mathbf{x}, \mathbf{x})^{-1}(\mathbf{f}(\mathbf{x}) - \mathbf{m}(\mathbf{x})), \quad (10)$$

and

$$\text{cov}(\mathbf{f}_*) = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{K}(\mathbf{x}_*, \mathbf{x}) \tilde{\mathbf{K}}(\mathbf{x}, \mathbf{x})^{-1} \mathbf{K}(\mathbf{x}, \mathbf{x}_*). \quad (11)$$

4 The Kernel Function

As shown earlier, with the mean function typically assumed to be zero, the GP is fully characterized by its kernel, which means that the main challenge lies in choosing an appropriate kernel.

4.1 Requirements for Valid Kernels

To choose a valid kernel, it must satisfy the following requirements:

1. The kernel needs to map $k : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$.
2. It needs to be symmetric: $k(x, x') = k(x', x)$.
3. The covariance matrix obtained from the kernel needs to be positive definite: $\mathbf{z}^T \mathbf{K}(\mathbf{x}, \mathbf{x}') \mathbf{z} \geq 0$ for all $\mathbf{z} \in \mathbb{R}^D \setminus \{0\}$ and with $\mathbf{K}(\mathbf{x}, \mathbf{x}')_{ij} = k(\mathbf{x}_i, \mathbf{x}'_j)$ being the Gram matrix of \mathbf{x} and \mathbf{x}' . This condition is fulfilled if and only if $k(x, x') \geq 0$ for all $x, x' \in X$ [1].

In addition to these strict mathematical requirements, the kernel should also reflect the prior knowledge about the functional shape of f as well as possible. Three important properties of kernels that influence the behavior of the GP are stationarity, differentiability, and periodicity.

1. *Stationarity* means that a kernel function is invariant to translations, i.e., $k(x+z, x'+z) = k(x, x') \quad \forall z$. To achieve this, typically the kernel is defined as a function of the distance between two points $r = |x - x'|$ (Euclidean distance if $d > 1$). If the kernel is stationary, the GP has the same property.
2. Likewise, *differentiability* directly translates to the GP: If the kernel is n times differentiable, the GP is n times differentiable as well. Typically it is desirable that the kernel be at least once differentiable to ensure that the GP is continuous.
3. A less commonly enforced property is *periodicity* $k(x, x') = k(x, x' + n \cdot z)$, $n \in \mathbb{Z}$ with periodicity z . This is less important for most applications, although periodic kernels do exist in specific contexts.

4.2 Common Kernel Functions

Perhaps the most commonly used kernel function is the *Radial Basis Function* (RBF) kernel. Typically, it is defined as

$$k^{\text{RBF}}(x, x') \equiv k^{\text{RBF}}(r) = C^2 \cdot \exp\left(-\frac{r^2}{2l^2}\right), \quad l \in \mathbb{R}, \quad (12)$$

where the output scale C^2 is commonly referred to as a *constant kernel*. The RBF kernel is infinitely differentiable, stationary, and not periodic. It produces very smooth GPs. It has been argued by some authors that this makes the kernel unsuitable for applications where real-world data is involved [5].

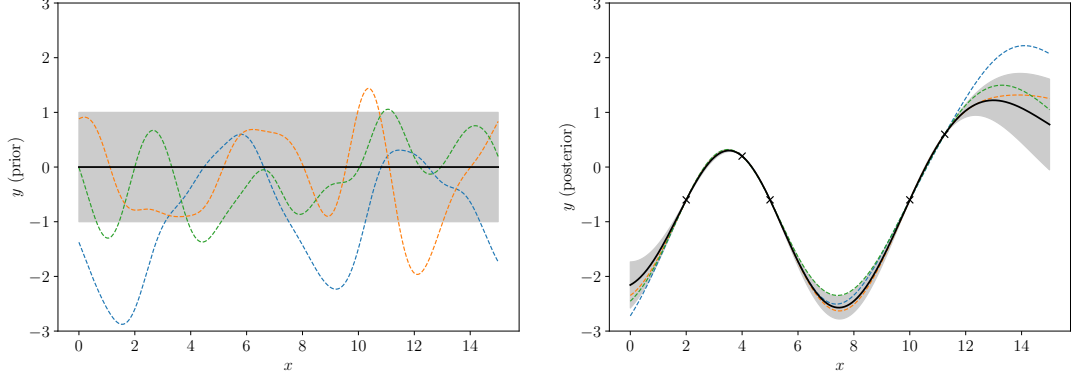


Figure 3: **Left:** Sample functions drawn from a GP with an RBF kernel ($l = 1$) as well as the value of the prior mean function and the standard deviation (solid black line and gray band). **Right:** Same GP after conditioning on five observations (black crosses). Note how the sample functions are very smooth.

The Matérn is a generalization of the RBF kernel. It introduces an additional parameter ν which controls the differentiability [4]:

$$k_{\nu}^{\text{Matern}}(r) = C^2 \cdot \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}r}{l} \right)^{\nu} \cdot K_{\nu} \left(\frac{\sqrt{2\nu}r}{l} \right), \quad l \in \mathbb{R}^+, \quad (13)$$

where Γ is the gamma function and K_{ν} the modified Bessel function of the second kind. The kernel is k times differentiable if $\nu > k$ [5]. For $\nu \rightarrow \infty$, the Matérn kernel approaches the RBF kernel. Typically, the Matérn kernel is used with $\nu = 3/2$ or $\nu = 5/2$, which correspond to once and twice differentiable functions, respectively. In this case, the kernel simplifies to

$$k_{\nu=3/2}^{\text{Matern}}(r) = C^2 \left(1 + \frac{\sqrt{3}r}{l} \right) \exp \left(-\frac{\sqrt{3}r}{l} \right), \quad (14)$$

$$k_{\nu=5/2}^{\text{Matern}}(r) = C^2 \left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2} \right) \exp \left(-\frac{\sqrt{5}r}{l} \right). \quad (15)$$

The exponential sine squared (ESS) kernel is an example of a stationary, periodic, and infinitely differentiable kernel [2]:

$$k^{\text{ESS}}(r) = C^2 \exp \left(-\frac{2 \sin^2 \left(\frac{\pi r}{p} \right)}{l^2} \right), \quad (16)$$

where p controls the periodicity.

Examples of GPs conditioned with these kernels are shown in Figure 4. The GPs are conditioned on the same data but with different kernels. The RBF kernel produces a very smooth GP, the Matérn kernels are less smooth, and the ESS kernel is periodic.

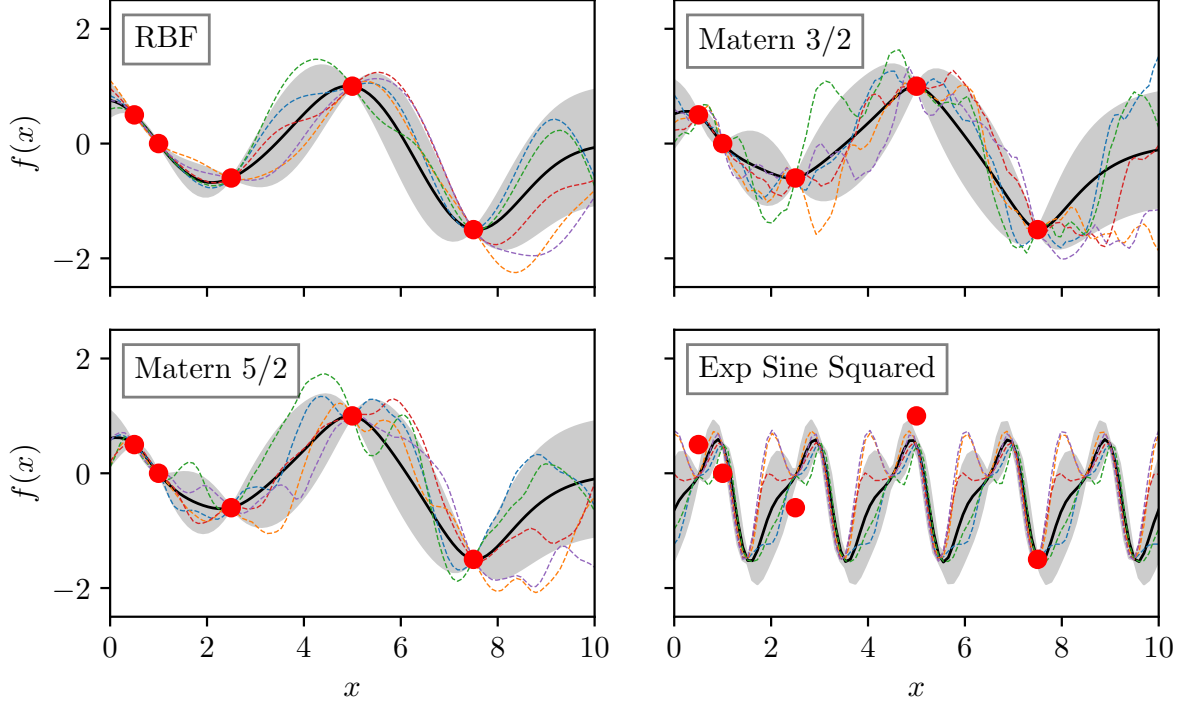


Figure 4: GPs with four different kernels: RBF (top left), Matérn $\nu = 3/2$ (top right), Matérn $\nu = 5/2$ (bottom left), and ESS (bottom right). The GPs are conditioned on the data shown as red dots. The solid black line is the GPs conditioned mean, and the standard deviation is depicted as the gray shaded region. The Matérn kernels are less smooth than the RBF kernel, and the ESS kernel is periodic.

Kernels can be combined to create more complex kernels, allowing for greater flexibility in modeling diverse data patterns. For example, the sum and product of two valid kernels are also valid kernels [2].

4.3 Extension to Higher Dimensions

Extending the kernel (and hence the GP) to higher dimensions is straightforward and done by promoting the kernel to a function of $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, where d is the dimensionality of the data.

For stationary kernels such as the RBF kernel, this can be achieved by either using the same length scale for all dimensions (promoting the scalar distance r to the Euclidean distance $\mathbf{r} = |\mathbf{x} - \mathbf{x}'|$) or by using a different length scale for each dimension. The latter is called an *anisotropic* kernel, which offers greater flexibility at the cost of additional hyperparameters. The anisotropic RBF kernel is defined as:

$$k^{\text{RBF},d}(\mathbf{X}_i, \mathbf{X}'_i) = C^2 \exp \left(- \sum_{k=1}^d \frac{(\mathbf{X}_{ik} - \mathbf{X}'_{ik})^2}{2l_k^2} \right), \quad (17)$$

where \mathbf{l} is a vector of length scales for each dimension.

5 Choosing the Kernel's Hyperparameters

The kernel typically has one or more free hyperparameters, denoted as $\boldsymbol{\lambda} = \{\lambda_1, \dots, \lambda_{n_\lambda}\}$. These can be determined in a Bayesian way through the likelihood of the data given the GP [5]:

$$\begin{aligned}\mathcal{L}^{\text{GP}}(\mathbf{y}|\mathbf{X}, \boldsymbol{\lambda}) &= \int p(\mathbf{y}|\mathbf{f}, \mathbf{X}, \boldsymbol{\lambda})p(\mathbf{f}|\mathbf{X}, \boldsymbol{\lambda})d\mathbf{f} \\ &= -\frac{1}{2}\mathbf{y}^T(\mathbf{K} + \sigma_n^2\mathbb{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K} + \sigma_n^2\mathbb{I}| - \frac{n}{2}\log 2\pi,\end{aligned}\tag{18}$$

where \mathbf{y} is the vector of training data and \mathbf{K} is the Gram matrix of the training set \mathbf{X} . Unfortunately, evaluating this likelihood involves the computationally expensive matrix inversion of the Gram matrix. This makes full Bayesian inference of $\boldsymbol{\lambda}$ infeasible in practice. Instead, one typically finds the maximum likelihood estimate, otherwise known as *Maximum a Posteriori* (MAP) or *Maximum Likelihood type II* estimate of $\boldsymbol{\lambda}$ using a gradient-based optimization algorithm.

6 The GP Regression Algorithm

Putting all the ingredients together, one arrives at the full GP regression algorithm. GP regression (sometimes also referred to as *Kriging*) involves fitting a GP with kernel $k(\mathbf{x}, \mathbf{x}'|\boldsymbol{\lambda})$ to a training set \mathbf{X}, \mathbf{y} with associated noise σ_n . This is done by finding the MAP estimate of $\boldsymbol{\lambda}$ by maximizing the marginal GP log-likelihood in Equation 18. This approach is convenient because the only choice left to the user is that of a suitable kernel function. Furthermore, and in contrast to most neural networks, the hyperparameters of the GP typically have an intuitive interpretation.

The algorithm consists of two simple steps:

1. In the *training* step, the hyperparameters of the model are optimized by maximizing Equation 18. Additionally, $(\mathbf{K} + \sigma_n^2\mathbb{I})^{-1}$ is precomputed for use in the prediction step.
2. In the *prediction* step, the GP is conditioned according to Equations 10 and 11.

The algorithm is divided into these two steps to highlight the distinction between the computationally expensive $\mathcal{O}(N^3)$ training phase and the relatively cheap $\mathcal{O}(N^2)$ prediction phase, where N is the number of points in the training set.

7 Conclusion

This tutorial has introduced the fundamental concepts of Gaussian Process regression. We have covered:

- The mathematical foundation of GPs as distributions over functions
- How to condition GPs on observed data to make predictions
- The role of kernel functions and their properties
- Common kernel functions (RBF, Matérn, ESS)
- The GP regression algorithm and hyperparameter optimization

GPs provide a powerful and flexible framework for regression tasks, with the key advantages of providing uncertainty estimates and having a solid mathematical foundation. The main challenge in using GPs is choosing an appropriate kernel function that reflects prior knowledge about the problem at hand.

References

- [1] Noel A. C. Cressie, editor. *Statistics for spatial data*. Wiley series in probability and mathematical statistics Applied probability and statistics. Wiley, New York, rev. ed edition, 2010. ISBN 978-1-119-11515-1.
- [2] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014. URL <https://www.repository.cam.ac.uk/handle/1810/247281>.
- [3] John Lamperti. *Stochastic Processes: A Survey of the Mathematical Theory*. Number v.23 in Applied Mathematical Sciences Ser. Springer New York, New York, NY, 1997. ISBN 978-1-4684-9358-0.
- [4] Kevin P. Murphy. *Machine Learning - A Probabilistic Perspective*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, 2014. ISBN 978-0-262-01802-9.
- [5] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9.
- [6] Bernt K. Øksendal. *Stochastic differential equations: an introduction with applications*. Universitext. Springer, Berlin Heidelberg New York Dordrecht London, sixth edition, sixth corrected printing edition, 2013. ISBN 978-3-642-14394-6.