Your Paper

You

August 1, 2023

Abstract

1 Introduction

2 Fundamentals

2.1 Gaussian Process

As defined in [6] a Gaussian Process is a collection of random variables for which every finite set of variables have a joint multivariate Gaussian distribution. For a Regression model we define a function $f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ with the input set $\boldsymbol{X} = \{\boldsymbol{x_n} \in \mathcal{X}\}_{n=1}^{N}$. We now let $f : \mathcal{X} \to \mathbb{R}$ be our Gaussian Process. To fully define the GP we need to specify a mean function $m(\boldsymbol{x})$ and a covariance function or kernel $k(\boldsymbol{x}, \boldsymbol{x'})$ for $f(\boldsymbol{x})$:

$$\begin{aligned} m(\boldsymbol{x}) &= \mathbb{E}[f(\boldsymbol{x})] \\ k(\boldsymbol{x}, \boldsymbol{x'}) &= \mathbb{E}[(f(\boldsymbol{x}) - m(\boldsymbol{x}))(f(\boldsymbol{x'}) - m(\boldsymbol{x'}))] \end{aligned} \tag{1}$$

In this work the mean function $m(\boldsymbol{x})$ of the GP is set to zero. This is quite common, mainly for notational reasons [6].
So what we are left with for $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x'})$ is:

$$\begin{aligned} m(\boldsymbol{x}) &= 0 \\ k(\boldsymbol{x}, \boldsymbol{x'}) &= \mathbb{E}[f(\boldsymbol{x})f(\boldsymbol{x'})] \end{aligned} \tag{2}$$

We now write our Gaussian Process $f(\boldsymbol{x})$ as:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x'})) \tag{3}$$

It is important to note that $k(\boldsymbol{x}, \boldsymbol{x'})$ has to be a positive definite in order to be a valid kernel function. Because of the definition of the GP we can now generate a joint Gaussian distribution for a finite set of points $\boldsymbol{X_*}$:

$$p(\boldsymbol{f_*}|\boldsymbol{X_*}) = \mathcal{N}(\boldsymbol{f_*}|\boldsymbol{0}, \boldsymbol{K_*}) \tag{4}$$

with $[\boldsymbol{K}]_{i,j} = k(\boldsymbol{x_{i*}}, \boldsymbol{x_{j*}})$.

2.2 Regression

The goal of regression is given a training dataset $\mathcal{D} = \{(\boldsymbol{x_i}, y_i)|i = 1, \ldots, N\}$ we want to make predictions for new test points $\boldsymbol{x_*}$ which are not in the training set. There are several methods which can be used to tackle this kind of problem. For example the basic linear regression using least squares and specific basis functions or a neural network that does regression. In this work we make use of the Gaussian process which is a non-parametric model. We will start with a Gaussian Process prior and then use the rules for Gaussian multivariate distributions to define a posterior predictive distribution, which we can then use to make predictions for new test points.

The general setup for regression is that we make observations/measurements y of an underlying function $f(\boldsymbol{x})$ at points $\boldsymbol{x}$ which are corrupted by a zero mean Gaussian noise with variance σ_n^2:

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X}) + \mathcal{N}(0|\sigma_n^2\boldsymbol{I}) \tag{5}$$

When we set the GP Prior onto the function f we have to add the normally distributed noise. To get the prior for the observations $\boldsymbol{y}$ we simply need to change the covariance function to $\operatorname{cov}(\boldsymbol{y_p}, \boldsymbol{y_q}) = k(\boldsymbol{x_p}, \boldsymbol{x_q}) + \sigma_n^2\delta_{pq}$.
As described before in equation 4 we can generate a Gaussian random vector for new test data $\boldsymbol{X_*}$. Both $\boldsymbol{y}$ and $\boldsymbol{f_*}$ are now multivariate Gaussians. We can therefore introduce the joint distribution of the observed data and the values of f at the test points $\boldsymbol{X_*}$ as:

$$\begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{f_*} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} K(\boldsymbol{X}, \boldsymbol{X}) + \sigma_n^2\boldsymbol{I} & K(\boldsymbol{X}, \boldsymbol{X_*}) \\ K(\boldsymbol{X_*}, \boldsymbol{X}) & K(\boldsymbol{X_*}, \boldsymbol{X_*}) \end{bmatrix}\right) \tag{6}$$

From this joint Gaussian prior we can then calculate the predictive posterior distribution by applying the rules for conditioning Gaussians. A good description for this conditioning can be found in [1] (chapter 2.3.1). The predictive distribution is then given by:

$$
\begin{aligned}
p(\boldsymbol{f_*}|\boldsymbol{X},\boldsymbol{y},\boldsymbol{X_*}) = \mathcal{N}(&\boldsymbol{f_*}|\boldsymbol{\mu_*},\boldsymbol{\Sigma_*}) \\
\boldsymbol{\mu_*} &= K(\boldsymbol{X_*},\boldsymbol{X})[K(\boldsymbol{X},\boldsymbol{X})+\sigma_n^2\boldsymbol{I}]^{-1}\boldsymbol{y} \\
\boldsymbol{\Sigma_*} &= K(\boldsymbol{X_*},\boldsymbol{X_*}) - K(\boldsymbol{X_*},\boldsymbol{X})[K(\boldsymbol{X},\boldsymbol{X})+\sigma_n^2\boldsymbol{I}]^{-1}K(\boldsymbol{X},\boldsymbol{X_*})
\end{aligned}
\tag{7}
$$

We can see that with the training data

2.3 Kernels

The general covariance function used in this work is the squared exponential covariance function (SE) as defined in [6]:

$$k(\boldsymbol{x_p},\boldsymbol{x_q}) = \sigma_f^2 \exp(-0.5(\boldsymbol{x_p}-\boldsymbol{x_q})^\intercal M(\boldsymbol{x_p}-\boldsymbol{x_q})) \tag{8}$$

with the variance hyperparameter σ_f^2 and the matrix $M = \mathrm{diag}(\boldsymbol{l})^{-2}$ containing the vector with the characteristic length scales l_i. The hyperparameters are collected in the vector $\boldsymbol{\theta} = (\boldsymbol{l}, \sigma_f^2)$.
For the one dimensional case the SE covariance function simply becomes:

$$k(x_p, x_q) = \sigma_f^2 \exp\left(-\frac{0.5}{l^2}(x_p - x_q)^2\right) \tag{9}$$

For a two-dimensional case with time being the second dimension, two independent length scale parameters are used for the separate space and time domains $\boldsymbol{l} = (l_x, l_t)$. For this case, we can re-write equation 7 the covariance function with $\boldsymbol{x_p} = (x_p, t_p)^\intercal$ and $\boldsymbol{x_q} = (x_q, t_q)^\intercal$ as the product of two single SE covariance functions:

$$k(\boldsymbol{x_p},\boldsymbol{x_q}) = \sigma_f^2 \exp\left(-\frac{0.5}{l_x^2}(x_p - x_q)^2\right) \exp\left(-\frac{0.5}{l_t^2}(t_p - t_q)^2\right) \tag{10}$$

There are many kernels one can choose from. A good overview over some kernel functions can be found in [2] as well as in [4].

2.4 Marginal likelihood

$$\log p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) = -\frac{1}{2}\boldsymbol{y}^\intercal K_y^{-1}\boldsymbol{y} - \frac{1}{2}\log|K_y| - \frac{n}{2}\log(2\pi) \tag{11}$$

with $K_y = K_f + \sigma_n^2\boldsymbol{I}$.

3 Physics informed framework

Our problem setup is the same as proposed in [5]. We have some sort of linear differential equation or ODE of a physical model:

$$\mathcal{L}_{\boldsymbol{x}}^{\phi} u(\boldsymbol{x}) = f(\boldsymbol{x}) \tag{12}$$

with $\mathcal{L}_{\boldsymbol{x}}^{\phi}$ being the differential operator that resembles the differential equation and ϕ being a list containing the a-priori unknown parameters which also define the differential equation.

We now start by assuming that $u(x)$ is a Gaussian Process of the form:

$$u(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, k_{uu}(\boldsymbol{x},\boldsymbol{x'};\boldsymbol{\theta})) \tag{13}$$

with $k_{uu}(\boldsymbol{x},\boldsymbol{x'};\boldsymbol{\theta})$ being the kernel depending on its hyperparameters θ for the $\mathcal{GP}$ (on) $u(\boldsymbol{x})$. With the known linear relation between u and f from equation 12, we can now say that $f(\boldsymbol{x})$ must also be $\mathcal{GP}$ [6] of form:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(\boldsymbol{0}, k_{ff}(\boldsymbol{x},\boldsymbol{x'};\boldsymbol{\theta})) \tag{14}$$

with now k_{ff} being the kernel defining the $\mathcal{GP}$ (on) $f(\boldsymbol{x})$. As shown in [3], [7] and as proposed in [5]

3.1 Optimization/Finding of the hyperparameters

4 Results

4.1 Heat Equation

The one dimensional time dependent heat equation with a forcing term is:

$$\frac{\partial u(x,t)}{\partial t} - \alpha \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t) \tag{15}$$

with α as the thermal diffusivity.

4.2 Wave Equation

$$\frac{1}{c^2} \frac{\partial^2 u(x,t)}{\partial t^2} - \frac{\partial^2 u(x,t)}{\partial x^2} = f(x,t) \tag{16}$$

4.3 Damped Oscillator

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* 2006.

[2] D. Duvenaud. Automatic model construction with gaussian processes. 2014.

[3] Roman Garnett. *Bayesian Optimization.* Cambridge University Press, 2023.

[4] Jochen Görtler, Rebecca Kehlbeck, and Oliver Deussen. A visual exploration of gaussian processes. *Distill*, 2019. https://distill.pub/2019/visual-exploration-gaussian-processes.

[5] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using gaussian processes. *Journal of Computational Physics*, 348:683–693, 2017.

[6] Carl Edward Rasmussen. *Gaussian processes for machine learning.* Adaptive computation and machine learning. 2006.

[7] Simo Särkkä. Linear operators and stochastic partial differential equations in gaussian process regression. pages 151–158, 06 2011.