Gaussian Process Regression

Vassili Papavassiliou, Stephen Pate-Morales,
Abinash Pun, Forhad Hossain, Dinupa Nawarathne,
Harsha Kaluarachchi

October 01, 2022

Abstract

Gaussian process regression (GPR) is a widely used learning technique in machine learning. Some of the basic concepts of Gaussian process (GP) are multivariate normal distributions, kernels and joint and conditional probability etc. In this note we explain the basic mathematics of GPR and one of the common library used in the python to make GPR predictions.

1 Introduction

The Gaussian process model is a probabilistic supervised machine learning technique used in classification and regression tasks. A Gaussian process regression (GPR) model can make predictions incorporating prior knowledge (kernels) and provide uncertainties of the predictions (*1*).

The advantages of the Gaussian process are (*2*);

- The prediction interpolates the observations (at least for regular kernels).
- The prediction is probabilistic (Gaussian) so that one can compute empirical confidence intervals and decide based on those if one should refit (online fitting, adaptive fitting) the prediction in some region of interest.
- Versatile: different kernels can be specified. Common kernels are provided, but it is also possible to specify custom kernels.

2 Mathematical Basics

Consider set of observed data points. We want to fit a function to represent these data points and then make a prediction at new data points. This is know as the regression. For a given set of observed data points, there are infinite number of possible functions that fit these data points. In GPR, Gaussian process conduct the regression by defining a distribution over these infinite number of functions.

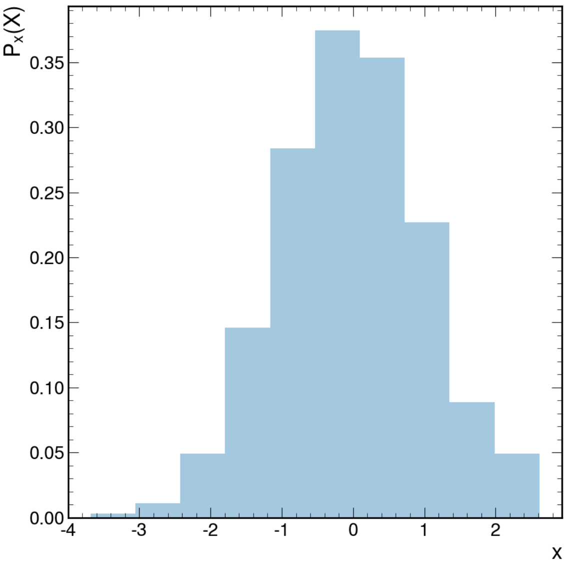

Figure 1: A uni-variate normal distribution for 1000 random points.

2.1 Gaussian Distribution

A random variable X is Gaussian or normally distributed with mean μ and variance σ^2 if its probability density function(PDF) is (*3*);

$$P_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{1}$$

where X is the random variables and x is the real argument. The normal distribution of X is represented by;

$$P_X \sim \mathcal{N}(\mu, \sigma^2) \tag{2}$$

2.2 Multivariate Normal Distribution

The PDF of an multivariate normal distribution (MVN) with dimension D is defined as (*3*);

$$\mathcal{N}(x|\mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp[-\frac{1}{2}(x-\mu)^T \sigma^{-1}(x-\mu)] \tag{3}$$

where D is the number of dimensions, x is the variable, μ is the mean vector and Σ is the covariance matrix. Consider the Gaussian random variable

$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$, mean $\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$ and covariance matrix $\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$. We have the following properties;

1. Normalization;

$$\int_y p(y;\mu,\Sigma) = 1 \tag{4}$$

2. Marginalization: The marginal distributions $p(y_1) = \int_{y_2} p(y_1, y_2; \mu, \Sigma)dy_2$ and $p(y_2) = \int_{y_1} p(y_1, y_2; \mu, \Sigma)dy_1$;

$$y_1 \sim \mathcal{N}(\mu_1, \Sigma_{11}) \tag{5}$$

$$y_2 \sim \mathcal{N}(\mu_2, \Sigma_{22}) \tag{6}$$

3. Summation: If $y_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $y_2 \sim \mathcal{N}(\mu_2, \Sigma_2)$, then;

$$y_1 + y_2 \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2) \tag{7}$$

4. Conditioning: The conditional distribution of y_1 on y_2;

$$p(y_1|y_2) = \frac{p(y_1, y_2; \mu, \Sigma)}{\int_{y_1} p(y_1, p_2; \mu, \Sigma)dy_1} \tag{8}$$

is also Gaussian;

$$y_1|y_2 = y_2 \sim \mathcal{N}(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}, \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}) \tag{9}$$

This property will be useful in deriving Gaussian process predictions.

2.3 Kernels

We need to have a smooth function to define the covariance matrix. This can be done by covariance functions.If a function is defined solely in terms of inner products in the input space, then the kernel function $k(x, x')$ is a kernel function. One of the most common used kernel function is radial basis function (RBF) kernel;

$$k(x_i, x_j) = \exp(-\frac{(x_i - x_j)^2}{2l^2}) \tag{10}$$

where l is a free parameter and $x_i - x_j$ is the Euclidean distance.

3 Gaussian Process

Consider an infinite dimensional function f, A Gaussian process(GP) is a collection of random variables (RV) such that the joint distribution of every finite subset of RVs is multivariate Gaussian;

$$f \sim GP(\mu, k) \tag{11}$$

where $\mu(x)$ and $k(x, x')$ are the mean and covariance function respectively.

3.1 Gaussian Process Regression

Consider the following properties of Σ (*4*);

1. $\Sigma_{ij} = E((Y_i - \mu_i)(Y_j - \mu_j))$.
2. Σ is always positive semi-definite.
3. $\Sigma_{ii} = \text{Variance}(Y_i)$, thus $\Sigma_{ii} > 0$.
4. If Y_i and Y_j are vary independent. i.e. x_i is very different from x_j, then $\Sigma_{ij} = \Sigma_{ji} = 0$.
5. If x_i is similar to x_j, then $\Sigma_{ij} = \Sigma_{ji} > 0$.

Let $\Sigma_{ij} = k(x_i, x_j)$, then we can decompose Σ as $\begin{bmatrix} K & K_* \\ K_*^T & K_{**} \end{bmatrix}$, where K is the training kernel matrix, K_* is the training-testing kernel matrix, K_*^T is the testing-training kernel matrix and K_{**} is the testing kernel matrix. Then the conditional distribution of values of the function f can be written as;

$$f_*|(Y_1 = y_1, ..., Y_n = y_n, x_1, ..., x_n) \sim \mathcal{N}(K_*^T K^{-1} y, K_{**} - K_*^T K^{-1} K_*) \tag{12}$$

where the kernel matrices K_*, K_{**}, K are function of $x_1, ..., x_n, x_*$.

4 Software

One of the most commonly used machine leaning library in `python` is `sci-kit learn` (*2*). `GaussianProcessRegressor` is the in build class in the `sklearn` library. This library also contains the several in build kernel function that can used in the regression. Any user defined kernel functions can be defined using these kernel functions.

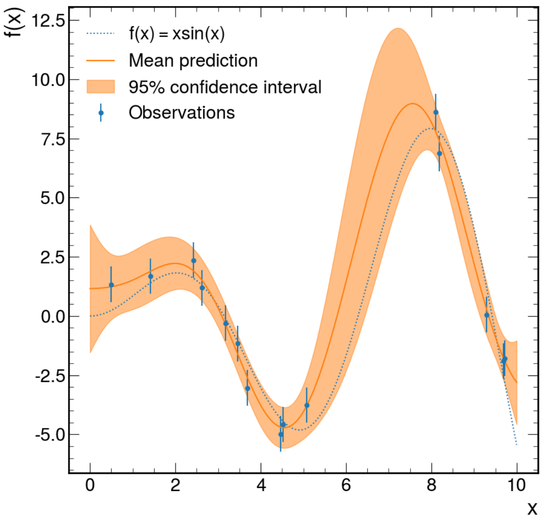

Figure 2: Prediction using GPR method with sklearn.

4.1 Example

Let's consider a distribution of the form $x\sin(x)$ with some random noise. Out goal is to get the mean prediction and the prediction error using `sklearn` library with GPR method. We use the RBF kernel as the covariance function to make this prediction. Here GPR class was trained with 9 iterations before making the perdition. Error of the prediction is simply the square root of the diagonal values of the covariance function.

According to figure 2 predition error is relatively minimum in the region $2. < x < 4.$. This is because we have more observations in that region hence the small predition error. In the region $6. < x < 8.$ we have a relative large error because we have very few observations to make a prediction. This will cause the large prediction error.

In RBF kernel class length scale and length scale bounds are the two parameters we can used to optimize in the GPR. Some of the common parameter we can optimize in GPR class is kernel, alpha value added to the diagonal of the kernel in the training, optimizer and number of restarts of the optimizer. More details about this class can be found in `sklearn` documentation. Some of the in build kernels is given in their documentation.

Some disadvantages of the Gaussian process include (*2*);

- They are not sparse, i.e., they use the whole samples/features information to perform the prediction.

- They lose efficiency in high dimensional spaces – namely when the number of features exceeds a few dozens.

References

1. C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning* (The MIT Press, Nov. 2005), ISBN: 9780262256834, (https://doi.org/10.7551/mitpress/3206.001.0001).
2. F. Pedregosa *et al.*, *Journal of Machine Learning Research* **12**, 2825–2830 (2011).
3. K. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, 2012), ISBN: 9780262018029, (https://books.google.com/books?id=NZP6AQAAQBAJ).
4. J. Wang, *arXiv preprint arXiv:2009.10862* (2020).