Machine Learning Club, Summer Induction Assignment, 2023 - Q3

1 What is a Kernel?

In the context of machine learning and Gaussian Process regression, a kernel is a function that measures the similarity between pairs of data points. It allows us to quantify how much influence one data point has on another. Kernels play a crucial role in Gaussian Process models as they determine the smoothness, flexibility, and generalization capabilities of the model.

A kernel function takes two inputs, often denoted as x and x', and produces a similarity measure or a distance metric between them. It maps the inputs into a higher-dimensional feature space, where the similarity or distance can be computed more easily. The choice of the kernel function depends on the specific problem and the underlying assumptions about the data.

In Gaussian Process regression, the kernel function defines the covariance structure of the Gaussian Process prior over the function values. It quantifies the similarity or correlation between different points in the input space. A positive-semidefinite kernel ensures that the resulting covariance matrix is positive-semidefinite, which is essential for the well-definedness of Gaussian Processes.

The Radial Basis Function (RBF) kernel, also known as the squared exponential kernel, is a commonly used kernel in Gaussian Process regression. It captures smooth variations in the data and allows the model to interpolate between observed data points while providing good extrapolation properties. The RBF kernel is defined by the formula:

$$k(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right)$$

where σ^2 represents the variance and l is the length scale parameter.

Understanding the properties and construction of kernel functions, such as positive-semidefiniteness and their impact on the model's behavior, is crucial for effectively using Gaussian Process regression and other machine learning techniques.

2 Conditional Property of Multivariate Gaussian Distribution

Given a joint distribution of multivariate Gaussian random variables $\mathbf{X} = \begin{bmatrix}\mathbf{X}_1 \\ \mathbf{X}_2\end{bmatrix}$, where $\mathbf{X}_1$ represents a subset of variables and $\mathbf{X}_2$ represents the remaining variables, the conditional distribution of $\mathbf{X}_1$ given $\mathbf{X}_2 = \mathbf{x}_2$ can be derived.

Let $\boldsymbol{\mu} = \begin{bmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{bmatrix}$ and $\boldsymbol{\Sigma} = \begin{bmatrix}\boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22}\end{bmatrix}$ represent the mean vector and covariance matrix of $\mathbf{X}$, respectively.

The conditional distribution of $\mathbf{X}_1$ given $\mathbf{X}_2 = \mathbf{x}_2$ is given by:

$$p(\mathbf{X}_1|\mathbf{X}_2 = \mathbf{x}_2) = \mathcal{N}(\boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2})$$

where

$$\boldsymbol{\mu}_{1|2} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)$$
$$\boldsymbol{\Sigma}_{1|2} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$$

The conditional property of the multivariate Gaussian distribution allows us to compute the conditional distribution of a subset of variables given the values of the remaining variables. This property is useful in various machine learning algorithms, such as Gaussian Process regression, where we can infer the distribution of unobserved variables given observed variables.

3 Derivation of RBF Kernel

To construct the Radial Basis Function (RBF) kernel, also known as the squared exponential kernel, we start with the assumption that the function values, denoted as $F(x)$, are drawn from a Gaussian Process.

The RBF kernel is defined as:

$$k(x, x') = \sigma^2 \exp\left(-\frac{\|x - x'\|^2}{2l^2}\right)$$

where σ^2 represents the variance and l is the length scale parameter.

To derive this kernel, we start by considering the covariance between two function values $F(x)$ and $F(x')$:

$$\text{cov}(F(x), F(x')) = k(x, x')$$

Now, let's assume a dataset with n input points, denoted as $X = \{x_1, x_2, \ldots, x_n\}$, and corresponding function values $F = \{F(x_1), F(x_2), \ldots, F(x_n)\}$. We can stack the function values into a column vector $\mathbf{f} = [F(x_1), F(x_2), \ldots, F(x_n)]^\top$.

Using the kernel matrix notation, we can write the covariance matrix as $\mathbf{K} = [k(x_i, x_j)]_{i,j=1}^n$.

To incorporate noise in the observed function values, we introduce the noise variance σ_n^2 and add it to the diagonal of $\mathbf{K}$:

$$\mathbf{K}' = \mathbf{K} + \sigma_n^2 \mathbf{I}$$

where $\mathbf{I}$ is the identity matrix.

The next step is to use the Cholesky decomposition which needs a Positive Semi Definite Matrix (PSD) to decompose $\mathbf{K}'$ as:

$$\mathbf{K}' = \mathbf{L}\mathbf{L}^\top$$

where $\mathbf{L}$ is a lower-triangular matrix.

By assuming the prior distribution of $\mathbf{f}$ as $\mathbf{f} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}')$, we can calculate the posterior predictive distribution at a test point x_*.

Let $\mathbf{f}_*$ represent the vector of function values at the test points, and $\mathbf{f}$ denote the observed function values. Then, the joint distribution of $\mathbf{f}$ and $\mathbf{f}_*$ can be written as:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}' & \mathbf{K}_* \\ \mathbf{K}_*^\top & \mathbf{K}_{**} \end{bmatrix}\right)$$

where $\mathbf{K}_*$ represents the covariance matrix between observed and test points, and $\mathbf{K}_{**}$ is the covariance matrix of test points.

Using the conditional distribution properties of multivariate Gaussian distributions, we can calculate the posterior predictive distribution as:

$$\mathbf{f}_* | \mathbf{f}, X, x_* \sim \mathcal{N}(\mathbf{K}_*^\top \mathbf{K}'^{-1} \mathbf{f}, \mathbf{K}_{**} - \mathbf{K}_*^\top \mathbf{K}'^{-1} \mathbf{K}_*)$$

4 Bonus Part

This is purely on my intuition from what I have read, but here I am giving it a shot.

1. Extrapolation: The mean on the extrapolation dataset, based on my observations during experiments, shows that tree-based models perform well for points near the test dataset. However, their performance significantly deteriorates for points that are far from the dataset.

   One key observation is that the uncertainty in predictions is considerably high for extrapolation scenarios.

2. Optimization: Gaussian Processes (GPs) do not optimize like traditional Neural Networks, which use back-propagation for their optimization, as GPs do not involve parameters and have a closed-form solution.

   However, for GPs, we can optimize using various different kernel functions. Furthermore, a very important optimization technique is restarting with different hyperparameters. For example, we can optimize the sigma in the RBF kernel, as it controls the smoothness and length scale of the kernel. It determines how fast the similarity between two data points decreases as their distance increases.