Gaussian Processes for Machine Learning

Summary

In this tutorial paper, Carl E. Rasmussen gives an introduction to Gaussian Process Regression focusing on the definition, the hyperparameter learning and future research directions.

A Gaussian process is completely defined by its mean function $m(\boldsymbol{x})$ and its covariance function (kernel) $k(\boldsymbol{x}, \boldsymbol{x}')$. The mean function $m(\boldsymbol{x})$ corresponds to the mean vector $\boldsymbol{\mu}$ of a Gaussian distribution whereas the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ corresponds to the covariance matrix $\boldsymbol{\Sigma}$. Thus, a Gaussian Process $f \sim \mathcal{GP}\left((m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right)$ is a generalization of a Gaussian distribution over vectors to a distribution over functions. A random function vector $\mathbf{f}$ can be generated by a Gaussian Process through the following procedure:

1. Compute the components μ_i of the mean vector $\boldsymbol{\mu}$ for each input $\boldsymbol{x}_i$ using the mean function $m(\boldsymbol{x})$
2. Compute the components Σ_{ij} of the covariance matrix $\boldsymbol{\Sigma}$ using the covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$
3. A function vector $\mathbf{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)]^T$ can be drawn from the Gaussian distribution $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$

Applying this procedure to regression, means that the resulting function vector $\mathbf{f}$ shall be drawn in a way that a function vector $\mathbf{f}$ is rejected if it does not comply with the training data $\mathcal{D}$. This is achieved by conditioning the distribution on the training data $\mathcal{D}$ yielding the posterior Gaussian Process $f|\mathcal{D} \sim \mathcal{GP}(m_D(\boldsymbol{x}), k_D(\boldsymbol{x}, \boldsymbol{x}')$ for noise-free observations with the posterior mean function $m_D(\boldsymbol{x}) = m(\boldsymbol{x}) + \boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{x})^T \boldsymbol{\Sigma}^{-1}(\mathbf{f} - \mathbf{m})$ and the posterior covariance function $k_D(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{x}')$ with $\boldsymbol{\Sigma}(\boldsymbol{X}, \boldsymbol{x})$ being a vector of covariances between every training case of $\boldsymbol{X}$ and $\boldsymbol{x}$.

Noisy observations $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ can be taken into account with a second Gaussian Process with mean m and covariance function k resulting in $f \sim \mathcal{GP}(m, k)$ and $y \sim \mathcal{GP}(m, k + \sigma_n^2 \delta_{ii'})$. The attached figure illustrates the cases of noisy observations (variance at training points) and of noise-free observation (no variance at training points).

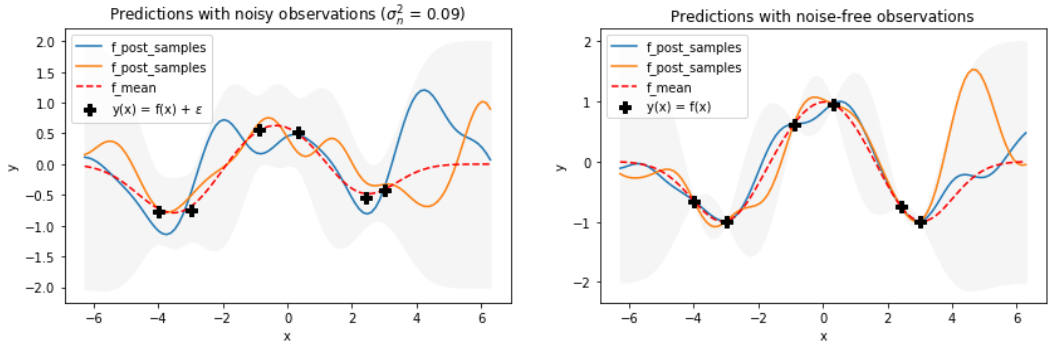

Figure 1: Gaussian Process Regression

In the Machine Learning perspective, the mean and the covariance function are parametrised by hyperparameters and provide thus a way to include prior knowledge e.g. knowing that the mean function is a second order polynomial. To find the optimal hyperparameters $\boldsymbol{\theta}$,

1. determine the log marginal likelihood $L = \log(p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}))$,
2. take the first partial derivatives of L w.r.t. the hyperparameters, and
3. apply an optimization algorithm.

It should be noted that a regularization term is not necessary for the log marginal likelihood L because it already contains a complexity penalty term. Also, the tradeoff between data-fit and penalty is performed automatically.

Gaussian Processes provide a very flexible way for finding a suitable regression model. However, they require the high computational complexity $\mathcal{O}(n^3)$ due to the inversion of the covariance matrix. In addition, the generalization of Gaussian Processes to non-Gaussian likelihoods remains complicated.