Start: 24.6.2024
Return: 8.7.2024
https://classroom.github.com/a/u2pnyuhH

Assignment 6
Introduction to Probabilistic Machine Learning

1 Introduction

In this assignment, you will implement a Gaussian Process regression model and use Gradient Descent to optimize its hyperparameters.

As usual, the repository contains all files needed. This exercise does not contain any theoretical questions. A Jupyter notebook is provided, that runs your code and applies it to a regression problem.

Part 1: Fundamental Gaussian Processes (4 Points)

- Relevant source files: `GP.jl`
- Notebook for code execution: `GP_visually.ipynb`
- Test files: `tests.jl`

The GP file contains a framework for Gaussian processes. In this assignment, it is your task to bring this framework to life. `train_gp` is the main entry point for the computation of all variables required for inference. Those must be stored in a struct of the type `GaussianProcess`, which includes input variables to the `train_gp` function and variables that must be computed in the process.

Two helper functions should assist you in that computation: `kernel_mat` and `rbf_kernel`. The latter should compute the value of the kernel between two inputs given its hyperparameters. `kernel_mat` should compute the kernel matrix, a.k.a. the element-wise kernel values between two input arrays. The RBF kernel function for one-dimensional feature vectors is given by:

$$C(x, x') = \sigma_f^2 \cdot \exp\left(-\frac{(x - x')^2}{2\lambda^2}\right) + \sigma_n^2 \cdot \mathbb{I}(x = x')$$

Two functions are left for the task: `predict_gp` should use a GP object and an input array to output predictions for every input value. `log_m_likelihood` should compute log marginal likelihood given an instance of a GP.

The code after the scaffold includes the logic to automatically tune the hyperparameters via gradient descent. In addition to the tests, you can use the provided notebook to observe the results of your code.

Part 2: Evaluation and Hyperparameter Optimization (1 Points)

- Relevant source files: `GP_visually.ipynb, GP.jl`
- Test files: `tests.jl`

When everything is set up correctly, you can use the notebook, to use a Gaussian process for regression on artificial toy data as well as a dataset of measurements of daylight time (sun hours) per year at different measurement locations in Germany. The experiment using the daytime dataset still includes a few gaps. Take a look at the artificial example and use your observations to set the correct parameters for the real-world dataset. Discuss your results with your tutor :).