Data-Driven Design & Analysis of Structures & Materials (3dasm)

Instructor: Miguel A. Bessa

Brown University

Homework 6

Deliver **a short PDF report** of this assignment containing the answers to the questions listed here. **UPLOAD to CANVAS in the Assignments section (Homework 6) by the due date.**

Due date: until 11:59pm of day announced in course home page.

This Homework aims to help you reason about Gaussian Processes by writing and running code, as well as by reflecting on the content of Lectures 14, 15 and 16. This is a **computational homework**, where you need to include the Python code used to solve it – **we advise you to solve it in a Jupyter Notebook and printing it as PDF including the answers**.

Exercise 1

1. Conceptual questions.
   1.1. Why is it important to appropriately scale the datasets for most machine learning models?
   1.2. What is scikit-learn's StandardScaler? Include the expression that defines this scaling procedure.
   1.3. What is scikit-learn's MinMaxScaler? Include the expression that defines this scaling procedure.
   1.4. Hyperparameters in Gaussian processes are usually determined by maximizing the (log) marginal likelihood, instead of doing Bayesian model selection. Why is it difficult to do Bayesian model selection for Gaussian processes (and many other models)?

Exercise 2

Solve the in-class Exercise of Lecture 14 (but now, you do it!)

2. Consider (again!) the function to be learned the same that we used in previous homeworks: $f(x) = x\sin(x)$ within the domain $x \in [0, 10]$.
   2.1. Using Gaussian Process Regression with a kernel that results from multiplying a Constant kernel (with hyperparameter s^2) by the RBF kernel (with hyperparameter l):

$$k(x_i, x_j) = s^2 \exp\left(-\frac{||x_i - x_j||^2}{2l^2}\right)$$

   with an initial guess for the hyperparameters of $s = 1$ and $l = 10$.

   Recreate the first plot (same dataset, same seed number, same train/test split) of that lecture but changing the domain bounds from $x \in (0, 10)$ to $x \in (-10, 20)$ so that you can see the model extrapolating.

   2.2. Now make the same plot but consider non-negligible aleatoric uncertainty for the training data, i.e. considering that the noise at each training point is $\sigma^2 = 2.5^2$.
   2.3. Revert back to the case with **negligible aleatoric uncertainty at the input points**, i.e. $\alpha = 1e-10$, and to **the original domain** $x \in (0, 10)$. **Turn off the hyperparameter optimizer** (optimizer = None), and see the effect of particular choices of hyperparameter l for the above mentioned kernel by creating the following plots:
      2.3.1. For fixed hyperparameters $s^2 = 1.0$ and $l = 10$.
      2.3.2. For fixed hyperparameters $s^2 = 1.0$ and $l = 1$.

2.3.3. For fixed hyperparameters $s^2 = 1.0$ and $l = 0.1$.

2.3.4. Based on the above 3 plots explain why the hyperparameter l is called the length scale hyperparameter or "roughness" parameter.

2.4. Now **consider the larger domain** $x \in (-10, 20)$ so that you can clearly see the extrapolatory regime. Still considering the hyperparameter optimizer turned off (optimizer = None), now investigate the effect of the hyperparameter s^2 in the above mentioned kernel by creating the following plots:

2.4.1. For fixed hyperparamaters $s^2 = 1.0$ and $l = 1$.

2.4.2. For fixed hyperparamaters $s^2 = 2.5^2$ and $l = 1$.

2.4.3. For fixed hyperparamaters $s^2 = 4^2$ and $l = 1$.

2.4.4. Based on the above 3 plots explain why s^2 is called the variance hyperparameter.

2.5. Now "turn on" the hyperparameter optimization again (optimizer='fmin_l_bfgs_b') and train the model. Comment on the quality of extrapolation and note what happens to the mean and the 95% confidence interval.

2.6. Finally, consider different kernels and plot the approximation using the same training data but considering two different kernels[1]:

2.6.1. Considering the Exponential-Sine-Squared kernel (only this kernel; do not multiply by the RBF).

2.6.2. Considering the Matérn 3-2 kernel.

2.6.3. How does the approximation compare with the one obtained with the Constant Kernel multiplied by the RBF kernel. Why is the ContantKernel*RBF kernel better in this case?

Exercise 3 (BONUS[2] question)

3. Multidimensional regression with GPR (Lecture 15).

3.1. Create the following datasets without noise:

- Go to the following website: `https://www.sfu.ca/~ssurjano/optimization.html`. Then, **select the first noiseless test function from each of the 6 categories** in the website, i.e. select the following functions: (1) Ackley function; (2) Bohachevsky Functions; (3) Booth Function; (4) Three-Hump Camel Function; (5) De Jong Function N. 5; and (6) Beale Function. **Consider the domain of analysis as suggested in the website for each function**, and also note that all of them have **two-dimensional inputs** and **one-dimensional output**.
- Randomly sample 60 points (do this for each of the 6 functions you selected). Note that the functions will be defined on different input domain bounds, so you should randomly sample 60 points every time you define a new function.
- Split each dataset in two sets (training and testing sets) using the "train_test_split" function of scikit-learn and consider 80% of the data is included in the training set. Set the "random_state" seed to the value 123.
- You don't need to create a pandas dataframe for each dataset.
- Consider the same kernel as in Exercise 2 (Constant kernel times RBF) with the same hyperparameters.
- (Although there should be no need to say this...) Make sure you scale the datasets accordingly.

3.2. Report the R^2 and MSE metrics in a table for the 6 functions selected. You do not need to show the plots for all the functions, just the performance metrics.

[1]Lecture 14 already has some examples of similar kernels, although that part is commented out. However, make sure you use only the ExpSineSquared (without multiplying by the RBF kernel) for Exercise 2.6.1.

[2]This question is not mandatory. Only do the BONUS questions if you finished all regular questions. These questions award additional points (up to a top mark of 100% in the Homework)

3.3. Identify a function that appears to be more difficult to learn and another that is particularly easy. Show plots of these two functions and provide plausible explanations for this behavior[3].

[3] Please note that you will be making conclusions based on simple experiments, using only one kernel and without considering a large number of training points. So, your conclusions should not be definitive. But this exercise will help you gain intuition about the GPR method.