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

Instructor: Miguel A. Bessa

Brown University

Homework 4

Deliver **a short PDF report** of this assignment containing the answers to the questions listed here. **UPLOAD to CANVAS in the Assignments section (Homework 4) by the due date.**

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

This Homework aims to help you reason about Linear Regression by writing and running code similar to what the one we covered on Lectures 8 and 9. 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. Explain from a Bayesian perspective the Linear Least Squares model and why do we use it to fit data with polynomial basis functions.
   1.2. What is a point estimate?
   1.3. Explain what are training, validation and testing sets.
   1.4. Define and explain the R^2 metric.
   1.5. Define and explain what is the MSE metric.
   1.6. Explain what is k-fold cross-validation.

Exercise 2

Consider the function to be learned as $f(x) = x\sin(x)$ within the domain $x \in [0, 10]$.

2. Linear regression model with Linear Least Squares for dataset without noise.
   2.1. Create a dataset without noise from the $f(x) = x\sin(x)$ function. **Differently from what was done in class**, create the dataset as follows:
   - Randomly sample 40 points from $f(x)$ within the defined domain ($x \in [0, 10]$). Use seed 123 for the pseudo-random generator.
   - After sampling the points, sort them in ascending order (from smaller to larger x).
   - Save the dataset as a pandas dataframe and write it in the existing folder called "your_data", with a file name of "HW4_noiseless_dataset.csv".
   - Split the 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 123.
   - **Important**: make sure that you use the same training and testing sets for fitting all models for this question as well as the remaining ones.

   2.2. Calculate the R^2 and MSE on the testing set after training Linear Regression models with polynomials of different degree (1, 3, 5, 10, 20) using three different training set sizes (respectively with the first 6, 11 and 21 points of the training set defined in 2.1.). Present the previously mentioned error metrics as suggested in Table 1 and show 3 figures with the y–x plots of the polynomial approximations with different degree considering 6, 11 and 21 training points (one figure per row of the table). **What can you conclude from these results**?

Table 1: Suggested table to report R^2 (similar for MSE) in each cell. **(NOTE: You do not need to present the values as in this table. You can present them in a different table arrangement, as long as you show the same information.)**

n_{train} \ degree	1	3	5	10	20
6					
11					
21					

2.3. Now consider the **entire training set** defined in 2.1. and use 6-fold cross validation to calculate the **mean** and **standard deviation** of the R^2 and MSE metrics when training Linear Regression models with polynomials of different degree (1, 5, 20). Present the result as suggested in Table 2. **Explain the obtained mean and standard deviation for the error metrics**.

Table 2: Suggested table to report R^2 and MSE using cross-validation where each cell contains their mean and standard deviation. **(NOTE: You do not need to present the values as in this table. You can present them in a different table arrangement, as long as you show the same information.)**

Error metric \ degree	1	5	20
MSE	mean ± std	mean ± std	mean ± std
R^2	mean ± std	mean ± std	mean ± std

Exercise 3

3. Linear regression model with Linear Least Squares for dataset **with** noise.

3.1. Use the dataset you created in 2.1. and perturb the output values of each point **with** noise. Consider the same type of noise used in Lecture 9[1].

- Once you generated the new dataset (same x values but perturbed y values), save it as a pandas dataframe **including the noise you used at each point, i.e. the standard deviation of the Gaussian noise**. Create (in the existing folder called "your_data") a corresponding file called "HW4_noisy_dataset.csv".
- As before, split the 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 123, i.e. **the same seed you used in 2.1.**

3.2. Repeat 2.2. but now for the noisy dataset created in 3.1.

3.3. Repeat 2.3. but now for the noisy dataset created in 3.1.

Exercise 4

4. Linear regression model with Ridge Regression for dataset **with** noise.

4.1. Explain from a Bayesian perspective what is Ridge Regression and why do we use it to fit data with polynomials.

4.2. Repeat 2.2., i.e. considering a **noiseless dataset**, but now using **Ridge Regression** with hyperparameter $\alpha = 10^{-4}$. What happens when you use a bigger or smaller value of α?

4.3. (**BONUS**[2] **question**) Use grid search and 4-fold cross-validation to determine a good alpha hyperparameter for Ridge regression when considering a **polynomial of degree 5**. The grid should

[1] Noise was created for each point from a Gaussian distribution whose standard deviation at that point comes from a random value between 0.5 and 1.5

[2] This question and the next are 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)

span the following alphas: $\alpha = [100, 10, 1, 0.1, 0.01, 0.001, 0.0001, 0]$. Report the improvement when compared to 4.2.

4.4. (**BONUS question**) Repeat 4.3. but now considering a grid search with 4-fold cross-validation that also involves the degree of the polynomial, i.e. the grid should span the following different alphas and different polynomial degrees: $\alpha = [100, 10, 1, 0.1, 0.01, 0.001, 0.0001, 0]$ and degree $= [1, 3, 5, 10, 20]$