The Mathematics of Machine Learning
Homework Set 2

Due 2 March 2023 before 13:00
via Canvas

You are allowed to work on this homework in pairs. One person per pair submits the answers via Canvas. Make sure to put both names on the submission.

1 Programming Exercise

The following programming exercise is to be implemented in Python, using a Jupyter notebook.

We will be predicting life expectancy in a country based on several demographic variables. The data come from the Kaggle machine learning competition website.

- Read the description of the data on `https://www.kaggle.com/kumarajarshi/life-expectancy-who` and download the data as `Life Expectancy Data.csv`.
- You will extend the `Homework2-start.ipynb` Jupyter notebook, which is available from the course website. This notebook already contains the code to read in and prepare the data. Read through it and make sure you understand how the data have been prepared.
- The notebook already shows how to apply least squares regression and the Lasso to the data, but the hyperparameter for the Lasso (called λ in the lecture but α in `sklearn`) has yet to be set to a sensible value *without looking at the test set.*
- You may either answer the questions below in a separate document or by inserting text into the notebook directly.

1. We will proceed assuming that the life expectancy data form independent, identically distributed (i.i.d.) samples from a fixed distribution.

   (a) In the data preparation, we keep only one year (2014) per country. Explain how the i.i.d. assumption would be violated if we kept all years for each country.

(b) As is common in applications, it is debatable whether the data are fully i.i.d., even with the current data preparation. Give at least one reason pro or con the applicability of the i.i.d. assumption. *Hint: consider whether it is natural for i.i.d. samples of countries that we do not see any repetitions of the same country in our data.*

2. In the notebook, the train error for least squares is much lower than its test error. For the Lasso with hyperparameter $\alpha = 1$, the train and test error are approximately equal. How can this be explained?

3. Use 10-fold cross-validation to select the hyperparameter α for the Lasso from the following set of candidate values: $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 1, 2, 3, 4, 5, 10\}$. *You have to implement cross-validation yourself, so you cannot use any of the `sklearn` functions for cross-validation!* NB. The data have already been shuffled, so you do not need to do that yourself.

   Hint: the notebook uses 'pandas' arrays. To index elements of a pandas array by their location in the array (as opposed to their label) use `.iloc`, as described here: `https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#indexing-integer`.

4. Given a choice $\hat{\alpha}$ for the hyperparameter, there are two ways to come up with a final predictor:

   - Average the predictions of the 10 predictors with parameter $\hat{\alpha}$ that you trained during cross-validation.
   - Train a new Lasso predictor with $\alpha = \hat{\alpha}$ on all the training data.

   Implement both of these.

   (a) Compare their average loss on the test data. Also compare to the average loss of the least squares predictor. Explain what you see.

   (b) Compare the sparsity (number of coefficients that are non-zero) for both predictors. What do you see? Can you explain this?