Machine Learning

Spring Semester 2023
Prof. Dr. Peter Zaspel

Assignment Sheet 4.

Submit on **Tuesday, March 7, 2023, 10:00**.

Excercise 1. (Calculating EPE and regressor)

In this task, we revise Examples 3.2 and 3.3 from the lecture notes for a different setup:
Let $X : \Omega \to \mathbb{R}$ be an input variable and $Y : \Omega \to \mathbb{R}$ be an output variable. For the input variable we assume that it follows the uniform distribution as $X \sim \mathcal{U}[-1, 1]$. (Note the other range!) Moreover, we make the very strong assumption to know the "true" dependency between X and Y. Specifically we define Y via

$$Y := g(X), \quad \text{with} \quad g(x) = e^x \,.$$

Now we look for a function f that shall approximate that (usually unknown) relationship between X and Y. We claim that

$$f(x) = 1 + x \,,$$

is a good approximation to $Y = g(X)$.

a) Calculate the expected (squared) prediction error for f. For now in $f(x)$ we used linear terms only, what would be the best quadratic approximation to the relationship between X and Y?

b) Find the regressor for $Y = g(x)$.

(4 Points)

Excercise 2. (Regressor derivation)

In the lecture slides, you have seen that the expected (squared) prediction error $EPE(f)$ is given by

$$EPE(f) = E(L_2(Y, f(X)) \,.$$

Theorem 3.1 then states that the function f that minimizes the expected (squared) prediction error $EPE(f)$ is given by

$$f(x) = E(Y|X = x)$$

Note that in the middle of the proof, we encounter an equation

$$\begin{aligned} EPE(f) = E_X E_{Y|X}[(f(X) - E[Y|X])^2 + 2(f(X) - E[Y|X])(E[Y|X] - Y) \\ + (E[Y|X] - Y)^2 | X] \,, \end{aligned}$$

where the second term can be dropped. Now, prove that this is possible, i.e. that it holds

$$2E_X E_{Y|X}[(f(X) - E[Y|X])(E[Y|X] - Y)|X] = 0 \,.$$

(4 Points)

Excercise 3. (kNN regression)

You are given the following training data:

$$\mathcal{T} = \left\{ \left((2,1)^\top, 19\right), \left((2,3)^\top, 21\right), \left((4,3)^\top, 22\right), \left((5,2)^\top, 11\right), \left((5,4)^\top, 15\right), \left((4,6)^\top, 12\right), \left((3,4)^\top, 32\right), \left((1,5)^\top, 12\right), \left((7,8)^\top, 16\right) \right\}$$

Manually carry a kNN regression prediction for $\mathbf{x}_1 = (7,4)^\top$, $\mathbf{x}_2 = (3,4)^\top$, $\mathbf{x}_3 = (6,2)^\top$ and $k = 1$, $k = 3$, $k = 6$. As part of the task, you have to draw the points in a scatter plot (on paper) and mark the respective neighborhoods that contribute to the final result.

(4 Points)

Programming Exercise 1. Consider the Examples 3.4 and 3.5 from the lecture, for which you also have access to the source code. Complete the following tasks:

a) (Re-)implement Example 3.4. This time, however, you need to implement the kNN regression by yourself, without a machine learning library and without a kNN search library. (If you implement in Python, just start from the available Jupyter notebook.) Verify the correctness of your implementation by cross-checking it with Example 3.4.

b) Apply your implementation to the Energy efficiency Data Set from the UCI Machine Learning Repository. Build the predictor for the required heating load on the full data set and predict the load on the first three samples of the data set and $k = 1$, $k = 3$, $= 10$.

c) (Re-)implement Example 3.5. This time, however, you need to implement the kNN classification by yourself, without a machine learning library and without a kNN search library. (If you implement in Python, just start from the available Jupyter notebook.) Verify the correctness of your implementation by cross-checking it with Example 3.5.

d) Now, we would like to use kNN classification for SPAM classification. Either you collect some e-mails (both SPAM and no SPAM) from your Inbox, label them manually and create the representation using your implementation from last week or you use the Spambase Data Set from the UCI Machine Learning Repository. Apply your implementation to one of these data sets and evaluate the predictor for three random samples in the data set and values $k = 1, k = 3, k = 10$. Compare these results to the training data.

Reference solutions will only be provided in Python+Matplotlib. The submission format for Python is a Jupyter notebook. The submission format for C/C++ is standard source files. Choose an appropriate format for the Gnuplot-related submission.

(4 Points)