Artificial Intelligence and Machine Learning

Exercises – Support Vector Machines

Question 1 (Computing a support vector machine by hand)

The aim of this question is to compute a hard-margin support vector machine (SVM) by hand. For this, let a dataset consisting of the two training examples

$$\left(\boldsymbol{x}^1 := \begin{pmatrix}-2 & -1\end{pmatrix}^\top, y_1 := +1\right) \qquad \text{and} \qquad \left(\boldsymbol{x}^2 := \begin{pmatrix}1 & 1\end{pmatrix}^\top, y_2 := -1\right)$$

be given. *Admittedly, this dataset is not very useful in practical applications, but using such a small dataset makes it feasible to work through the computations by hand. This enhances your understanding of support vector machines.*

Please work through the following tasks to train the SVM:

1. Write down the hard-margin SVM dual optimization problem for the dataset above.

2. State the *Karush-Kuhn-Tucker (KKT)* conditions for this optimization problem. Are the KKT conditions sufficient for a solution in this case? Is the solution unique?

3. Compute the optimal Lagrange multipliers α_1 and α_2 by solving the KKT system which you have specified in task 2.

4. Compute the optimal model parameters $\boldsymbol{w}$ and b.

5. Let the test example $\boldsymbol{x}' := \begin{pmatrix}-1/2 & 1\end{pmatrix}^\top$ be given. Use the model parameters you have computed in task 4 to classify this new example.

Question 2 (Implementing a hard-margin SVM)

Implement a linear hard-margin SVM (i. e. without kernel functions and slack) and classify the linearly separable dataset generated by the code snippet below. Use the Python package `cvxopt` for the optimization. `cvxopt` uses a custom data type called `cvxopt.matrix`, i. e. you have to convert all `Numpy` arrays to that data type. The optimization can be done with `cvxopt.solvers.qp(...)`.

```
import numpy as np

def make_linear():
    X = np.asarray([
        [3.00, 1.00], [3.20, 2.20], [3.15, 4.80],
        [3.35, 1.20], [3.05, 3.50], [3.55, 2.85],
        [1.50, 2.25], [2.88, 2.18], [1.95, 4.00],
        [3.01, 2.95], [2.85, 3.01], [5.85, 2.20],
```

```
            [4.19, 4.00], [5.15, 3.50], [5.07, 2.89],
            [4.87, 3.54], [4.44, 3.78], [4.48, 3.94],
            [5.51, 3.80]
        ])

        y = np.asarray([
            -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1,
            1, 1, 1, 1, 1, 1, 1, 1
        ])

        return X, y


    X, y = make_linear()
```

Plot the decision boundary and highlight the support vectors.

Question 3 (Kernels and feature maps)

Consider the following kernel functions and determine the corresponding feature map.
Let $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^2$ and $c \in \mathbb{R}$.

1. $k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z})^2$
2. $k(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x}^\top \boldsymbol{z} + c)^2$