Machine Learning:
Assignment Sheet #11

Due on May 2, 2022 at 10:00

Group HB
Henri Sota, Enis Mustafaj

Problem 11.1

In this paper & pencil task, you compare the quadratic model based on the basis functions $\{1, X, X^2\}$ to the linear model for the following training set.

$$\mathcal{T} = \{(-1, -1), (0, -4), (2, 2)\}$$

a) Compute the linear model using least squares for the given training data.

Firstly, we create the matrix $\mathcal{X}$:

$$\mathcal{X} = \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 2 \end{pmatrix} \qquad \mathcal{X}^\top = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \end{pmatrix} \qquad y = \begin{pmatrix} -1 \\ 4 \\ 2 \end{pmatrix}$$

By using the least squares, we compute the coefficients of the model:

$$\alpha = (\mathcal{X}^\top \cdot \mathcal{X})^{-1} \cdot \mathcal{X}^\top \cdot y$$

$$\alpha = \left(\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 \\ 1 & 0 \\ 1 & 2 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} -1.4285714285714286 \\ 1.28571429 \end{pmatrix}$$

So the derived model is:

$$f_\alpha(\mathbf{X}) = -1.4285714285714286 + 1.28571429 \cdot \mathbf{X}$$

b) Compute the quadratic model using least squares for the given training data.

Firstly, we create the matrix $\mathcal{X}$:

$$\mathcal{X} = \begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{pmatrix} \qquad \mathcal{X}^\top = \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \\ 1 & 0 & 4 \end{pmatrix} \qquad y = \begin{pmatrix} -1 \\ 4 \\ 2 \end{pmatrix}$$

By using the least squares, we compute the coefficients of the model:

$$\alpha = (\mathcal{X}^\top \cdot \mathcal{X})^{-1} \cdot \mathcal{X}^\top \cdot y$$

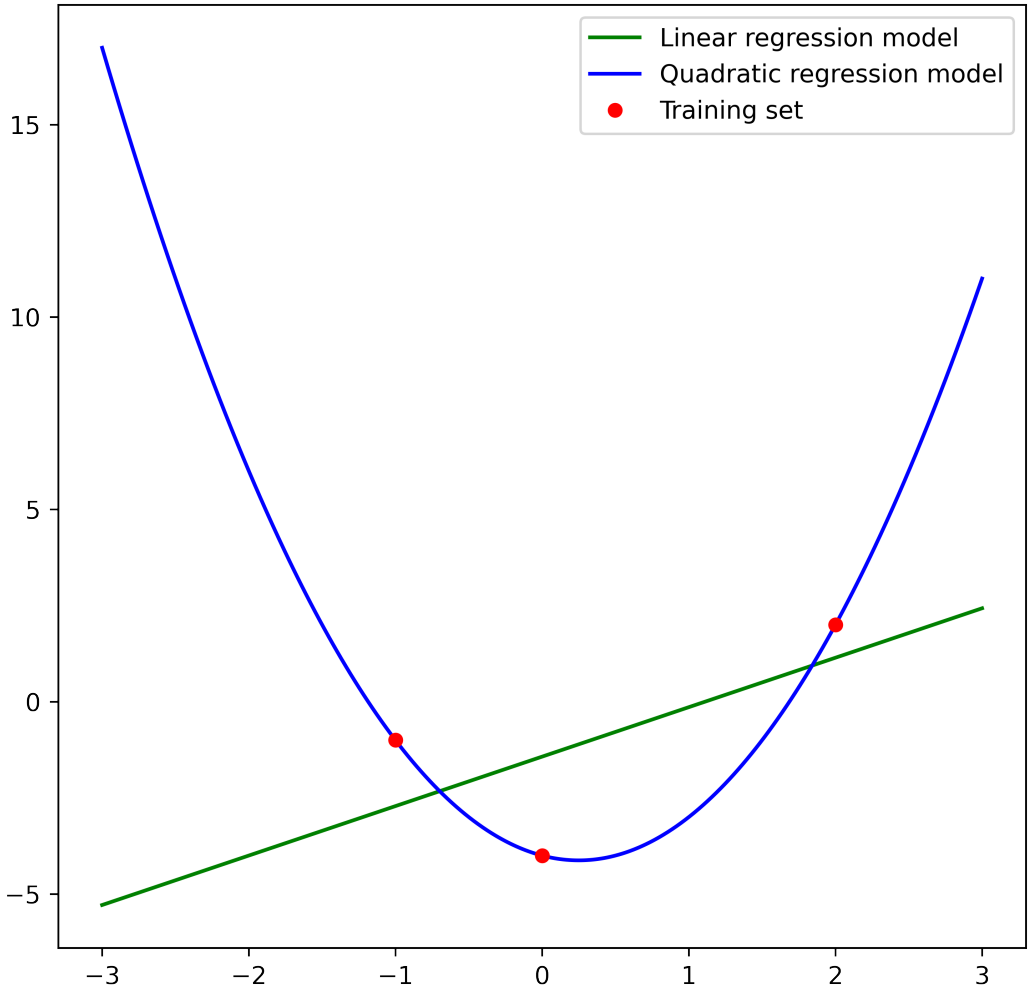
$$\alpha = \left(\begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \\ 1 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & 1 \\ 1 & 0 & 0 \\ 1 & 2 & 4 \end{pmatrix} \right) \cdot \begin{pmatrix} 1 & 1 & 1 \\ -1 & 0 & 2 \\ 1 & 0 & 4 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 4 \\ 2 \end{pmatrix} = \begin{pmatrix} -4 \\ -1 \\ 2 \end{pmatrix}$$

So, the derived model is:

$$f_\alpha(\mathbf{X}) = -4 - \mathbf{X} + 2 \cdot \mathbf{X}^2$$

c) Draw (by hand) the linear and the quadratic model in a plot, together with the training samples.

The graph of linear and quadratic model is represented as following:



Problem 11.2

You are given the training data

$$\mathcal{T} = \{(-2, 2)^\top, (0, 0)^\top, (1, 2)^\top\}$$

a) Use the Gaussian kernel with kernel width $\sigma = 1$ and compute by hand the kernel-based model using least squares for the given training data.

Firstly, we calculate the Gaussian kernel for each of the input points:

$$k(-2, 0) = 0.0183156389 \qquad k(-2, 1) = 0.000123409804 \qquad k(0, 1) = 0.367879441$$

The Gaussian kernel of a point with itself is 1 since the distance between a point and itself is 0. The Gaussian kernel is also symmetric, so the kernel of the opposite points related to above ones will be the same. The kernel matrix is as following:

$$\mathcal{A} = \begin{pmatrix} 1 & 0.0183156389 & 0.000123409804 \\ 0.0183156389 & 1 & 0.367879441 \\ 0.000123409804 & 0.367879441 & 1 \end{pmatrix}$$

Now we calculate the coefficient vector by using the formula:

$$\alpha = \mathcal{A}^{-1} \cdot y$$

$$\alpha = \begin{pmatrix} 1 & 0.0183156389 & 0.000123409804 \\ 0.0183156389 & 1 & 0.367879441 \\ 0.000123409804 & 0.367879441 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 2.01607799 \\ -0.89351756 \\ 2.32845794 \end{pmatrix}$$

The derived model is:

$$f_\alpha(\mathbf{X}) = 2.01607799 \cdot k(X, -2) - 0.89351756 \cdot k(X, 0) + 2.32845794 \cdot k(X, 1)$$

b) Repeat the previous task, but this time you compute by hand the kernel-based model using ridge regression with regularization parameter $\lambda = 1$.

We calculate the coefficient vector by the formula:

$$\alpha = (\mathcal{A} + \lambda \cdot \mathcal{I})^{-1} \cdot y$$

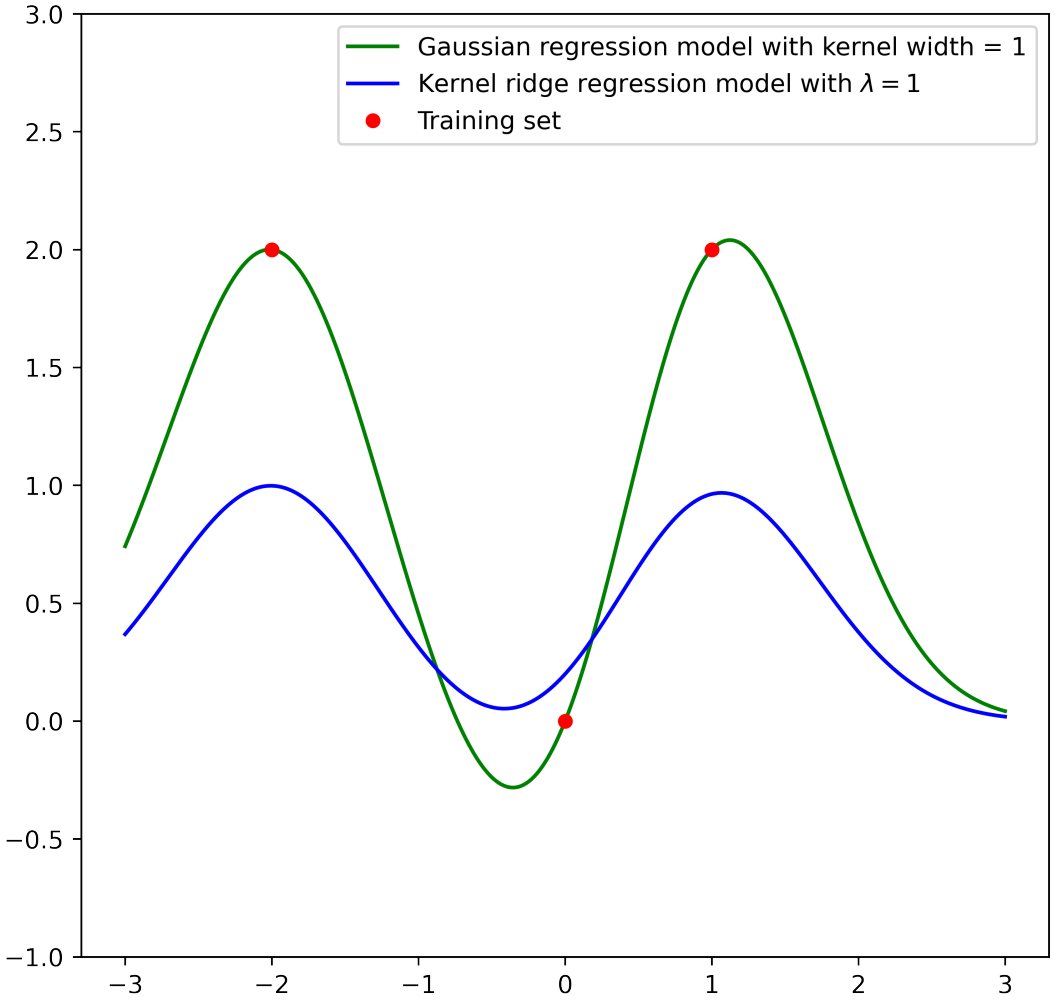
$$\alpha = \left(\begin{pmatrix} 1 & 0.0183156389 & 0.000123409804 \\ 0.0183156389 & 1 & 0.367879441 \\ 0.000123409804 & 0.367879441 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 2 \\ 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 1.00176635 \\ -0.19986453 \\ 1.03670121 \end{pmatrix}$$

The derived model is:

$$f_\alpha(\mathbf{X}) = 1.00176635 \cdot k(X, -1) - 0.19986453 \cdot k(X, 0) + 1.03670121 \cdot k(X, 1)$$

c) Draw both above models including the training data in one plot.

The graphs of both models are represented as following:



Problem 11.3

Prove Lemma 10.2 from the lecture notes.
Recalling Lemma 10.2:
Let $\{\boldsymbol{x}_i\}_{i=1}^N$ be a set of points such that $\boldsymbol{x}_i \in \mathbb{R}^D$ and let $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ be a positive definite kernel. Then, the matrix

$$\mathcal{A} = \begin{pmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{pmatrix}$$

is (symmetric) positive definite.

A kernel is a symmetric function. Therefore, the matrix $\mathcal{A}$ is symmetric. The underlying properties of a symmetric matrix are that the eigenvalues are real and the eigenvectors are perpendicular to each other.

To prove the positive definite property of the matrix $\mathcal{A}$, the quadratic form can be used. Quadratic form states that if for all $\boldsymbol{x}! = 0$, the product $x^T \mathcal{A} x$ is greater than 0, then the matrix is positive definite.

We choose to use the orthonormal basis, $\{e_1, \dots, e_n\}$, of the eigenvectors corresponding to the eigenvalues, $\{\lambda_1, \dots, \lambda_n\}$ of matrix $\mathcal{A}$. We can express any x using the orthonormal basis, $\boldsymbol{x} = c_1 e_2 + \dots + c_n e_n$.

The quadratic form can then be expressed as:

$$\begin{aligned} \boldsymbol{x}^T \mathcal{A} \boldsymbol{x} &= (c_1 e_1^T + \dots + c_n e_n^T) \mathcal{A} (c_1 e_1 + \dots + c_n e_n) \\ &= (c_1 e_1^T + \dots + c_n e_n^T)(c_1 \mathcal{A} e_1 + \dots + c_n \mathcal{A} e_n) \\ &= (c_1 e_1^T + \dots + c_n e_n^T)(c_1 \lambda_1 e_1 + \dots + c_n \lambda_n e_n) \\ &= c_1 e_1^T \cdot c_1 \lambda_1 e_1 + c_1 e_1^T \cdot c_2 \lambda_2 e_2 + \dots c_n^2 \cdot e_n^T \lambda_n e_n \\ &= c_1^2 \lambda_1 + 0 + \dots + c_n^2 \lambda_n \\ &= c_1^2 \lambda_1 + \dots + c_n^2 \lambda_n \end{aligned}$$

Since $\boldsymbol{x}$ is not equal to the null vector, one of the constants will at least be non-zero.

Programming Problem 11.1

In this task, we implement kernel ridge regression. To this end, we start from Example 10.6 from the lecture notes, which is available as Jupyter notebook. However, instead of using scikit-learn to compute the ridge regression model, you replace that part of the code by your implementation of kernel ridge regression, which follows Algorithm 17 from the lecture notes. Due to the existing implementation, it will be easier to evaluate, whether your implementation is correct.