

Submission Date: **12.11.2025**

## 1 Support Vector Machines

1. Explain the “kernel trick” and why we use it in SVMs.
2. The SVM problem can be considered as optimizing the balance between the average hinge loss over the examples and the regularization term that keeps the parameters small (increasing the margin). This balance is set by the regularization term  $\lambda > 0$ . Here, we focus on the case without the offset parameter  $w_0$  (setting it to zero), hence the training objective will become:

$$\frac{1}{n} \sum_{i=1}^n \mathcal{L}(y_i, \mathbf{w}^T \mathbf{x}_i) + \frac{\lambda}{2} \sum_{j=1}^m \mathbf{w}_j^2$$

where the hinge loss is given by

$$\mathcal{L}(y_i, \mathbf{w}^T \mathbf{x}_i) = \max(0, 1 - y_i(\mathbf{w}^T \mathbf{x}_i))$$

- (a) Compute the gradient of the training objective w.r.t.  $\mathbf{w}$  and describe the pseudocode for the gradient descent algorithm.
- (b) Use gradient descent to update the weights based on the following example:

$$\lambda = 0.5, \eta = 0.01, y = 1, \mathbf{x} = \begin{bmatrix} 1.0 \\ 0.0 \end{bmatrix}, \mathbf{w} = \begin{bmatrix} 1.0 \\ 1.0 \end{bmatrix}$$

## 2 Linear Separability

Given the following dataset:

<hr/>	<hr/>	<hr/>
<hr/>	<hr/>	<hr/>
-3	-1	
-2	-1	
-1	-1	
0	1	
1	1	
2	-1	
3	-1	

1. Create a sketch of the data. Is it linearly separable? If so, draw a separating hyperplane.
2. Apply the mapping  $g : \mathbb{R} \mapsto \mathbb{R}^2$  defined by:

$$g(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix} \quad (1)$$

on all the data points to create the transformed data set and create a plot of it. Is the transformed data set linearly separable? If yes, find a separating hyperplane  $\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$ , compute its parameters<sup>1</sup> and plot it.

3. Use the computed hyperplane from 2. and compute its output for  $x = \frac{1+\sqrt{5}}{2}$ . Does it belong to the positive or negative class and why?

### 3 Linear and Non-Linear SVM

 **Using Python and SciKit-Learn:** Included in the exercise is a file containing data about patients with diabetes (`diabetes.csv`). Additionally, there is a jupyter notebook that contains some code snippets to help you get started. The notebook requires a virtual environment using anaconda or venv with numpy, pandas, sklearn, jupyter and matplotlib, seaborn.

1. Load, analyze and pre-process the given data:
  - Handle missing values and outlier data
  - Normalize features
  - Create train and test data splits so that 80% of the data is used for training and the remaining 20% is used for testing.
2. Train a linear SVM model and measure its accuracy.
3. Train a non-linear SVM using a Radial Basis Function (RBF) and measure its accuracy.
4. **Next mile:** Use grid search to find the best hyperparameters among different values for the regularization parameter  $C \in \{0.1, 1, 10, 100\}$  and various types of kernels (RBF, linear, polynomial, sigmoid).

---

<sup>1</sup>It is fine if you use the border points only, i.e. only active constraints