MAS374 OPTIMIZATION THEORY
HW#3 (PROGRAMMING)

The least squares (LS) problem

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \|A\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 \tag{1}$$

is one of the rare instances of optimization problems where an optimal solution can be found in a closed form. More specifically, when A is full column rank the optimal solution of (1) is

$$\boldsymbol{\theta}^* = (A^\top A)^{-1}A^\top \boldsymbol{y} = A^\dagger \boldsymbol{y}. \tag{2}$$

Problem 1. Least squares problems can also be used in classification. Suppose we have n data points $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)} \in \mathbb{R}^d$, and say that each $\boldsymbol{x}^{(i)}$ is labeled by some number $y_i \in \mathbb{R}$. Our goal is to classify the points $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$ according to their label. In this problem, we assume that $y_i \in \{\pm 1\}$, i.e. there are only two classes that a data can belong to.

Given a function $f : \mathbb{R}^d \to \mathbb{R}$, let us consider the value $f(\boldsymbol{x}^{(i)})$ as a prediction of the label y_i. Then of course, we want the predictions to be not so different from the actual labels. One way of achieving this is to minimize the *sum of squares*

$$\sum_{i=1}^{n} \left| f(\boldsymbol{x}^{(i)}) - y_i \right|^2. \tag{3}$$

Now the goal is to find the optimal fuction f that minimizes (3). However we don't want f to be an arbitrary function. Indeed, any function that assigns each $\boldsymbol{x}^{(i)}$ to y_i will minimize (3). A widely used restriction on f is asserting it to be a polynomial of degree less than a certain number.

In this problem—so that you can easily visualize your results—let $d = 2$, and let us classify the given data with a bivariate polynomial of degree 2, i.e. by a polynomial of the form

$$f(\boldsymbol{x}) = \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \theta_4 x_1^2 + \theta_5 x_1 x_2 + \theta_6 x_2^2$$

where $\theta_1, \ldots, \theta_6$ are constants to be determined.

(a) Write a function `my_lstsq(A, y)` which takes A and $\boldsymbol{y}$ as its input and uses the SVD of A to compute and return the optimal solution $\boldsymbol{\theta}^*$ of the LS problem (1).
(Hint: The MATLAB function `svd` or the NumPy function `numpy.linalg.svd` will be helpful.)

(b) Generate 250 random samples $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(250)}$ uniformly from $[-2, 2] \times [-2, 2] \in \mathbb{R}^2$. For each sample $\boldsymbol{x}^{(i)} = (x_1^{(i)}, x_2^{(i)})$, label the point according to the rule

$$y_i = \begin{cases} +1 & \text{if } (x_1^{(i)})^2 + (x_2^{(i)})^2 \le 1 \\ -1 & \text{otherwise} \end{cases}. \tag{4}$$

Convert the problem of minimizing (3) over $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_6]^\top$ into an LS problem, and use `my_lstsq` to compute the optimal polynomial that minimizes (3). The curve $f(\boldsymbol{x}) = 0$ can be viewed as the *decision boundary*, a guideline that separates the whole space $\mathbb{R}^2$ into regions according to the predicted labels for each point. **Make a guess on how the decision boundary would look like, and verify it. Give reasons for your guess and observations.**

(c) This time, generate 250 random samples uniformly from $[0, 2] \times [0, 2] \in \mathbb{R}^2$. Label the sample points according to the rule (4). **Make a guess on the decision boundary. What is the optimal polynomial that minimizes (3) in this case? Does the result coincide with your intuition? Explain your answers.**

Caution:

- Fill out the provided code template appropriately, and submit it through KLMS.
- You can choose either using MATLAB or using Python 3 (with NumPy).
- You may define your own functions in the code if you need.
- With the code, submit a report with answers to the questions written in **bold** in (b) and (c).
- 10 points each are assigned to the code and the report.

Date: October 7th, 2021.