EE2211: Spring 2023
Assignment 1 (5%)
Submission: 23:59 on 17th Feb 2023 (Friday of Week 6)

In this assignment, we are interested in using Python to solving a weighted least squares (WLS) problem. Compared to ordinary least squares that minimizes the mean squared error, the WLS problem instead assigns a unique weight α_i to each sample and minimizes the *weighted mean squared error*

$$\mathrm{WMSE}(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \alpha_i (\mathbf{x}_i^\top \mathbf{w} - y_i)^2,$$

where $(\mathbf{x}_i, y_i)$ (for $1 \le i \le m$) represents a training sample and its target, $\alpha_i \in \mathbb{R}$ (usually positive) is the *weight* assigned to sample i and $\mathbf{w} = [w_1, w_2, \ldots, w_d]^\top \in \mathbb{R}^d$ are the parameters we want to estimate. As in lecture, we can stack the training samples and targets into a matrix (known as the *design matrix*) and vector respectively. We denote these as

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_m^\top \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

For example for $d = 2$ and $m = 5$, we may have the design matrix, target vector respectively as

$$\mathbf{X} = \begin{bmatrix} 1 & 4 \\ 4 & 2 \\ 5 & 6 \\ 3 & -3 \\ 9 & -10 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} -1 \\ 2 \\ 1 \\ 0 \\ 4 \end{bmatrix}.$$

We may also transform the weights $\alpha_1, \alpha_2, \ldots, \alpha_m$ into a diagonal weight matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ as

$$\mathbf{A} = \begin{bmatrix} \alpha_1 & 0 & \ldots & 0 \\ 0 & \alpha_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \alpha_m \end{bmatrix}.$$

For example, if $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) = (1, 2, 1, 3, 0.5)$, then

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}.$$

The WLS solution for $\mathbf{w} = \arg\min_{\mathbf{w}} \text{WMSE}(\mathbf{w})$ is known to be

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{A}\mathbf{X})^{-1}\mathbf{X}^\top \mathbf{A}\mathbf{y}. \tag{1}$$

(See the last page for a proof that you do not need to know for the purposes of this assignment.)

Write a Python script to find the WLS solution for $\mathbf{w}$ given an arbitrary matrix $\mathbf{X} \in \mathbb{R}^{5\times 2}$, a vector $\mathbf{y} \in \mathbb{R}^5$ and a diagonal matrix $\mathbf{A} \in \mathbb{R}^{5\times 5}$. Submit your Python code as a function ("`def A1_MatricNumber(X,A,y)`") that takes in $\mathbf{X}$, $\mathbf{A}$ and $\mathbf{y}$ as inputs and generates $(\mathbf{X}^\top \mathbf{A}\mathbf{X})^{-1}$ and $\mathbf{w}$ as outputs in a single file with the filename "`A1_StudentMatriculationNumber.py`". Your Python routine should return a matrix $(\mathbf{X}^\top \mathbf{A}\mathbf{X})^{-1}$ and the WLS solution vector $\mathbf{w}$ (as a numpy array). Hint: you will need "`import numpy as`" and its matrix manipulation functions.

Precise Instructions:

1. Please use the python template provided to you. Do not comment out any lines. Remember to rename both "`A1_StudentMatriculationNumber.py`" and "`A1_MatricNumber`" using your student matriculation number. For example, if your matriculation ID is A1234567R, then you should submit `A1_A1234567R.py` that contains the function `A1_A1234567R`.
2. Please do NOT zip/compress your file. Please do not redefine $\mathbf{X}, \mathbf{y}$ and $\mathbf{A}$ inside your function. The function will take in inputs $\mathbf{X} \in \mathbb{R}^{5\times 2}$, $\mathbf{y} \in \mathbb{R}^5$ and $\mathbf{A} \in \mathbb{R}^{5\times 5}$.
3. Please test your code at least once. Python is case sensitive.
4. Note that when we test your code, the matrices that are inputs to your function will be non-singular.
5. Because of the large class size, points will be deducted if instructions are not followed carefully. The way we would run your code might be something like this:

```
>> import A1_A1234567R as grading
>> InvXTX, w = grading.A1_A1234567R(X,A,y)
```

6. The allocation of the total mark (5%) is based on the two outputs: `InvXTAX` (2%) and `w` (3%).

Gentle Reminders:

1. Please make sure you replace "StudentMatriculationNumber" and "MatricNumber" with your matriculation number!
2. Date of release: Friday of Week 4
3. Submission: Canvas/EE2211/Assignments/Assignment 1
4. The submission folder in Canvas will be closed on 17th Feb 2023 (Friday of Week 6) at 23:59 sharp! No extensions will be entertained.

The following is **OPTIONAL** knowledge.

Proof of the WLS solution in (1). First we claim that (m times) the objective function can be written in matrix form as

$$\sum_{i=1}^{m} \alpha_i (\mathbf{x}_i^\top \mathbf{w} - y_i)^2 = (\mathbf{X}\mathbf{w} - \mathbf{y})^\top \mathbf{A} (\mathbf{X}\mathbf{w} - \mathbf{y}). \tag{2}$$

This is merely a matter of bookkeeping. Note that

$$\mathbf{X}\mathbf{w} - \mathbf{y} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_m^\top \end{bmatrix} \mathbf{w} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} - y_1 \\ \mathbf{x}_2^\top \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_m^\top \mathbf{w} - y_m \end{bmatrix}.$$

As such, the RHS of (2) is

$$\begin{aligned} (\mathbf{X}\mathbf{w} - \mathbf{y})^\top \mathbf{A} (\mathbf{X}\mathbf{w} - \mathbf{y}) &= \begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} - y_1 \\ \mathbf{x}_2^\top \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_m^\top \mathbf{w} - y_m \end{bmatrix}^\top \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha_m \end{bmatrix} \begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} - y_1 \\ \mathbf{x}_2^\top \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_m^\top \mathbf{w} - y_m \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} - y_1 \\ \mathbf{x}_2^\top \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_m^\top \mathbf{w} - y_m \end{bmatrix}^\top \begin{bmatrix} \alpha_1(\mathbf{x}_1^\top \mathbf{w} - y_1) \\ \alpha_2(\mathbf{x}_2^\top \mathbf{w} - y_2) \\ \vdots \\ \alpha_m(\mathbf{x}_m^\top \mathbf{w} - y_m) \end{bmatrix} = \sum_{i=1}^{m} \alpha_i (\mathbf{x}_i^\top \mathbf{w} - y_i)^2 = \text{LHS of (2)}. \end{aligned}$$

To minimize $f(\mathbf{w}) = (\mathbf{X}\mathbf{w} - \mathbf{y})^\top \mathbf{A} (\mathbf{X}\mathbf{w} - \mathbf{y})$, we note that it can be written as

$$f(\mathbf{w}) = \mathbf{w}^\top \mathbf{X}^\top \mathbf{A} \mathbf{X} \mathbf{w} - 2\mathbf{w}^\top \mathbf{X}^\top \mathbf{A} \mathbf{y} + \mathbf{y}^\top \mathbf{A} \mathbf{y}.$$

Differentiating this with respect to $\mathbf{w}$, we obtain

$$\nabla f(\mathbf{w}) = 2\mathbf{X}^\top \mathbf{A} \mathbf{X} \mathbf{w} - 2\mathbf{X}^\top \mathbf{A} \mathbf{y}.$$

Setting this to zero and assuming that $(\mathbf{X}^\top \mathbf{A} \mathbf{X})$ is invertible, we obtain that the optimal $\mathbf{w}$ is

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{A} \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{A} \mathbf{y}$$

as desired. □

Remark 1. *Under what conditions on* $\mathbf{X}$ *and* $\mathbf{A}$ *is* $\mathbf{X}^\top \mathbf{A} \mathbf{X}$ *invertible?*