DDA2020: Assignment I

February 19, 2022

Homework due: 11:59 pm, March 5, 2022. The exercise numbers refer to Kevin P. Murphy's book "Machine Learning: A Probabilistic Perspective".

1 Written Problems (6 points)

1. MLE minimizes KL divergence to the empirical distribution **(Exercise 2.15 of Murphy's book)** (1 point)

2. Centering and ridge regression **(Exercise 7.3 of Murphy's book)** (1 point)

3. Symmetric version of ℓ_2 regularized multinomial logistic regression **(Exercise 8.5 of Murphy's book)** (1 point)

4. Elementary properties of ℓ_2 regularized logistic regression **(Exercise 8.6 of Murphy's book)** (1 point)

5. Given the following denominator layout derivatives, (2 points)

   - **Differentiation of a scalar function *w.r.t.* a vector**: If $f(\mathbf{w})$ is a scalar function of d variables, $\mathbf{w}$ is a $d \times 1$ vector, then differentiation of $f(\mathbf{w})$ *w.r.t.* $\mathbf{w}$ results in a $d \times 1$ vector

   $$\frac{df(\mathbf{w})}{d\mathbf{w}} = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix}$$

   - **Differentiation of a vector function *w.r.t.* a vector**: If $\mathbf{f}(\mathbf{w})$ is a vector function of size $h \times 1$ and $\mathbf{w}$ is a $d \times 1$ vector, then differentiation of $\mathbf{f}(\mathbf{w})$ *w.r.t.* $\mathbf{w}$ results in a $d \times h$ vector

   $$\frac{d\mathbf{f}(\mathbf{w})}{d\mathbf{w}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_h}{\partial w_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial w_d} & \cdots & \frac{\partial f_h}{\partial w_d} \end{bmatrix}$$

Please prove the following derivatives, and $\mathbf{X}$ and $\mathbf{y}$ are not functions of $\mathbf{w}$:

$$\frac{d(\mathbf{X}^\top \mathbf{w})}{d\mathbf{w}} = \mathbf{X},$$
$$\frac{d(\mathbf{y}^\top \mathbf{X}\mathbf{w})}{d\mathbf{w}} = \mathbf{X}^\top \mathbf{y}$$
$$\frac{d(\mathbf{w}^\top \mathbf{X}\mathbf{w})}{d\mathbf{w}} = (\mathbf{X} + \mathbf{X}^\top)\mathbf{w}$$

2 Programming (5 points)

Given $\mathbf{X}\mathbf{w} = \mathbf{y}$ where $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 4 & 3 \\ 5 & 6 \\ 3 & 8 \\ 9 & 10 \end{bmatrix}$ and $y = \begin{bmatrix} -1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ which constitute an exemplary problem. Write a Python routine to find the least squares solution for $\mathbf{w}$ given arbitrary $\mathbf{X} \in \mathcal{R}^{5\times 2}$ and $\mathbf{y} \in \mathcal{R}^{5\times 1}$. Submit your Python codes as a function routine ("def A1_MatricNumber(X,y)") that takes in $\mathbf{X}$ and $\mathbf{y}$ as inputs and generate $(\mathbf{X}^\top \mathbf{X})^{-1}$ and $\mathbf{w}$ as outputs in a single file with filename "A1_StudentMatriculationNumber.py". Your Python routine should return the least squares solution vector $\mathbf{w}$ (as numpy array) and two matrices $\mathbf{X}^\top$ and $(\mathbf{X}^\top \mathbf{X})^{-1}$. **Hint**: you will need "import numpy as np" and its matrix manipulation functions.

Please use the python template provided to you. Remember to rename both "A1_StudentMatriculationNumber.py" and "A1_MatricNumber" using your student matriculation number. For example, if your matriculation ID is 123456789, then you should submit "A1_123456789.py" that contains the function "A1_123456789". The way we would run your code might be something like this:
≫ import A1_123456789 as grading
≫ w, XT, InvXTX = grading.A1_123456789(X,y)
Marks allocation is based on the three outputs: w (3 points), XT (1 point), InvXTX (1 point)

NOTE: Please do **NOT** zip/compress your file. Please make sure you replace "StudentMatriculationNumber" and "MatricNumber" with your matriculation number! Because of the large class size, **points will be deducted if instructions are not followed**.