Supervised Machine Learning (DS 5220)
Homework 1

Instructor: Ehsan Elhamifar

Due Date: October 1, 2020, 11:59pm

1) Show that we can always write an arbitrary matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ as the sum of a symmetric and an anti-symmetric matrix, i.e., $\boldsymbol{A} = \boldsymbol{B} + \boldsymbol{C}$, where $\boldsymbol{B}$ is symmetric and $\boldsymbol{C}$ is anti-symmetric. Obtain the $\boldsymbol{B}$ and $\boldsymbol{C}$ in your answer.

2) For a matrix $\boldsymbol{A} \in \mathbb{R}^{m \times n}$, we define the Frobenius norm as $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}$, where a_{ij} indicates the (i, j)-th entry of $\boldsymbol{A}$. Show that $\|\boldsymbol{A}\|_F^2 = \mathrm{trace}(\boldsymbol{A}^\top \boldsymbol{A})$.

3) Assume $\boldsymbol{U} \in \mathbb{R}^{n \times n}$ is an orthonormal matrix, i.e., $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{I}_n$. Show that for an arbitrary vector $\boldsymbol{x} \in \mathbb{R}^n$, we have $\|\boldsymbol{U}\boldsymbol{x}\|_2^2 = \|\boldsymbol{x}\|_2^2$. In other words, transformation of a vector by an orthonormal matrix preserves the norm (magnitude) of the vector.

4) For vectors $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{a} \in \mathbb{R}^n$ and matrices $\boldsymbol{X} \in \mathbb{R}^{n \times n}$, $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, show the following.

1. $\frac{\partial \boldsymbol{a}^\top \boldsymbol{x}}{\partial \boldsymbol{x}} = \boldsymbol{a}$.
2. $\frac{\partial \boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x}}{\partial \boldsymbol{x}} = (\boldsymbol{A} + \boldsymbol{A}^\top)\boldsymbol{x}$.
3. $\frac{\partial\, \mathrm{trace}\,(\boldsymbol{A}^\top \boldsymbol{X})}{\partial \boldsymbol{X}} = \boldsymbol{A}$.
4. $\frac{\partial \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2}{\partial \boldsymbol{x}} = 2\boldsymbol{A}^\top(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$.

5) Determine whether each of the following functions is convex or not.

1. $f(x) = e^{ax}$, for a fixed $a \in \mathbb{R}$.
2. $f(x) = -\log(x)$, with the domain $x \in (0, +\infty)$.
3. $f(x) = e^{g(x)}$, where $g(x)$ is convex.

6) Consider the training dataset $\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_N, y_N)\}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$. Consider the modified regression problem

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{i=1}^{N} s_i (y_i - \boldsymbol{\theta}^\top \boldsymbol{x}_i)^2,$$

where each $s_i \in [0, 1]$ is given and indicates the importance score of $(\boldsymbol{x}_i, y_i)$ for regression.

1. Write down the closed-form solution of the above regression problem. Provide all steps of your derivations.

2. Write down the steps of the gradient-descent algorithm to solve for for the above cost.

7) Consider the function $f(x_1, x_2) = x_1^2 + (x_2 - 2)^2$. Write down the expression of the gradient of f.

1. Plot the function in 3D.
2. Write a python code that gets as input i) an initial point (x_1^0, x_2^0), ii) the maximum number of iterations of the gradient descent, iii) learning rate ρ; and runs the gradient descent algorithm starting from the initial point until convergence or until the maximum number of iterations is achieved. The output of the code must be the sequence of points (starting from initial point and ending with the last point) obtained by the gradient descent.
3. Use the learning rate $\rho = 0.01$ and the initial point $(x_1, x_2) = (1, 1)$. Plot the sequence of obtained points. After how many iterations does GD converge?
4. Use the learning rate $\rho = 0.1$ and the initial point $(x_1, x_2) = (1, 1)$. Plot the sequence of obtained points. After how many iterations does GD converge?
5. Use the learning rate $\rho = 5$ and the initial point $(x_1, x_2) = (1, 1)$. Plot the sequence of obtained points. After how many iterations does GD converge?

8) Consider the training dataset

$$\{(0.10, 0.65), (0.50, 0.10), (0.90, 0.35), (-0.20, 0.17), (-0.5, 0.42), (1.50, 2.62)\},$$

where in $(\cdot, \cdot)$, the first entry is the input variable, x, and the second entry is the output variable (response), y. Consider the regression model $y = \theta_1 x + \theta_0$.

1. Write a python code that inputs the above data and outputs the optimal regression value (θ_1^*, θ_0^*), using the closed-form solution.
2. Plot the data in 2D and plot the estimated line $y = \theta_1^* x + \theta_0^*$.
3. Remove the last point, i.e., (1.50,2.62), from the data and repeat the previous part. Is the new estimated line close to the estimated line in the previous part? Provide a justification.

Homework Submission Instructions: Please submit both the analytical part and the programming part of the homework via email by the DEADLINE according to the following format. To submit, please send an email to the instructor and cc the TA.
– The title of your email must be “DS5520: HW01:Your-Last-Name”.
– Please attach a single zip file to your email that contains 1) the PDF of your analytical solution (you can write on papers and then scan them into a single pdf or you can type in latex/word and convert to pdf), 2) a folder of all python codes (.py file or jupyter notebook are both acceptable), generated plots and a readme file on how to run your files.
– Please name your zip file as “HW01:Your-Last-Name”.