Homework 1

Deadline: Sunday, Feb 5th, at 11:59pm.

Submission: You need to submit a `APL405_HW1_<RollNumber>_<FirstName>.zip` file on Microsoft Teams

- Put all your solutions and answers to all questions as a PDF file titled `hw1_writeup.pdf`. You can produce the file however you like (e.g. LATEX, Microsoft Word, scanner), as long as it is readable.
- Your code for Question 2 should be submitted as a Python file `hw1_code.py`.

Late Submission: 50% of the marks will be deducted for any submission beyond the deadline. No submissions will be accepted after two days past the deadline.

Collaboration: Homeworks are individual work. Please refrain from copying.

1. **[4 marks] Curse of dimensionality in k-NN**: It was told in the lecture that k-NN does not work very well for high-dimensional input space, as most points in such a space are quite far away from each other and therefore every points is approximately similar distance apart from each other.

   (a) [**1 mark**] You will first verify using simulation. Consider different sizes of dimensions $d \in [2^0, 2^1, 2^2, \ldots, 2^{10}]$. For each choice of dimension, sample 100 points from the unit (hyper) cube and record the average squared Euclidean distance between all pairs of points, as well as the standard deviation of the squared Euclidean distances. Plot both the average and the standard deviation as a function of d. (Use `np.mean` and `np.std` to compute the statistics, and `matplotlib` for plotting. You may find `numpy.random.rand` helpful in sampling from the unit cube). Include the output figure in your solution PDF.

   (b) [**1.5 mark**] Furthermore, we want to verify our simulations in (a) analytically by calculating the averaged distance and the variance of distance. First, consider two independent univariate random variables X and Y sampled uniformly from the unit interval $[0, 1]$. Determine the expectation and variance of the random variable Z, defined as the squared distance $Z = (X - Y)^2$.

   (c) [**1 mark**] Now consider that we sample two points independently from a unit cube in d dimensional space. Observe that each coordinate is sampled independently from $[0, 1]$, i.e. view this as sampling random variables $X_1, \ldots, X_d$, $Y_1, \ldots, Y_d$ independently from $[0, 1]$. The squared Euclidean distance can be written as $S = Z_1 + \ldots + Z_d$, where $Z_i = (X_i - Y_i)^2$. Using the properties of expectation and variance, determine $\mathbb{E}[S]$ and $\texttt{Var}[S]$. You may give your answer in terms of the dimension d, and $\mathbb{E}[S]$ and $\texttt{Var}[S]$ obtained from part (b).

   (d) [**0.5 mark**] Using Markov's inequality, one can derive for any random variable Z that $p(|Z - \mathbb{E}[Z]| \geq a) \leq \dfrac{\texttt{Var}[Z]}{a^2}$. Based on part (c), explain why does this support the claim that in high dimensions, *most points in high-dimensional space are far away from each other and therefore they are approximately similar distance far apart from each other*?

2. **[2 marks]** **k-NN with handwritten digits**: As was told in one of the lectures, the MNIST dataset has around 70000 grayscale (meaning black and white) images of handwritten digits. These images are represented using matrices with any component of the matrix having values between 0 and 255. Use the `mnistData` dataset (provided in the zip file)and write an `mnist1NNdemo.py` python script (or a `.ipynb` notebook) with proper comments.

   - Use first 60000 images for training (from training dataset) and first 10000 images for testing (from testing dataset).
   - Use $k = 1$ and $k = 3$.

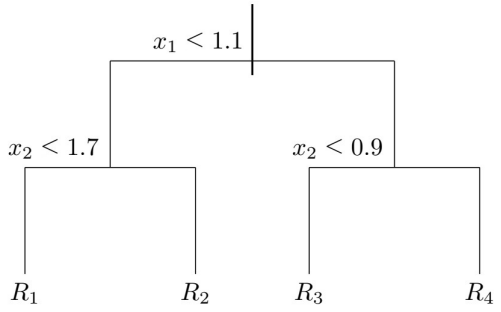
   (a) **[1.5 marks]** Report that the misclassification rate on the test set of MNIST using 1-NN classifier and 3-NN classifier, respectively.

   (b) **[0.5 mark]** Modify the code so that you first randomly permute the features (columns of the training and test design matrices), and then apply the 1-NN classifier. Apply the same permutations in the test dataset and training dataset. How does the misclassification rate change?

3. **[4 marks]** **Regression Trees** Consider a regression problem with a two input variables $x = [x_1\ x_2]^T$, where $x_1 \in [0\ \ 3]$ and $x_2 \in [0\ \ 3]$, and one output $y \in \mathbb{R}$. Based on the following training data

i	x_1	x_2	y
1	1.4	1.4	0.1
2	2.2	2.2	0.6
3	0.2	0.8	-0.7
4	1.0	2.8	-1.8
5	0.6	0.2	0.3
6	0.4	2.2	-1.9
7	0.6	1.4	-0.9
8	1.2	1.8	-0.5
9	1.2	0.6	0.7
10	1.8	0.6	1.5

the regression tree shown below has been constructed using recursive binary splitting.



You don't need to write any code for this problem. Use drawings and show calculations in a tabular fashion.

(a) **[1 mark]** Draw the partitions of the regression tree on the input space $[0\ \ 3] \times [0\ \ 3]$. Mark the regions with the names of the leaf nodes $R_1, \ldots, R_4$. Also show the observations lying in each region.

(b) **[0.5 mark]** Use the regression tree to predict the output of the test input $\mathbf{x}^* = [1.5\ \ 1.8]^T$

(c) **[2 marks]** Continue to grow the tree by adding splits such that there are **at most** two data points in each region. Choose your splits in such a way as to minimize the *squared error loss*.

(d) **[0.5 mark]** What is the predicted output of the test input $\mathbf{x}^*$ from (b) using this new deeper tree?

4. **[1.5 marks] Derivation of Normal Equations**: It was mentioned in class that linear regression using maximum likelihood approach (assuming Gaussian distribution on i.i.d. noise ϵ), we can get a log-likelihood which we maximize to obtain the *maximum likelihood estimate* of $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \underbrace{\ln p\left(\mathbf{y} \mid \mathbf{X}; \boldsymbol{\theta}\right)}_{\text{log-likelihood}}$$

It was also mentioned that the estimate of the parameter vector admits a closed form solution of the form

$$\left(\mathbf{X}^T\mathbf{X}\right)\hat{\boldsymbol{\theta}} = \mathbf{X}^T\mathbf{y}$$

(a) **[1 mark]** Use a calculus-based approach (by taking derivative of the log likelihood w.r.t. $\boldsymbol{\theta}$ and setting it to zero) to derive the closed-form solution by hand.

(b) **[0.5 mark]** What does it mean in practice if $\left(\mathbf{X}^T\mathbf{X}\right)$ is not invertible?

5. **[3.5 marks] Derivation of Logistic Regression Model**: In class, we derived the logistic regression model for binary classification by considering the two output class labels as $\{-1,\ \ 1\}$. In this problem, consider the class labels as $\{0,\ \ 1\}$.

(a) **[0.5 mark]** Show that the derivative of the logistic function $h(x)$ is given by

$$\frac{dh(x)}{dx} = h(x)\left(1 - h(x)\right)$$

(b) **[1 mark]** Derive the negative log-likelihood function associated with the two-class $\{0,\ \ 1\}$ logistic regression.

(c) **[1 mark]** Derive the gradient of the negative log-likelihood function associated with the two-class $\{0,\ \ 1\}$ logistic regression.

(d) **[1 mark]** Derive the Hessian matrix of the negative log-likelihood function associated with the two-class $\{0,\ \ 1\}$ logistic regression.