CSCI 5521: Machine Learning Fundamentals (Fall 2022)[1]

Homework 1 (Tuesday, Sep. 20) Version 2

Due on Gradescope at 11:59 PM, Friday, Sep. 30

Instructions:

- This homework has 7 questions, 100 points, on 4 pages.
- Please provide the detailed calculation or derivation, or explanation for each question to avoid point reduction.

1. (**12 points**) Calculate the **expectation** $E(X)$ and the **variance** $Var(X)$ of the following corresponding continuous or discrete variables X,

   (a) (**4 points**) X is a continuous variable whose Probability Density Function (PDF) is:

   $$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0 \\ 0 & x < 0 \end{cases}, \quad \lambda > 0 \tag{1}$$

   (b) (**4 points**) X is a continuous variable whose Probability Density Function (PDF) is:

   $$f(x;\theta) = \begin{cases} \frac{1}{\theta} & 0 \leq x \leq \theta \\ 0 & otherwise \end{cases}, \quad \theta > 0 \tag{2}$$

   (c) (**4 points**) X is a discrete variable whose Probability Mass Function (PMF) is:

X	0	1	2	3	4
$P(x)$	0.1	0.2	0.4	0.2	0.1

   (3)

2. (**18 points**) Suppose that we have 2 classes, C_1 and C_2, with the conditional probability density functions as listed below. Assume that $0 \leq x \leq 10$. Answer the following questions.

   $$P(x|C_1) = \frac{3}{1000}x^2 \tag{4}$$

   $$P(x|C_2) = -\frac{1}{50}(x-10) \tag{5}$$

   (a) (**4 points**) Given $x = 5$, express $P(C_1|x)$ and $P(C_2|x)$ in terms of $P(C_1)$ and $P(C_2)$ *only*.

   (b) (**6 points**) Assuming $x = 5$, find the value of $P(C_1)$ that the posterior probabilities of the point $x = 5$ from C_1 and C_2 are the same.

   (c) (**8 points**) Assume that C_1 and C_2 have equal priors, compute the range of x such that x will always be classified as C_1. Note: make sure you take into account the range of x as shown in the question. What about if $P(C_1) = 0.4$ and $P(C_2) = 0.6$?

[1]Instructor: Kshitij Tayal (tayal@umn.edu). TA: Xianyu Chen, and Yoshitaka Inoue (csci5521@umn.edu).

3. (**14 points**) After your yearly checkup, the doctor has good news and bad news. The bad news is that you tested positive for a rare disease and that the test is very accurate: the probability of testing positive when you do have the disease is 0.983, and the probability of testing negative when you don't have the disease is 0.945. The good news is that this is a rare disease, striking only one in ten thousand people in your demographic.

   (a) (**4 points**) What are the chances you have the disease considering on the doctor provided above good news and bad news?

   (b) (**5 points**) Now assign a cost to the errors: deciding to seek treatment for the rare disease when in fact you are healthy will cost you \$1000 in unnecessary tests and the recovery therefrom. Deciding to forgo treatment when in fact you have the rare disease will cost you and your family \$1,000,000 in loss of life/income etc. Assume a correct decision (seek treatment if you have rare disease, forgo treatment if you are healthy) has no cost, for simplicity. What is the expected cost (i.e., "risk") assuming the test comes out positive and you undergo treatment?

   (c) (**5 points**) What should your decision (forgo treatment or seek treatment) be after a positive test? (Is this different from the answer to part (a)?)

4. (**10 points**) Consider the linear mapping:

$$\Phi : \mathbb{R}^3 \to \mathbb{R}^4$$

$$\Phi\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) = A_\Phi \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3x_1 + 2x_2 + x_3 \\ x_1 + x_2 + x_3 \\ x_1 - 3x_2 \\ 2x_1 + 3x_2 + x_3 \end{bmatrix} \tag{6}$$

   (a) (**4 points**) Find the transformation matrix A_Φ.

   (b) (**4 points**) Could you say whether $|A_\Phi A_\Phi^T| = 0$, without any computations? Clearly explain your answer.

   (c) (**2 points**) Could you say what Rank(A_Φ) is, without any computations? Clearly explain your answer.

5. (**16 points**) Linear regression learns a linear function of feature variables X to fit the responses y. In this problem, you will derive the closed-form solution for the two variations of the linear regression.

   (a) (**8 points**) The standard linear regression can be formulated as solving a least square problem

$$\underset{w}{\text{minimize}} \quad ||Xw - y||^2$$

   Where $X \in \mathbb{R}^{m \times n}$ represents the feature matrix, $y \in \mathbb{R}^{m \times 1}$ represents the response vector. $w \in \mathbb{R}^{n \times 1}$ is the variable in the least square problem. This is a convex objective function of w. Derive the optimal w by setting the derivative of the function wrt w to zero to minimize the objective function.

(b) (**8 points**) In practice, a L2-norm regularizer is often introduced into linear regression, called Ridge Regression, to overcome ill-posed problems, where the hessian matrix is not positive definite. The objective function of ridge regression is defined as

$$\underset{w}{\text{minimize}} \quad ||Xw - y||^2 + \lambda ||w||^2,$$

where $\lambda > 0$. This objective function is strictly convex. Derive the solution of the ridge regression problem to find the optimal w.

Programming assignments: The next two problems involve programming.

6. (**15 points**) Here, we plan to review the model underfitting and overfitting problems. The more representative model with less amount of data points would cause the overfitting problem, while the less representative model would lead to the underfitting problem. In this problem, we would like to explore these cases in the regression paradigm. We use our generated data, where (x^t, y^t, s^t) represents the t-th data sample. More specifically, x^t is the attribute value, y^t is the class label, and s^t represents the fold for the t-th sample.

   (a) (**8 points**) Implement the linear regression algorithm (`MyRegression.py`) and use the 10-fold cross validation to evaluate different order of the polynomial regression with orders $n = 2, 3, 4, 5, 6$. We define $\mathbf{X} \in \mathbb{R}^{N \times 1}$ as the feature matrix of N samples in 1 dimension, $\mathbf{y} \in \mathbb{R}^{N \times 1}$ as the continuous annotated label vector, and $\mathbf{s} \in \mathbb{R}^{N \times 1}$ as the fold $(0, 1, \cdots, 9)$. Now, you can write a linear regression algorithm with orders $n = 2, 3, 4, 5, 6$ polynomial regression and the 10-fold cross-validation and report the mean square error (MSE) of each fold for the different orders of polynomial regression separately.

   (b) (**7 points**) Draw an error figure, where the x-axis is the order of polynomial regression and the y-axis is the mean of the MSE of the 10 folds. And briefly describe what your observation is. **The visualization code is presented in `problem6.py`, and you can use it to verify your implementation of the polynomial regression algorithm. To submit your assignment, you need to include the generated plot. Note that you do not need to modify this file**.

7. (**15 points**) In this problem, we introduce a simple perceptron algorithm for binary classification. At first, we provide the following pseudo-code of perceptron algorithm. We define the t-th data sample as $(\mathbf{x}^t, y^t)$, where $\mathbf{x}^t$ is the attribute value and $y^t = \pm 1$ is the human annotated label for the t-th sample:

   1. Initialize $\mathbf{w} = \mathbf{w}_0$
   2. **Do** Iterate until $\mathbf{w}$ converge
   3.   **For** each sample for the data $(\mathbf{x}^t, y^t), t = 1, 2, \cdots$
   4.   **If** $y^t \langle \mathbf{w}, \mathbf{x}^t \rangle \leq 0$
   5.   Update $\mathbf{w} = \mathbf{w} + y^t \mathbf{x}^t$

   In this problem, we define the "convergence" as that $\mathbf{w}$ will not change if the algorithm goes through all the training dataset as mentioned in Line 3 in the algorithm. For notation: we use $\mathbf{x}^t$ to denote the t-th sample in the training data, located in the t-th row of the matrix $\mathbf{X}$. We use an upper case T to represent the transpose of a vector or matrix $\mathbf{X}$ as $\mathbf{X}^T$.

(a) (**8 points**) Implement the corresponding perceptron algorithm (`MyPerceptron.py`) and then test your algorithm in the data provided. There is a total of N samples in our training dataset. For features and labels, we denote them as $\mathbf{X} \in \mathbb{R}^{N\times 2}$ and $\mathbf{y} \in \mathbb{R}^{N\times 1}$. We use the initial value $\mathbf{w}_0 = [0.1; -1]^T$. Run your perceptron algorithm with the provided data and answer the following questions. How many iterations does it take to converge? What is the error rate of the resulting fit? In other words, it means the portion of points is misclassified by your implemented perceptron classifier.

(b) (**7 points**) Visualize all the samples. In this case, you can use 2 different colors/shapes for these 2 different classes and then plot the decision boundary for the initial weights $\mathbf{w}_0$ and your obtained weights $\mathbf{w}$ that is returned by the perceptron program. **The code is presented in `problem7.py`, and you can use it to verify your implementation of the perceptron algorithm. To submit your assignment, you need to include the generated plots "initial_result_w0.png" and "perceptron_result.png". Note that you do not need to modify this file**.

Submission

- Additional Instructions: You need to submit the code written in Python instead of any other programming language. Other programming languages are not accepted and would cause credit loss. We provide the template including the corresponding inputs and the outputs of the function, you need to fill the function and generate the expected results.
- Things to submit:
  - `hw1_sol.pdf`: the document including all of your answers to the problem including the summary or figure of the programming problems. The PDF format is accepted. If you would like to use the hand-written document, you can scan it and make sure that the scanned copy is clearly readable. If it is difficult to read, credit would be lost.
  - `MyRegression.py`: a text file containing the python function for Problem 6. Use the skeleton file `MyRegression.py` and then fill in the missing parts.
  - `MyPerceptron.py`: a Python source file for Problem 7. Use this skeleton file `MyPerceptron.py` found with the data on the class website, and fill in the missing parts.
- Submit: You need to submit the materials electronically to the Gradescope. **For This assignment, you need to submit the materials [Hw1-Written (for `hw1_sol.pdf`) and the HW1 programming (the compressed file such zip file including the Python codes)].** We will grade your code in vanilla Python.