CSCI 5521: Introduction to Machine Learning (Fall 2021)[1]

Homework 0

Due date: September 14, 2021 11:59pm

This homework is intended for you to test your preparation for this class. We will record your answers and mark that you have handed this in, but it will not be fully graded. It will also test that you have figured out the software environment needed for this class:

- you have installed Python, Numpy, Matplotlib, and can write simple programs;
- you can access Canvas, and submit assignments through Gradescope.

1. Linear regression learns a linear function of feature variables $\mathbf{X}$ to fit the responses y. In this problem, you will derive the closed-form solution for linear regression formulations.

   (a) (**20 points**) The standard linear regression can be formulated as solving a least square problem

   $$\underset{\mathbf{w}}{\text{minimize}} \quad \phi(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 = \langle \mathbf{X}\mathbf{w} - \mathbf{y}, \mathbf{X}\mathbf{w} - \mathbf{y} \rangle$$

   where $\mathbf{X} \in \mathbb{R}^{n \times m}$ $(n \geq m)$ represents the feature matrix, $\mathbf{y} \in \mathbb{R}^{n \times 1}$ represents the response vector and $\mathbf{w} \in \mathbb{R}^{m \times 1}$ is the vector variable of the linear coefficients. Here the $i-j$-th element of $\mathbf{X}$, denoted x_{ij}, is the j-th attribute value for the i-th data sample (observation) and y_i is the true response for the i-th data sample. This is a convex objective function of $\mathbf{w}$. Derive the conditions for the optimal $\mathbf{w}$ by setting the gradient of the function wrt $\mathbf{w}$ to zero to minimize the objective function. The conditions will be in the form a square system of linear equations that $\mathbf{w}$ must satisfy. To find the gradient, you can use the following formula

   $$\begin{aligned} \phi(\mathbf{w} + \boldsymbol{\delta}) &= [\mathbf{X}(\mathbf{w} + \boldsymbol{\delta})]^T \mathbf{X}(\mathbf{w} + \boldsymbol{\delta}) - 2 [\mathbf{X}(\mathbf{w} + \boldsymbol{\delta})]^T \mathbf{y} + \mathbf{y}^T \mathbf{y} \\ &= \phi(\mathbf{w}) + 2 [\mathbf{X}\boldsymbol{\delta}]^T [\mathbf{X}\mathbf{w} - \mathbf{y}] + (\mathbf{X}\boldsymbol{\delta})^T \mathbf{X}\boldsymbol{\delta}, \end{aligned}$$

   and note that $\mathbf{w}$ must be determined so that $\phi(\mathbf{w} + \boldsymbol{\delta}) \geq \phi(\mathbf{w})$ for any possible vector $\boldsymbol{\delta}$ (why?). Here $\mathbf{X}^T$ denotes the transpose of $\mathbf{X}$, and $[\mathbf{X}\boldsymbol{\delta}]^T = \boldsymbol{\delta}^T \mathbf{X}^T$.

   (b) (**10 points**) In practice, a L2-norm regularizer is often introduced with the least squares, 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

   $$\underset{\mathbf{w}}{\text{minimize}} \qquad \tilde{\phi}(\mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w}\|^2 = \left\| \begin{pmatrix} \mathbf{X} \\ \sqrt{\lambda} I \end{pmatrix} \mathbf{w} - \begin{pmatrix} \mathbf{y} \\ \mathbf{0} \end{pmatrix} \right\|^2$$

[1] Instructor: Catherine Qi Zhao. TA: Shi Chen, Xianyu Chen, Helena Shield, Jinhui Yang, Yifeng Zhang. Email: csci5521.f2021@gmail.com

where $\lambda > 0$ and I is an $m \times m$ identity matrix. This objective function is strictly convex. Derive the solution of the ridge regression problem to find the optimal $\mathbf{w}$.

2. (a) (**15 points**) Let D be a random variable for a given disease, assume that the probability a person has the disease is 0.1. Based on this information, researchers developed a new method to say if a person has the disease: for each 10 people that do the test, they randomly report that 1 of them has the disease. Will the method correctly identify if the person has the disease? Briefly explain your answer.

   (b) (**15 points**) Another group of researchers developed a new blood test to identify the same disease. The test result is given by a random variable X, with sensitivity and specificity given by 0.7 and 0.8, respectively (that means $p(X = 1|D = 1) = 0.7$ and $p(X = 0|D = 0) = 0.8$). If a patient did the blood test and the result is positive, what is the probability that the person has the disease?
   Hint: you might want to use the Bayes Rule: $p(b|a) = \frac{p(a|b)p(b)}{p(a)}$

3. (**30 points**) Below is the pseudo-code of perceptron algorithm for binary classification, where $(\mathbf{x}^t, y^t)$ is the t-th data sample: $\mathbf{x}^t$ is the vector of attribute values (real numbers) and $y^t = \pm 1$ is the class label for the t-th sample:

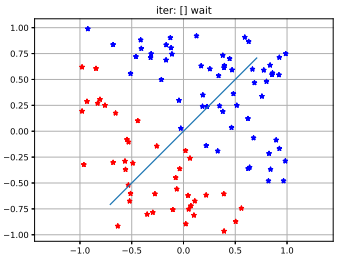
```
1. w = w_0.
2. Do Iterate until convergence
3.     For each sample (x^t, y^t), t = 1, 2, ...
4.         If y^t <w, x^t> <= 0
5.             w = w + y^t x^t
```

   Here "convergence" means $\mathbf{w}$ does not change at all over one pass through the entire training dataset in the loop starting in step 3. A note on notation: $\mathbf{x}^t$ denotes the t-th sample in the training data, which is found in the t-th row of the matrix $\mathbf{X}$. This is the notation used in the textbook. The transpose of a vector or matrix M is denoted M^T with an upper case T.

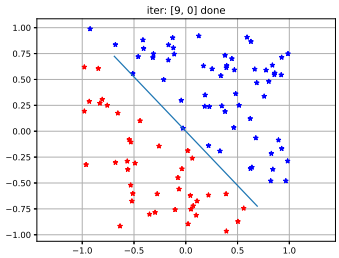
   (a) Implement the perceptron algorithm (`MyPerceptron.py`) and test it on the data provided on the class web site. $\mathbf{X} \in \mathbb{R}^{N \times 2}$ is the feature matrix of N samples in 2 dimensions and $\mathbf{y} \in \mathbb{R}^{N \times 1}$ is the label vector (± 1). Use initial value $\mathbf{w}_0 = [1; -1]^T$. Now, run your perceptron algorithm on the given data. How many iterations does it take to converge? What is the error rate of the resulting fit (i.e., how many points are misclassified by this classifier)? The prediction on a single data point $\mathbf{x}^t$ can be computed by comparing $\langle \mathbf{w}, \mathbf{x}^t \rangle$ with the threshold 0 (*e.g.*, $\langle \mathbf{w}, \mathbf{x}^t \rangle >= 0$ then the predicted label is $+1$).

   (b) Visualize all the samples (use 2 different colors for the 2 different classes), and plot the decision boundary defined by the initial $\mathbf{w}_0$ (before training) and $\mathbf{w}$ returned by the perceptron program (after training). **The code for plotting is included in `hw0.py`, you can use it to verify your implementation**

of the perceptron algorithm. It will save the plots under the name "initial.png" and "perceptron.png". Note that you do not need to modify the file.



before iteration



after convergence

4. (**10 points**) Let $\mathbf{X}$ be the matrix from Problem 3, and $\mathbf{y} \in \{-1, 1\}^N$ be the corresponding labels from Problem 3. Solve the least squares problem (`MyLeastSquare.py`) as shown in Problem 1 for the weights $\mathbf{w} \in \mathbb{R}^2$ and report the error rate for the resulting fit. Visualize the decision boundary defined by $\mathbf{w}$ (the plot will be saved under the name "least_square.png").

Submission

- **Things to submit:**
  1. hw0_sol.pdf: a document containing all your answers of Problem 1-2 and the three plots asked by Problem 3-4.
  2. `MyPerceptron.py`: a text file containing the python function for Problem 3 with header `def MyPerceptron(X, y, w0)`, where X is the feature matrix, `y` is a label vector (± 1) and `w0` is the initial value for the parameter vector $\mathbf{w}$. Use the skeleton file `MyPerceptron.py` found with the data on the class web site, and fill in the missing parts. The output of the function should be a tuple consisting of the final weight vector $\mathbf{w}$, the number of iterations k, and the fraction of training samples that are misclassified. Include the number of iterations, error rate, and the plots of decision boundary on your PDF submission.
  3. `MyLeastSquare.py` a text file containing the python function for Problem 4 with header `def MyLeastSquare(X, y)`, where X is the feature matrix, `y` is a label vector (± 1). Use the skeleton file `MyLeastSquare.py` found with the data on the class web site, and fill in the missing parts. The output of the function should be a tuple consisting of the final weight vector $\mathbf{w}$ and the fraction of training samples that are misclassified. Include the error rate, and the plot of decision boundary on your PDF submission.
- **Submit**: All material must be submitted electronically via Gradescope. **Note that There are two entries for the assignment, *i.e.*, Hw0-Written (for hw0_sol.pdf)**

and Hw0-Programming (for a zipped file containing the Python code), please submit your files accordingly. We will grade the assignment with vanilla Python, and code submitted as iPython notebooks will not be graded. This homework will not be graded but required as a proof of satisfying the prerequisites for taking the class.