Homework Assignment 4

COGS 118A: Supervised Machine Learning Algorithms

Due: January 29th, 2020, 11:59pm (Pacific Time).

Instructions: There are two parts to this homework. Part 1 is distributed here in this PDF. Part 2 is the code skeleton for questions 5 & 6, and it is distributed in a zip file that can be downloaded from Canvas at `Modules -> Week 4 -> HW4`. Open the two Jupyter notebooks from the zip file and complete the missing code.

Part 1 : Your math can be **handwritten**. In that case use a scanner app to take photos of the pages and turn them into a PDF. Adobe, Evernote, Microscoft, and many others offer free scanner apps. Alternatively, your math may be done in LaTeX or in a Jupyter Notebook using LaTeX markdown and from there turned into PDF.

Part 2: Output from Jupyter Notebooks (math markdown, code, or plots) can be turned into PDF using `File->Print->Preview` or `File->Download as->PDF via LaTeX`

In the end: all of these different sources must be stitched together into **a single PDF file**. PDF Merge tool (e.g. smallpdf, comebinepdf) can do this among many other tools. Make sure to show the steps of your solution, not just the final answer. You may search information online but you will need to write code/find solutions to answer the questions yourself.

Late Policy: Late assignments are deducted 5% each day they are late. No late assignments are accepted after one week.

System Setup: For this class, please use **Python 3.6** or later for homework with recent copies of the libraries NumPy, SciPy, Pandas, Scikit-learn, Matplotlib, Seaborn, and Jupyter-Notebook.

To run a Jupyter Notebook you may use UCSD Datahub or Google Colab or your own local installation. If you decide on local installation, and you are creating your own setup for the first time we highly recommend you use Anaconda as it will come with all the required libraries and more.

If you are not feeling comfortable with the programming assignments in this homework, it might help to take a look at `https://github.com/UCSD-COGS108/Tutorials`

Grade: ____ out of 100 points

1 (15 points) Conceptual Questions

Select the correct option(s). Note that there might be multiple correct options.

1. (3 points) For two monotonically increasing functions $f(x)$ and $g(x)$:

   A. $f(x) + g(x)$ is always monotonically increasing.
   B. $f(x) - g(x)$ is always monotonically increasing.
   C. $f(x^2)$ is always monotonically increasing.
   D. $f(x^3)$ is always monotonically increasing.

2. (3 points) Which one of the following statements is true about solving the regression problem?

   A. Matrix inverse could be expensive for computing the closed form solution in least square estimation when the dataset consists of samples in high dimension spaces.
   B. The learning rate in the gradient descent algorithm should be set as large as possible to speed up the convergence.
   C. There exists a fixed small learning rate that works for all the gradient descent algorithms.

3. (3 points) For a function $f(x) = x(10 - x), x \in \mathbb{R}$, please choose the correct statement(s) below:

   A. $\arg\max_x f(x) = 5$.
   B. $\arg\min_x f(x) = 25$.
   C. $\min_x f(x) = 5$.
   D. $\max_x f(x) = 25$.

4. (3 points) Assume we have a function $f(x)$ which is differentiable at every $x \in \mathbb{R}$. There are three properties that describe the function $f(x)$:
   (1) $f(x)$ is a convex function.
   (2) When $x = x_0$, $f'(x_0) = 0$.
   (3) $f(x_0)$ is a global minimum of $f(x)$.

   Which one of the following statements is **wrong**?
   Hint: You can use a failure case to disprove a statement.

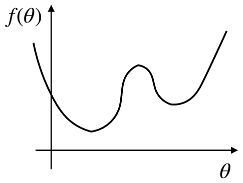
   A. Given (1) and (2), we can prove that (3) holds.
   B. Given (2) and (3), we can prove that (1) holds.
   C. Given (1) and (3), we can prove that (2) holds.

5. (3 points) Suppose in the final project of COGS 118A, you're doing polynomial regression. You try two separate loss functions – L_1-norm and L_2-norm – for this task. When you fit the regression, the training set loss after convergence for the L_1-norm is 32, and for the L_2-norm it's 73 on the same dataset. Since training set loss for the L_1-norm is lower than for the L_2-norm, you conclude that the model using L_1-norm is better. Is this conclusion correct?

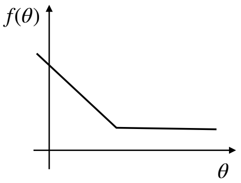
   [True] [False]

2 (12 points) Convex

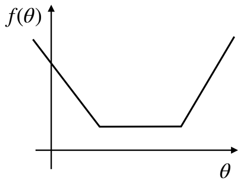
Identify the convexity for the following six functions (a-f). Simply write down whether the function is **convex** or **non-convex** for your answers.



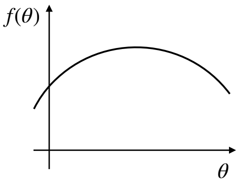
(a)



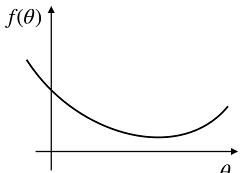
(b)



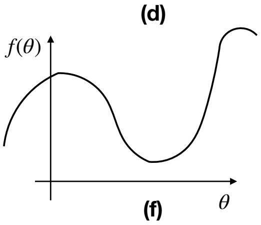
(c)



(d)



(e)



(f)

3 (7 points) Argmin and Argmax

An unknown estimator is given an estimation problem to find the minimizer and maximizer of the objective function $G(w) \in (0, 3]$:

$$(w_a, w_b) = (\arg\min_w G(w), \arg\max_w G(w)). \tag{1}$$

The solution to Eq. 1 by the estimator is $(w_a, w_b) = (15, 25)$.

Given this information, please obtain the value of w^* such that:

$$w^* = \arg\min_w [10 - 3 \times \ln(G(w))]. \tag{2}$$

4 (6 points) Error Metrics

Imagine there are actually 100 positive cases of SARS-CoV-2 infection among 1,000 people as verified by RT-PCR testing. You have an ML system that is trying to diagnose SARS-CoV-2 infection from recordings of coughing[1]. Your trained ML system makes predictions on these 1000 people, and the prediction outcomes are shown in Table 1.

- True Negative (TN): A case which is actually negative and predicted negative.
- True Positive (TP): A case which is actually positive and predicted positive.
- False Negative (FN): A case which is actually positive but predicted negative.
- False Positive (FP): A case which is actually negative but predicted positive.

	Predicted Negative	Predicted Positive
Negative Cases	810	90
Positive Cases	70	30

Table 1: Prediction outcome.

1. What's the precision score?

   A. 9/10
   B. 7/12
   C. 1/4
   D. 3/10

2. What's the recall (sensitivity) score?

   A. 1/4
   B. 3/10
   C. 9/10
   D. 7/12

3. What's the specificity score?

   A. 1/4
   B. 9/10
   C. 3/10
   D. 7/12

[1] Curious about the story in this problem? Check out `https://news.mit.edu/2020/covid-19-cough-cellphone-detection-1029`

5 (30 points) Coding: Hyper-plane Estimation

Assume we are given a dataset $S = \{(\boldsymbol{x}_i, y_i), i = 1, \ldots, n\}$. Here, $\boldsymbol{x}_i \in \mathbb{R}^2$ contains two features, i.e. $\boldsymbol{x}_i = [x_{i1}, x_{i2}]$. In this section, we aim to fit the data points with a hyper-plane:

$$y = w_0 + w_1 x_1 + w_2 x_2 \tag{3}$$

where $w_0, w_1, w_2 \in \mathbb{R}$ are three parameters to determine the hyper-plane and $\boldsymbol{x} = [x_1, x_2]^T$ represents the variable of two features. Then, we represent the data points as matrices $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^T$ and $Y = [y_1, y_2, \ldots, y_n]^T$, where $\mathbf{x}_i = [1, x_{i1}, x_{i2}]^T$ is a feature vector corresponding to $\boldsymbol{x}_i$. The parameters of the hyper-plane are also represented as a matrix $W = [w_0, w_1, w_2]^T$. Thus, we define a sum-of-squares error function $\mathcal{L}(W)$:

$$\mathcal{L}(W) = \|XW - Y\|_2^2 = (XW - Y)^T (XW - Y) \tag{4}$$

In this section, we attempt to obtain the best parameters W^* that minimizes the error function $\mathcal{L}(W)$ by **closed form solution** and **gradient descent**:

$$W^* = \arg\min_W \mathcal{L}(W) \tag{5}$$

Please fill in the blanks in the notebook `hyperplane-estimation.ipynb` to complete the steps needed to solve this question. You also need to download the data (`hyperplane-estimation.npy`) from canvas for this question.

6 (30 points) Coding: Parabola Estimation

We are given a dataset $S = \{(x_i, y_i), i = 1, \dots, n\}$. Here, x_i is a feature scalar. In this section, we aim to fit data points with a parabola:

$$y = w_0 + w_1 x + w_2 x^2 \tag{6}$$

where $w_0, w_1, w_2 \in \mathbb{R}$ are three parameters to determine the parabola. Then, we represent the data points as matrices $X = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^T$ and $Y = [y_1, y_2, \dots, y_n]^T$, where $\mathbf{x}_i = [1, x_i, x_i^2]^T$ is a feature vector corresponding to the data x_i. The parameters of the parabola are also represented as a matrix $W = [w_0, w_1, w_2]^T$. Next, we define some loss function $\mathcal{L}(W)$ and attempt to obtain the best parameters W^* that minimizes $\mathcal{L}(W)$. Please use the notebook `parabola-estimation.ipynb` (and its data file `parabola-estimation.npy`) to complete the steps needed to solve this question.

6.1 Squared L_2 Norm (5 points)

Consider squared L_2 norm as the loss function $\mathcal{L}(W)$:

$$\begin{aligned}
\mathcal{L}(W) &= \sum_{i=1}^{n} (\mathbf{x}_i^T W - y_i)^2 && (7)\\
&= \|XW - Y\|_2^2 && (8)\\
&= (XW - Y)^T (XW - Y). && (9)
\end{aligned}$$

Use the **closed form OLS solution** to compute $W^* = \arg\min_W \mathcal{L}(W)$ and plot the scatter graph of data and estimated parabola. Report the fitted parabola function parameters and the scatter graph in your submission.

6.2 L_1 Norm (10 points)

Consider L_1 norm as the loss function $\mathcal{L}(W)$:

$$\begin{aligned}
\mathcal{L}(W) &= \sum_{i=1}^{n} |\mathbf{x}_i^T W - y_i| && (10) \\
&= \|XW - Y\|_1 && (11)
\end{aligned}$$

Use **gradient descent** to find $W^* = \arg\min_W \mathcal{L}(W)$ and plot the scatter graph of data and estimated parabola. Report the parabola function and the figure in submission.

Hint:

1. You may refer to HW3 to derive the gradient of the loss function.

2. In `NumPy`, you can use `np.sign(x)` to compute the sign of matrix `x`.

6.3 Squared L_2 Norm and L_1 Norm (12 points)

Combine the squared L_2-norm and the L_1-norm:

$$\begin{aligned} \mathcal{L}(W) &= \sum_{i=1}^{n} \Big(\alpha \big(\mathbf{x}_i^T W - y_i \big)^2 + (1-\alpha) \big| \mathbf{x}_i^T W - y_i \big| \Big) && (12) \\ &= \alpha \left\| XW - Y \right\|_2^2 + (1-\alpha) \left\| XW - Y \right\|_1 && (13) \end{aligned}$$

Use the **gradient descent** to find W^* when $\alpha = 0$, $\alpha = 0.03$, $\alpha = 0.05$, $\alpha = 0.1$, and $\alpha = 1$ respectively. Plot the scatter graph of data and all estimated parabolas in one figure. Report the parabola functions and the figure in submission.

6.4 Comparison (3 points)

Compare the parabolas in Problem 6.3. Try to explain the trend from $\alpha = 0$ (i.e. L_1 norm), $\alpha = 0.03$, $\alpha = 0.05$, $\alpha = 0.1$, till $\alpha = 1$ (i.e. squared L_2 norm).

Hint: You may need to consider the outliers in the data points for your reasoning.