Homework Assignment 4

COGS 118A: Introduction to Machine Learning I

Due: 11:59pm, Tuesday, February 13th, 2018 (Pacific Time).

Instructions: Answer the questions below, attach your code, and insert figures to create a PDF file; submit your file via TritonEd (ted.ucsd.edu). You may look up the information on the Internet, but you must write the final homework solutions by yourself.

Late Policy: 5% of the total points will be deducted on the first day past due. Every 10% of the total points will be deducted for every extra day past due.

System Setup: You can install Anaconda to setup the Jupyter Notebook environment. Most packages have been already installed in Anaconda. If some package is not installed, you can use `pip` to install the missing package, that is, just type `pip install PACKAGE_NAME` in the terminal.

Grade: ____ out of 100 points

1 (40 points) Parabola Estimation

Similar to the Problem 3 in Homework 3, you will estimate a parabola function. As before, you are given the data $S = \{(x_i, y_i), i = 1, \ldots, n\}$. The data are expressed as matrices $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top$ and $Y = [y_1, y_2, \ldots, y_n]^\top$ where $\mathbf{x}_i$ is $[1, x_i, x_i^2]^\top$. The parabola function is defined as: $f(x; W) = w_0 + w_1 x + w_2 x^2 = \mathbf{x}^\top W$ where $\mathbf{x}$ is $[1, x, x^2]^\top$ and W is $[w_0, w_1, w_2]^\top$. Please download `hw4-q1-parabola.npy` from website as data source.

(a) Consider L_2 norm as your loss function:

$$g(W) = \|XW - Y\|_2^2 = \sum_{i=1}^{n} \big(y_i - f(x_i; W)\big)^2$$

Similar to Problem 3 in Homework 3, please use the **closed form solution** to compute W and plot the scatter graph of data and estimated parabola. Report the parabola function and the figures.

(b) Consider L_1 norm as your loss function:

$$g(W) = \|XW - Y\|_1 = \sum_{i=1}^{n} |y_i - f(x_i; W)|$$

(1) Derive the gradient (some hints are mentioned in the slides):

$$\frac{\partial g(W)}{\partial W} = \Big(\big(\text{sign}(XW - Y)\big)^\top X\Big)^\top$$

where

$$\text{sign}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0. \end{cases}$$

and if X is a matrix, $\text{sign}(X)$ means performing element-wise $\text{sign}(x_{ij})$ over all element x_{ij} in X.

(2) Similar to Problem 4 in Homework 3, use the **gradient descent method** to compute W and plot the scatter graph of data and estimated parabola. Report the parabola function and the figures.

Hint 1: In NumPy, you can use `np.sign(x)` to compute the sign of matrix `x`.

Hint 2: You may need to change the number of iterations to 300000, learning rate to 0.000001 and error threshold for W to 0.00001. Same settings may apply to sub-problem **(c)**.

(c) Consider L_1 and L_2 norm as your loss function:

$$\begin{aligned} g(W) &= \alpha \|XW - Y\|_2^2 + (1 - \alpha) \|XW - Y\|_1 \\ &= \sum_{i=1}^{n} \Big(\alpha\big(y_i - f(x_i; W)\big)^2 + (1 - \alpha)|y_i - f(x_i; W)|\Big) \end{aligned}$$

Similar to **(b)**, use the **gradient descent method** to compute W when $\alpha = 0.3$, $\alpha = 0.5$, and $\alpha = 0.7$ respectively and plot the scatter graph of data and estimated parabola. Report the parabola function and the figures.

(d) Compare **(a)** **(b)** and **(c)** and draw all curves in one figure with scatter graph of data. Try to explain the reason to **(1)** the position of each curve compared to the position of valid data points and outliers **(2)** difference between L_2 curve and L_1 curve **(3)** similarity among L_2 curve and $L_1 + L_2$ curves.

2 (30 points) Logistic Regression

In the binary classification problem, we need to predict a binary label $y \in \{0, 1\}$ for each input $\mathbf{x} = [x_0, x_1, \ldots, x_K]^\top$ where $x_0 = 1$. Note that we need to add a bias term $x_0 = 1$ to express linear functions with bias. Then, we parameterize conditional probability (or confidence) as

$$P(Y = 1|X = \mathbf{x}) = h(\mathbf{x}; \theta) = g(f(\mathbf{x}; \theta)) = \frac{1}{1 + e^{-f(\mathbf{x};\theta)}} = \frac{1}{1 + e^{-\sum_{k=0}^{K} \theta_k x_k}}$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

is called sigmoid function or logistic function. When $h(\mathbf{x}; \theta) \geq 0.5$, the classifier should predict label 1, and when $h(\mathbf{x}; \theta) < 0.5$, the classifier should predict label 0.

Given dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}$, we wish to optimize the following negative conditional log-likelihood $\mathcal{L}(\theta) = -\sum_i \ln P(y^{(i)}|\mathbf{x}^{(i)})$:

$$\mathcal{L}(\theta) = -\sum_i \left[y^{(i)} \ln h(\mathbf{x}^{(i)}; \theta) + (1 - y^{(i)}) \ln(1 - h(\mathbf{x}^{(i)}; \theta)) \right].$$

In this problem, we will minimize $\mathcal{L}(\theta)$ using gradient descent.
(a) Show that the gradient of $\mathcal{L}(\theta)$ is given by:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} = -\sum_i (y^{(i)} - h(\mathbf{x}^{(i)}))\mathbf{x}^{(i)}.$$

(b) Download the training data file `hw4-q2-logistic-train.npy`, test data file `hw4-q2-logistic-test.npy`, and the skeleton code `hw4-q2-logistic.ipynb` from the course website. Complete the skeleton code. In order to get full marks, your report should include:

1. Your code
2. Optimal θ^*
3. Equation of decision boundary corresponding to the optimal θ^*
4. Plot of training data along with decision boundary
5. Plot of test data along with decision boundary
6. Training accuaracy and test accuracy
7. Training curve.

3 (30 points) Linear Discriminant Analysis

Linear discriminant analysis has many applications, such as dimensionality reduction and feature extraction. In this problem, we consider a simple task. In data file `hw4-q3-lda.npy`, there are two classes: class 0 and class 1. The data are expressed as matrices X_0 for class 0 and X_1 for class 1. Each $X_j = [\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \ldots, \mathbf{x}_n^{(j)}]$. Note that in this problem we use **column vector** $\mathbf{x}_i^{(j)}$ for a single data point to simplify the calculation. Please fill the blanks in skeleton code `hw4-q3-lda.ipynb` to solve the following sub-problems:

(a) Compute the mean for each class, μ_0 and μ_1.

(b) Compute the covariance matrix for each class, Σ_0 and Σ_1.

The Fisher's linear discriminant analysis is defined to maximize criterion function:

$$S(\mathbf{w}) = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{within}}} = \frac{\left(\mathbf{w}^\top \mu_0 - \mathbf{w}^\top \mu_1\right)^2}{\mathbf{w}^\top (\Sigma_0 + \Sigma_1)\mathbf{w}}$$

An optimal solution $\mathbf{w}^*$ is:

$$\mathbf{w}^* = (\Sigma_0 + \Sigma_1)^{-1}(\mu_0 - \mu_1)$$

(c) Find the optimal $\widetilde{\mathbf{w}}^*$ with unit length.

Hint: The optimal $\mathbf{w}^*$ above is unnormalized. To normalize $\mathbf{w}^*$ to unit length in order to get $\widetilde{\mathbf{w}}^*$, you need to divide $\mathbf{w}^*$ by $||(\Sigma_0 + \Sigma_1)^{-1}(\mu_0 - \mu_1)||_2$, which is the L_2 norm of $\mathbf{w}^*$.

(d) Compute the projection on $\widetilde{\mathbf{w}}^*$ for each data point. Plot such projected data points with original data points in one figure.

Hint: Suppose we have a data point $\mathbf{x} = (x_1, x_2)^\top$, here, the data point $\mathbf{x}$ and $\widetilde{\mathbf{w}}^*$ are both column vectors. The projection on vector $\widetilde{\mathbf{w}}^*$ for $\mathbf{x}$ is simply the dot product:

$$\mathbf{x}_{\text{projected}} = \left((\widetilde{\mathbf{w}}^*)^\top \mathbf{x}\right)\widetilde{\mathbf{w}}^*$$