ISYE 6740/CSE 6740/CS 7641: HW 2

90 Points Total v1.0

Due: 11:59am Oct. 17 Tuesday

Name:

GT ID:

GT Account:

Instruction: Please write a report including answers to the questions and the plotted figures. Please write the code in `MATLAB` and submit your code in a **'zip' file** via T-Square. You can not use any existing package/library when solving these problems. You need to show the iterative procedures in the code. Your code is also supposed to have explanatory comments.

1) Ordinary Least Square Regression (10 points) We are given n observations denoted by $\{x_i, y_i\}_{i=1}^n$. For simplicity, we denote

$$X = \begin{pmatrix} x_1^\top \\ x_2^\top \\ \vdots \\ x_n^\top \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ x_{21} & \cdots & x_{2p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}, \quad y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{pmatrix},$$

where $X \in \mathbb{R}^{n \times p}$ is the design matrix, and $y \in \mathbb{R}^n$ is the response vector, and $\beta \in \mathbb{R}^p$ is the unknown regression coefficient vector.

The file "`MLR.csv`" is a 1000-by-31 matrix. Specifically, the i-th row corresponds to the i-th sample $\{x_i, y_i\}_{i=1}^n$. In other words, the first 30 columns correspond to X and the 31st column (last column) corresponds to y. The file "`True_Beta.csv`" contains the true regression coefficient vector $\beta^* \in \mathbb{R}^{30}$.

(a) Please implement the Ordinary Least Square estimator. (5 points)

(b) Please compute the squared error $\|\widehat{\beta} - \beta^*\|_2^2$. (5 points)

Submission Requirement: Please submit an executable Matlab script file named as "`HW2_P1_gtusername.m`", where "`gtusername`" is replaced by your GA Tech user id, which is similar to "`hjiang98`". We will test your code by "`HW2_P1_gtusername`" command in Matlab. Your code should read "`MLR.csv`" and "`True_Beta.csv`" from the same folder as your "`.m`" file. In terms of output, your code is supposed to print the squared error as (b) describes.

2) Gradient Descent for OLS (25 points) We apply the gradient descent algorithm to solve the following convex program:

$$\widehat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|_2^2.$$

The step size parameter is chosen as $1/L$, where L is the largest eigenvalue of $\frac{1}{n} X^\top X$. Please run the algorithm for 1000 iterations.

(a) Please use the all-zero vector as the initialization. Please plot a figure of $\log(f(\beta^{(k)}) - f(\widehat{\beta}))$, where $f(\cdot)$ denotes the objective function, and k is the iteration index. (10 points)

(b) Please plot a figure of $\|\beta^{(k)} - \beta^*\|_2^2$ versus the number of iterations. (5 points)

(c) Recall that $\widehat{\beta}$ is the OLS estimator in Problem 1. Please plot a figure of $\|\beta^{(k)} - \widehat{\beta}\|_2^2$ versus the number of iterations. (5 points)

(d) What can you tell from the figures in (b) and (c)? (5 points)

Submission Requirement: For parts (a), (b) and (c), please submit an executable Matlab script file named as "`HW2_P2_gtusername.m`". We will test your code by "`HW2_P2_gtusername`" command in Matlab. Your code should read "`MLR.csv`" and "`True_Beta.csv`" from the same folder as your "`.m`" file. In terms of output, your code is supposed to plot three separate figures as parts (a), (b) and (c) describe. After plotting each figure, please use "`pause`" command to pause your program.

3) Stochastic Gradient Descent for OLS (35 points) We apply the stochastic gradient descent algorithm to solve the optimization problem in Problem 2. For notational simplicity, we rewrite it as

$$\widehat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2}(y_i - x_i^\top \beta)^2.$$

For simplicity, we do not randomly select samples in each iteration. We simply iterate over all samples in a cyclic order. Specifically, at the first iteration, we use (x_1, y_1); At the second iteration, we use (x_2, y_2), ...; When we exhaust all samples, we restart with (x_1, y_1). Here, one **pass** of the data means your algorithm goes through the whole dataset for one time, which is equivalent to n iterations in SGD. Please run **20 passes** with step size $0.1/L$, where L is defined in Problem 2.

(a) Please plot a figure of $\log(f(\beta^{(k)}) - f(\widehat{\beta}))$, where $f(\cdot)$ denotes the objective function, and k is the number of passes. (5 points)

(b) Please plot a figure of $\|\beta^* - \beta^{(k)}\|_2^2$ versus number of passes. (5 points)

(c) Please plot a figure of $\|\widehat{\beta} - \beta^{(k)}\|_2^2$ versus number of passes. (5 points)

(d) Please repeat the above experiments with different step sizes $(1.7/L, 1/L, 0.01/L)$. (15 points)

(e) Please compare the above experiments. What can you tell from the above experiments? (5 points)

Submission Requirement: You need to submit your code for parts (a), (b), (c), and (d). Similarly, please submit an executable Matlab script file named as "`HW2_P3_gtusername.m`". In terms of output, your code is supposed to plot 3 separate figures as parts (a), (b), (c), and (d) describe. After plotting each figure please use "`pause`" command to pause your program.

4) Email Spam Filter Via Discriminant Analysis (20 points) We build spam email filters based on four models:

Gaussian Discriminant Analysis (GDA)

Naive Bayes Gaussian Discriminant Analysis (NB-GDA)

Naive Bayes Bernoulli Discriminant Analysis (NB-BDA)

Quadratic Discriminant Analysis (QDA)

The data set has been cleaned and saved in"`spamdata.mat`". For the details of these models, please refer to the slides of Lecture 2.

(a) Please use GDA to build an email spam filter, and report the training error and testing error. (5 points)

(b) Please use NB-GDA to build an email spam filter, and report the training error and testing error. (5 points)

(c) Please use NB-BDA to build, and report the training error and testing error. If the parameter of Bernoulli distribution you estimated is 0, you can replace it with a small value (0.01). (5 points)

(d) Please use QDA to build an email spam filter, and report the training error and testing error. If you find out the estimated covariance matrix is singular, you can add a small identity matrix to it ($0.01I$). (5 points)

Submission Requirement: Similarly, please submit an executable Matlab script file named as "`HW2_P4_gtusername.m`". Your program is supposed to load "'`spamdata.mat`" from the same folder and print the training error and testing error for each model in the order of model (a),(b),(c) and (d).