CIS 520, Machine Learning, Fall 2020
Homework 2
Due: Monday, September 28th, 11:59pm
Submit to Gradescope

Instructions. Please write up your responses to the following problems clearly and concisely. We require you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Gradescope. We will not accept paper copies of the homework.**

Collaboration. You are allowed and encouraged to work together. You may discuss the **written homework** to understand the problem and reach a solution in groups. However, **it is recommended that each student also write down the solution independently and without referring to written notes from the joint session.** You must understand the solution well enough to reconstruct it by yourself. (This is for your own benefit: you have to take the exams alone.)

Assignment Policies

Instructions. Please write up your responses to the following problems clearly and concisely. We require you to write up your responses using LaTeX; we have provided a LaTeX template, available on Canvas, to make this easier. **Submit your answers in PDF form to Gradescope. We will not accept paper copies of the homework.**

Learning Objectives

After completing this assignment, you will:

- Understand how linear regression interacts with the number of training examples
- Be able to recognize tradeoffs in performance between OLS and gradient descent linear regression
- Understand how linear regression relates to squared error
- Be able to compute the MLE of any distribution

Deliverables

This homework can be completed individually or in groups of 2. You need to make one submission per group. Make sure to add your team member's name on Gradescope when submitting the homework's written and coding part.

1. **A PDF compilation of `hw2.tex` with team member's names in the agreement**
2. **A `.ipynb` file with the functions implemented**

Homework Submission Instructions

Written Homeworks

All written homework **must** be submitted as a PDF to Gradescope. **Handwritten assignments (scanned or otherwise) will not be accepted.** We require the use of LaTeXto generate your final PDF. We will be posting the homeworks in both PDF and LaTeX source form, and we encourage you to use this source as a template for your submission. **We recommend using Overleaf,** a free online LaTeX editor, though you are welcome to edit assignments however you like.

Coding Homeworks

All coding assignments will be done in Jupyter Notebooks. We will provide a `.ipynb` template for each assignment as well as function stubs for you to implement. Your final submission will be a `.ipynb` file compiled from your notebook submitted to Gradescope. Though you are free to use your own installation of Jupyter for the assignments, **we recommend using Google Colab**, which provides a Jupyter environment connected to Google Drive along with a hosted runtime containing a CPU and GPU.

1 Programming: Least Squares Regression [35 points]

[**11 points**] Write a small piece of Python code to implement linear least squares regression, with and without L2 regularization. The input to your code is a training data set ($\mathbf{X}$,y), where $\mathbf{X}$ is an $m \times d$ matrix of training examples and $\mathbf{y}$ is an m-dimensional vector of real-valued labels associated with the instances in $\mathbf{X}$. You should **not** be importing any modules besides numpy to work with matrices. You will be graded for the correctness of code worth 11 points.

You are provided with two data sets for this problem: a synthetic 1-dimensional data set (data set 1), and a real 8-dimensional data set (data set 2); each data set is split into training and test sets.

1.1 Data Set 1 (synthetic 1-dimensional data)

This data set contains 100 training examples and 1000 test examples, all generated i.i.d. from a fixed probability distribution. For this data set, you will run unregularized least squares regression.

1. **[4 points] Learning Curve.** Use your implementation of (unregularized) least squares regression to learn a regression model from 10% of the training data, then 20% of the training data, then 30% and so on up to 100% (separate files containing r% of the training examples are provided under the folder for this problem with file names **Data-set-1/Train-subsets/X_train_r%.txt**, and the corresponding labels are provided with the file names **y_train_r%.txt** in the same folder). In each case, measure both the L_1 and L_2 error on the training examples used, as well as the error on the given test set. Plot a curve showing both errors (on the *y-axis*) as a function of the number of training examples used (on the *x-axis*).

2. **[4 points] Analysis of model learned from full training data.** Write down the weight and bias terms, $\widehat{w}$ and $\widehat{b}$, learned from the full training data. Also, write down the L_2 training and test error of this model. In a single figure, draw a plot of the learned linear function (input instance on *x-axis* and the predicted value on the *y-axis*), along with a scatter plot depicting the true label associated with each test instance.

1.2 Data Set 2 (real 8-dimensional data)

This is a real data set that involves predicting median house value from the location coordinates, demographics and the number of rooms and bedrooms in the houses in total in the block (or district). The data set is a subset of a larger dataset curated for a research conducted by Pace,R. Kelly and Ronald Barry in 1997.This subset has 960 training examples and 240 test examples. For this data set, you will run both unregularized least squares regression and L_2-regularized least squares regression (ridge regression).

1. **[4 points] Regression on 5% of the training data.** Use your implementation of L_2-regularized least squares regression to learn a model on 5% of the training data. Select the regularization parameter from the range {0.1,1,10,50,100,150,200,500,1000,2500,5000} using 5-fold cross validation on the relevant training data. Draw a plot showing λ on the *x-axis* and the training, test, and cross validation errors on the *y-axis* using the L_2 error. Then record the chosen value of λ along with the weight vector, bias term, and all corresponding errors for the chosen value of λ.

2. **[4 points] Regression on 100% of the training data.** Repeat the above process, but instead learn from the full training data for L_2-regularized regression. Plot all of the errors, and record the chosen value of λ along with the weight vector, bias term, and all corresponding errors for the chosen value of λ.

3. **[4 points] Comparison of models learned by two methods** For each of the two training sets considered above (5% and 100%), compare the training and test errors of the models learned using ridge regression. What can you conclude from this about the value of regularization for small and large training sets?

4. **[4 points] Theoretical Value of λ.** For each of the two training sets considered above (5% and 100%), Which λ should be larger by theory?why? Do those values align with the conclusion you made in part 1.3?

2 Programming: Batch Gradient Descent [15 points]

[4 points] Write a small piece of Python code to implement batch gradient descent for unregularized least squares regression. The input to your code is a training data set ($\mathbf{X}$,y), where $\mathbf{X}$ is an $m \times d$ matrix of training examples and $\mathbf{y}$ is an m-dimensional vector of real-valued labels associated with the instances in $\mathbf{X}$, a non-negative integer number of iterations to perform batch gradient descent ($\mathbf{N}$), and the learning rate

(α). You may **not** import any modules besides numpy to work with matrices. Your implementation will be graded for correctness, worth 4 points.

1. **[4 points] OLS runtime.** Time the closed-form unregularized linear regression implementation you wrote in previous section on the full training data for Data Set 1. Write down the weight and bias terms, $\widehat{w}$ and $\widehat{b}$, learned from the full training data, as well as the L_2 error on the test data, and the time it took to run the full process.

2. **[4 points] Gradient descent runtime.** Time the gradient descent implementation you just wrote on the full training data for Data Set 1 with iterations from range {100, 1000, 2000}, and a learning rate of 0.01. Write down the weight and bias terms, $\widehat{w}$ and $\widehat{b}$, learned from the full training data, as well as the L_2 error on the test data, and the time it took to run the full process.

3. **[3 points] Comparison of algorithms.** Which algorithm runs faster? Why might that be the case? Why would we ever use gradient descent linear regression in practice while a closed form solution exists?

3 Regression Models and Squared Errors [25 points]

Regression problems involves instance spaces $\mathcal{X}$ and labels, and the predictions, which are real-valued as $\mathcal{Y} = \widehat{\mathcal{Y}} = \mathbb{R}$. One is given a training sample $S = ((x_1, y_1), ..., (x_m, y_m)) \in (\mathcal{X} \times \mathbb{R})^m$, and the goal is to learn a regression model $f_S : \mathcal{X} \to \mathbb{R}$. The metric used to measure the performance of this regression model can vary, and one such metric is the squared loss function. The questions below ask you to work with regression problems and squared error losses.

1. **[10 points]** The squared error is given by $\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x) - y)^2]$, where the examples are drawn from a joint probability distribution $p(X, Y)$ on $\mathcal{X} \times \mathbb{R}$. Find the lower bound of the expression $\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x) - y)^2]$. From this lower bound, what is the optimal expression of $f(x)$, in terms of x and Y?

   Hint: Let $\widehat{y} = \widehat{y}(x)$ be the estimated regression model, and think about how to include $\widehat{y}$ into $\mathbb{E}_{(x,y)\sim p(X,Y)}[(f(x) - y)^2]$ to derive squared difference terms between $\widehat{y}$ and f, $\widehat{y}$ and y. When is $\widehat{y}$ optimal?

2. **[5 points]** With this result, complete the following two problems. Consider the regression task in which instances contain two features, each taking values in $[0, 1]$, so that the instance space is $\mathcal{X} = [0, 1]^2$, and with label and prediction spaces belonging to the real space. Suppose examples $(\mathbf{x}, y)$ are drawn from the joint probability distribution D, whose marginal density on $\mathcal{X}$ is given by

   $$\mu(\mathbf{x}) = 2x_1, \quad \forall \mathbf{x} = (x_1, x_2) \in \mathcal{X}$$

   and the conditional distribution of Y given $\mathbf{x}$ is given by

   $$Y|X = \mathbf{x} \sim \mathcal{N}(x_1 - 2x_2 + 2, 1)$$

   What is the optimal regression model $f^*(X)$ and the minimum achievable squared error for D?

3. **[2 points]** Suppose you give your friend a training sample $S = ((\mathbf{x}_1, y_1), ..., (\mathbf{x}_m, y_m))$ containing m examples drawn i.i.d from D, and your friend learns a regression model given by

   $$f_S(\mathbf{x}) = x_1 - 2x_2, \quad \forall \mathbf{x} = (x_1, x_2) \in \mathcal{X}$$

   Find the squared error of f_S with respect to D.

4. **[8 points]** Consider a linear model of the form

$$f(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{P} w_i x_i$$

together with a sum of squares error function of the form

$$L_P(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} (f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2$$

where P is the dimensionality of the vector $\mathbf{x}$, N is the number of training examples, and $\mathbf{t}$ is the ground truth target . Now suppose that the Gaussian noise ϵ_i with zero mean and variance σ^2 is added independently to each of the input variables x_i. By making use of $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i \epsilon_j] = \delta_{ij}\sigma^2$, show that minimizing L_P averaged over the noise distribution is equivalent to minimizing the sum of squares error for noise-free input variables L_P with the addition of a weight-decay regularization term, in which the bias parameter w_0 is omitted from the regularizer.

4 Maximum Likelihood Estimation [25 points]

1. (a) **[3 points]** Suppose that there are blue (B) and red (R) balls in a box and the frequency of blue balls is θ. That is, a random draw from the basket will result in drawing a blue ball with a probability θ and drawing a red ball with a probability $1-\theta$. Let's say each person draws two balls replacing the ball after each draw. So a person can draw either one of these three combinations (BB, BR, RR). What are the probabilities of each of the outcomes?

   (b) **[5 points]** Suppose that in the population p_1 people draw BB, p_2 people draw BR and p_3 people draw RR. What is the log likelihood function $LL(P(D/\theta)$ Find the Maximum Likelihood estimate of θ?

   (c) **[2 points]** Suppose that out of 100 people, 50 draw BB combination, 10 draw BR combination and 40 draw RR combination. What is the MLE estimate of θ (fraction of blue balls)?

2. **[5 points]** We have a dataset with N records in which the i^{th} record has one real-valued input attribute x_i and one real-valued output attribute y_i. The model has one unknown parameter w to be learned from data, and the distribution of y_i is given by

$$y_i \sim \mathcal{N}(\log(wx_i), 1)$$

Suppose you decide to do a maximum likelihood estimation of w. What equation does w need to satisfy to be a maximum likelihood estimate?

3. **[10 points]** Consider a linear basis function regression model for a multivariate target variable $\mathbf{t}$ having a Gaussian distribution of the form

$$p(\mathbf{t}|\mathbf{W}, \mathbf{\Sigma}) = \mathcal{N}(\mathbf{t}|f(\mathbf{x}, \mathbf{W}), \mathbf{\Sigma})$$

where

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}^T \phi(\mathbf{x})$$

together with a training data set comprising input basis vectors $\phi(\mathbf{x}_n)$ and corresponding target vectors $\mathbf{t}_n$ with $n = 1, 2, \ldots, N$. Show that the maximum likelihood solution $\mathbf{W}^*$ for the parameter matrix $\mathbf{W}$ has the property that each column is given by the solution to a univariate target variable. Note that this is independent of the covariance matrix $\mathbf{\Sigma}$.

Also, give the maximum likelihood solution for $\mathbf{\Sigma}$ – feel free to use standard results for the MLE solution of Σ in your answer.