

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

Your name(s) here

## 1 Programming: Least Squares Regression

### 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. Learning Curve.

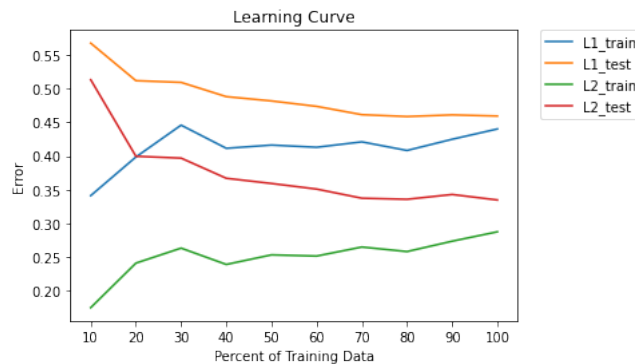


Figure 1: Curves showing  $L1$  and  $L2$  errors as a function of number of training samples used.

#### 2. Analysis of model learned from full training data.

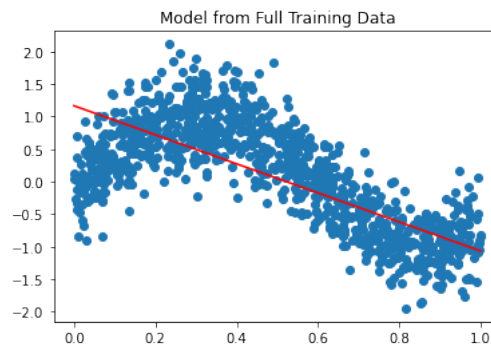


Figure 2: Plot of learned linear function and scatter plot depicting true test labels.

```

w:  [[-2.23442304]]
b:  1.1671738819238153
L2 training error: 0.2876339586619919
L2 test error      : 0.33475968697353337

```

## 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. Regression on 5% of the training data.

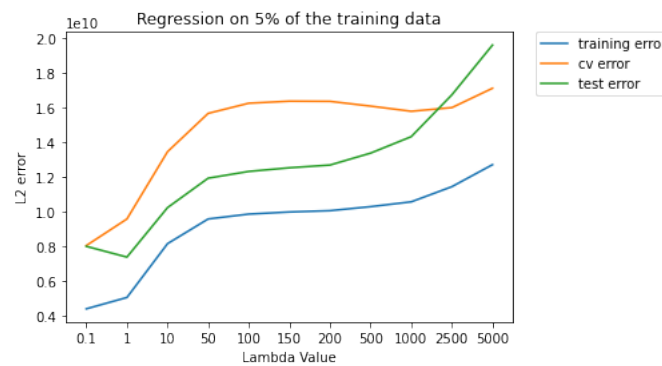


Figure 3: Plot showing  $\lambda$  and cross-validation loss.

```

lambda: 0.1
b: -3064.11067472829
w: [ [ 3.38723280e+03]
      [ 7.58457227e+03]
      [ 2.69630490e+03]
      [-6.17752959e+01]
      [ 3.01287546e+02]
      [-2.03570908e+01]
      [ 1.48763631e+02]
      [ 5.06386976e+04]]
training error 4418834306.376352
test error 8020741257.03428
val error 4856822219.80739

```

## 2. Regression on 100% of the training data.

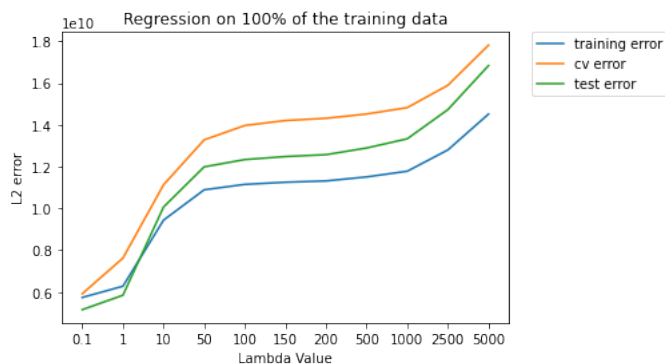


Figure 4: Plot showing  $\lambda$  and cross-validation loss.

```
lambda: 0.1
b: -1727.2134438706562
w: [ [-1.33350946e+03]
      [-5.25444158e+03]
      [ 1.50281613e+03]
      [-3.44638514e+01]
      [ 1.52907091e+02]
      [-3.01196974e+01]
      [ 1.15840351e+02]
      [ 4.78933125e+04]]
training error 5736263552.429791
test error 5152682299.804243
val error 6672178868.376767
```

3. **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?

100% dataset:

```
training error 5.74e9
test error 5.15e9
```

5% dataset:

```
training error 4.42e9
test error 8.02e9
```

On the small training set, training error is lower but test error is much higher. This points to the fact that the regularization parameter is effective in preventing overfitting in large datasets, but not as effective in smaller datasets.

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

As the size of the dataset grows, the size of the optimal  $\lambda$  should shrink, in-line with theory.

## 2 Programming: Batch Gradient Descent

1. **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,  $\hat{w}$  and  $\hat{b}$ , 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.

$$\begin{aligned}\hat{w} &= -2.23442 \\ \hat{b} &= 1.16717 \\ L2 \text{ error} &= 0.33476 \\ \text{time} &= 0.00133\end{aligned}$$

2. **Gradient descent runtime.** Time the gradient descent implementation you just wrote on the full training data for Data Set 1 with iterations from range  $\{10, 100, 1000\}$ , and a learning rate of 0.01. Write down the weight and bias terms,  $\hat{w}$  and  $\hat{b}$ , 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.

### 100 Iterations:

$$\begin{aligned}\hat{w} &= 0.0300179 \\ \hat{b} &= 0.09480728 \\ L2 \text{ error} &= 0.6611824852426563 \\ \text{time} &= 0.002912282943725586\end{aligned}$$

### 1000 Iterations:

$$\begin{aligned}\hat{w} &= 0.75698276 \\ \hat{b} &= -1.47418892 \\ L2 \text{ error} &= 2.4401471307741174 \\ \text{time} &= 0.030885696411132812\end{aligned}$$

### 2000 Iterations:

$$\begin{aligned}\hat{w} &= 1.04606972 \\ \hat{b} &= -2.00997274 \\ L2 \text{ error} &= 3.698775500730846 \\ \text{time} &= 0.044022321701049805\end{aligned}$$

3. **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?

In this example, the closed form solution was faster. In practice, as the number of data points scales, gradient descent becomes faster because it does not require inverting the  $n \times p$  sized matrix, and converges quickly.

## 3 Regression Models and Squared Errors

Regression problems involves instance spaces  $\mathcal{X}$  and labels, and the predictions, which are real-valued as  $\mathcal{Y} = \hat{\mathcal{Y}} = \mathbb{R}$ . One is given a training sample  $S = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathbb{R})^m$ , and the goal is to learn

a regression model  $f_S : \mathcal{X} \rightarrow \mathbb{R}$ . 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. 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  $Y$ ?

$$\begin{aligned}
\mathbb{E}_{(x,y) \sim p(X,Y)}[(f(x) - y)^2] &= \mathbb{E}_{(x,y) \sim p(X,Y)}[(f(x) + \mathbb{E}_y(y|x) - \mathbb{E}_y(y|x) - y)^2] \\
&= \mathbb{E}_{(x,y) \sim p(X,Y)}[(f(x) - \mathbb{E}_y(y|x))^2 + (f(x) - \mathbb{E}_y(y|x))(\mathbb{E}_y(y|x) - y) + (\mathbb{E}_y(y|x) - y)^2] \\
&= \mathbb{E}_{(x,y) \sim p(X,Y)}[(f(x) - \mathbb{E}_y(y|x))^2] \\
&\quad + \mathbb{E}_{(x,y) \sim p(X,Y)}[(f(x) - \mathbb{E}_y(y|x))(\mathbb{E}_y(y|x) - y)] \\
&\quad + \mathbb{E}_{(x,y) \sim p(X,Y)}[(\mathbb{E}_y(y|x) - y)^2] \\
&= \mathbb{E}_{(x,y) \sim p(X,Y)}[(f(x) - \mathbb{E}_y(y|x))^2] + \mathbb{E}_{(x,y) \sim p(X,Y)}[(\mathbb{E}_y(y|x) - y)^2] \\
&\geq \mathbb{E}_{(x,y) \sim p(X,Y)}[(\mathbb{E}_y(y|x) - y)^2]
\end{aligned}$$

$$f(x) = \mathbb{E}_y(y|x)$$

2. 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  $D$ ?

$$f(x) = \mathbb{E}_y(Y|X) = x_1 - 2x_2 + 2$$

$$\begin{aligned}
L_D[f^*] &= \mathbb{E}_D[(\mathbb{E}_y(Y|X) - Y)^2] = \mathbb{E}_D[(x_1 - 2x_2 + 2 - y)^2] \\
&= \int_0^1 \int_{\mathbb{R}} ((x_1 - 2x_2 + 2 - y)^2 (2x_1) \mathcal{N}(x_1 - 2x_2 + 2, 1)) dy dx \\
&= \int_0^1 2x_1 \int_{\mathbb{R}} ((x_1 - 2x_2 + 2 - y)^2 \mathcal{N}(x_1 - 2x_2 + 2, 1)) dy dx
\end{aligned}$$

Let  $a = x_1 - 2x_2 + 2$ ,

$$\begin{aligned}
\int_{\mathbb{R}} ((x_1 - 2x_2 + 2 - y)^2 \mathcal{N}(x_1 - 2x_2 + 2, 1)) dy &= \int_{\mathbb{R}} (a - y)^2 \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-a)^2}{2}} dy \\
&= \int_{\mathbb{R}} y^2 \frac{1}{\sqrt{2\pi}} \times e^{-\frac{y^2}{2}} dy \\
&= \mathbb{E}(Y^2) \\
&= (\mathbb{E}(Y))^2 + \text{Var}(Y) \\
&= 0 + 1 = 1
\end{aligned}$$

Thus,

$$\begin{aligned} L_D[f^*] &= \int_0^1 2x_1 dx_1 \\ &= 1 \end{aligned}$$

3. Suppose you give your friend a training sample  $S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m))$  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  $D$ .

$$\begin{aligned} L_D[f_S] &= \mathbb{E}_D[(x_1 - 2x_2 - y)^2] \\ &= \int_0^1 \int_{\mathbb{R}} ((x_1 - 2x_2 + 2 - y - 2)^2 (2x_1) \mathcal{N}(x_1 - 2x_2 + 2, 1)) dy dx \\ &= \int_0^1 2x_1 \int_{\mathbb{R}} (x_1 - 2x_2 + 2 - y - 2)^2 \mathcal{N}(x_1 - 2x_2 + 2, 1) dy dx \end{aligned}$$

Following the steps in part 2, we have

$$\begin{aligned} \int_{\mathbb{R}} (x_1 - 2x_2 + 2 - y - 2)^2 \mathcal{N}(x_1 - 2x_2 + 2, 1) dy &= \int_{\mathbb{R}} ((y - a)^2 + 4(y - a) + 4) \mathcal{N}(a, 1) dy \\ &= \int_{\mathbb{R}} (y - a)^2 \mathcal{N}(a, 1) dy \\ &\quad + \int_{\mathbb{R}} 4(y - a) \mathcal{N}(a, 1) dy \\ &\quad + \int_{\mathbb{R}} 4 \mathcal{N}(a, 1) dy \\ &= 1 + 0 + 4 = 5 \end{aligned}$$

Thus,

$$\begin{aligned} L_D[f_S] &= 5 \int_0^1 2x_1 dx \\ &= 5 \end{aligned}$$

4. 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  $\epsilon_i$  with zero mean and variance  $\sigma^2$  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.

Firstly, we take  $N=1$

$$\begin{aligned}
\mathbb{E}[\tilde{L}] &= \mathbb{E}\left[\frac{1}{2}(f(\mathbf{x}, \mathbf{w}) - t)^2\right] \\
&= \mathbb{E}\left[\frac{1}{2}\left(w_0 + \sum_{i=1}^P w_i(x_i + \epsilon_i) - t\right)^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[w_0^2 + t^2 + \sum_{i=1}^P w_i^2 x_i^2 + w_0\left(w_0 + \sum_{i=1}^P w_i x_i - t\right) - t\left(w_0 + \sum_{i=1}^P w_i x_i - t\right) + \sum_{i=1}^P w_i x_i \left(w_0 + \sum_{j \neq i}^P w_j x_j - t\right)\right] \\
&= \frac{1}{2}(f(\mathbf{x}, \mathbf{w}) - t)^2 + \frac{\sigma^2}{2} \sum_{i=1}^P w_i^2
\end{aligned}$$

In general,

$$\begin{aligned}
\mathbb{E}[\tilde{L}] &= \frac{1}{2} \sum_{n=1}^N (f(\mathbf{x}_n, \mathbf{w}) - \mathbf{t}_n)^2 + \frac{N\sigma^2}{2} \sum_{i=1}^P w_i^2 \\
&= L_P(\mathbf{w}) + \frac{N\sigma^2}{2} \sum_{i=1}^P w_i^2
\end{aligned}$$

## 4 Maximum Likelihood Estimation

1. (a) Suppose that there are blue (B) and red (R) balls in a box and the frequency of blue balls is  $\theta$ . That is, a random draw from the basket will result in drawing a blue ball with a probability  $\theta$  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?

$$\begin{aligned}
P(BB) &= \theta^2 \\
P(BR) &= 2\theta(1 - \theta) \\
P(RR) &= (1 - \theta)^2
\end{aligned}$$

- (b) 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  $\theta$ ?

$$\begin{aligned}
LL(P(D/\theta)) &= \frac{1}{p_1 + p_2 + p_3} [p_1 \times \log P(BB) + p_2 \times \log P(BR) + p_3 \times \log P(RR)] \\
&= \frac{(2p_1 + p_2) \times \log \theta + (2p_3 + p_2) \times \log(1 - \theta) + p_2 \times \log 2}{p_1 + p_2 + p_3}
\end{aligned}$$

The derivative vanishes at the maximum:

$$0 = LL'(\theta) = \frac{1}{p_1 + p_2 + p_3} \left( \frac{2p_1 + p_2}{\theta} - \frac{2p_3 + p_2}{1 - \theta} \right)$$

The derivative vanishes at the maximum. Thus,

$$MLE(\theta)(\hat{\theta}) = \frac{2p_1 + p_2}{2p_1 + p_2 + p_3}$$

- (c) 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  $\theta$  (fraction of blue balls)? Plugging into the equation above

$$\hat{\theta} = \frac{2 * 50 + 10}{200} = \frac{11}{20}$$

2. 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?

Firstly, we calculate LL

$$\begin{aligned} LL &= -\log\left(\prod_{i=1}^N \mathcal{N}(\log_{10}(wx_i), 1)\right) \\ &= \frac{1}{2} \sum_{i=1}^N (y_i - \log(wx_i))^2 + C \end{aligned}$$

Take  $LL'(w) = 0$ , we have

$$w = \exp\left(\frac{\sum_{i=1}^N (y_i - \log(x_i))}{N}\right)$$

3. 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  $n = 1, 2, \dots, N$ . 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  $\mathbf{\Sigma}$  in your answer.

$$\begin{aligned} LL &= -\log\left(\prod_{n=1}^N \mathcal{N}(\mathbf{t}_n|\mathbf{W}^T \phi(\mathbf{x}_n), \mathbf{\Sigma})\right) \\ &= \frac{1}{2} \sum_{n=1}^N (\mathbf{t}_n - \mathbf{W}^T \phi(\mathbf{x}_n))^T \mathbf{\Sigma}^{-1} (\mathbf{t}_n - \mathbf{W}^T \phi(\mathbf{x}_n)) + C \end{aligned}$$

From the above, we have

$$LL'(\mathbf{W}) = -\mathbf{\Sigma}^{-1} \sum_{n=1}^N (\mathbf{t}_n - \mathbf{W}^T \phi(\mathbf{x}_n)) \phi(\mathbf{x}_n)^T$$

Take  $LL'(\mathbf{W}) = 0$

$$\mathbf{W}^* = (\Phi^T \Phi)^{-1} \Phi^T$$

Where  $\Phi = [\phi(\mathbf{x}_1), \dots, \phi(\mathbf{x}_N)]^T$ ,  $T = [\mathbf{t}_1, \dots, \mathbf{t}_N]^T$ .

Then  $i$ th column of  $\mathbf{W}^*$  is  $(\Phi^T \Phi)^{-1} \Phi^T \mathbf{t}^i$ , which is the solution to target value  $t_n^i$ , i.e. the  $i$ th component



of  $\mathbf{t}_n$ . So each column of  $\mathbf{W}^*$  is given by the solution to a univariate target variable. The maximum likelihood solution for  $\mathbf{\Sigma}$  is

$$\mathbf{\Sigma}^* = \frac{1}{N} \sum_{n=1}^N (\mathbf{t}_n - \mathbf{W}^{*T} \phi(\mathbf{x}_n)) \phi(\mathbf{x}_n)^T$$