COL 774: Machine Learning. Assignment 1

Due Date: 11:50 pm, Friday Sep 9, 2022. Total Points: 80

Notes:

- You should submit all your code as well as any graphs that you might plot. Do not submit answers to theoretical questions.
- Do not submit the datasets.
- Include a **single write-up (pdf) file** which includes a brief description for each question explaining what you did. Include any observations and/or plots required by the question in this single write-up file.
- You should use Python for all your programming solutions.
- Your code should have appropriate documentation for readability.
- You will be graded based on what you have submitted as well as your ability to explain your code.
- Refer to the course website for assignment submission instructions.
- This assignment is supposed to be done individually. You should carry out all the implementation by yourself.
- We plan to run Moss on the submissions. We will also include submissions from previous years since some of the questions may be repeated. Any cheating will result in a zero on the assignment, an addtional penalty of the negative of the total weightage of the assignment and possibly much stricter penalties (including a **fail grade** and/or referring to a **DisCo**).
- Many of the problems below have been adapted from the Machine Learning course offered by Andrew Ng at Stanford.
- You should normalize the data (x's) to have zero mean and unit variance in each dimension for Q1, Q3 and Q4, as described in class. Do Not perform any normalization for Q2.

1. **(20 points) Linear Regression**
   In this problem, we will implement least squares linear regression to predict density of wine based on its acidity. Recall that the error metric for least squares is given by:

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right)^2$$

   where $h_\theta(x) = \theta^T x$ and all the symbols are as discussed in the class. The files linearX.csv and linearY.csv contain the acidity of the wine ($x^{(i)}$'s, $x^{(i)} \in \mathcal{R}$) and its density ($y^{(i)}$'s, $y^{(i)} \in \mathcal{R}$), respectively, with one training example per row. We will implement least squares linear regression to learn the relationship between $x^{(i)}$'s and $y^{(i)}$'s.

   (a) (**8 points**) Implement batch gradient descent method for optimizing $J(\theta)$. Choose an appropriate learning rate and the stopping criteria (as a function of the change in the value of $J(\theta)$). You can initialize the parameters as $\theta = \vec{0}$ (the vector of all zeros). Do not forget to include the intercept term. Report your learning rate, stopping criteria and the final set of parameters obtained by your algorithm.

(b) (**3 points**) Plot the data on a two-dimensional graph and plot the hypothesis function learned by your algorithm in the previous part.

(c) (**3 points**) Draw a 3-dimensional mesh showing the error function ($J(\theta)$) on z-axis and the parameters in the $x - y$ plane. Display the error value using the current set of parameters at each iteration of the gradient descent. Include a time gap of 0.2 seconds in your display for each iteration so that the change in the function value can be observed by the human eye.

(d) (**3 points**) Repeat the part above for drawing the contours of the error function at each iteration of the gradient descent. Once again, chose a time gap of 0.2 seconds so that the change be perceived by the human eye.(Note here plot will be 2-D)

(e) (**3 points**) Repeat the part above (i.e. draw the contours at each learning iteration) for the step size values of $\eta = \{0.001, 0.025, 0.1\}$. What do you observe? Comment.

2. **(20 points) Sampling and Stochastic Gradient Descent**
In this problem, we will introduce the idea of sampling by adding Gaussian noise to the prediction of a hypothesis and generate synthetic training data. Consider a given hypothesis h_θ (*i.e.* known $\theta_0, \theta_1, \theta_2$) for a data point $x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}$. Note that $x_0 = 1$ is the intercept term.

$$y = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Adding Gaussian noise, equation becomes

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \epsilon$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$

To gain deeper understanding behind Stochastic Gradient Descent (SGD), we will use the SGD algorithm to learn the original hypothesis from the data generated using sampling, for varying batch sizes. We will implement the version where we make a complete pass through the data in a round robin fashion (after initially shuffling the examples). If there are r examples in each batch, then there is a total of $\frac{m}{r}$ batches assuming m training examples. For the batch number b ($1 \leq b \leq \frac{m}{r}$), the set of examples is given as: $\{x^{(i_1)}, x^{(i_2)}, \cdots, x^{(i_r)}\}$ where $i_k = (b-1)r + k$. The Loss function computed over these r examples is given as:

$$J_b(\theta) = \frac{1}{2k} \sum_{k=1}^{r} \left(y^{(i_k)} - h_\theta(x^{(i_k)})\right)^2$$

(a) **(4 points)** Sample 1 million data points taking values of $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$, $x_1 \sim \mathcal{N}(3, 4)$ and $x_2 \sim \mathcal{N}(-1, 4)$ independently, and noise variance in y, $\sigma^2 = 2$.

(b) **(6 points)** Implement Stochastic gradient descent method for optimizing $J(\theta)$. Relearn $\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{bmatrix}$ using sampled data points of part a) keeping everything same except the batch size. Keep $\eta = 0.001$ and initialize $\forall j\ \theta_j = 0$. Report the θ learned each time separately for values of batch size $r = \{1, 100, 10000, 1000000\}$. Carefully decide your convergence criteria in each case. Make sure to watch the online video posted on the course website for deciding the convergence of SGD algorithm.

(c) **(6 points)** Do different algorithms in the part above (for varying values of r) converge to the same parameter values? How much different are these from the parameters of the original hypothesis from which the data was generated? Comment on the relative speed of convergence and also on number of iterations in each case. Next, for each of learned models above, report the error on a new test data of 10,000 samples provided in the file named q2test.csv. Note that this test set was generated using the same sampling procedure as described in part (a) above. Also, compute the test error with respect to the prediction of the original hypothesis, and compare with the error obtained using learned hypothesis in each case. Comment.

(d) **(4 points)** In the 3 dimensional parameter space(θ_j on each axis), plot the movement of θ as the parameters are updated (until convergence) for varying batch sizes. How does the (shape of) movement compare in each case? Does it make intuitive sense? Argue.

3. **(15 points) Logistic Regression**
Consider the log-likelihood function for logistic regression:

$$L(\theta) = \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$$

For the following, you will need to calculate the value of the Hessian H of the above function.

(a) **(10 points)** The files logisticX.csv and logisticY.csv contain the inputs ($x^{(i)} \in R^2$) and outputs ($y^{(i)} \in \{0, 1\}$) respectively for a binary classification problem, with one training example per row. Implement[1] Newton's method for optimizing $L(\theta)$, and apply it to fit a logistic regression model to the data. Initialize Newton's method with $\theta = \vec{0}$ (the vector of all zeros). What are the coefficients θ resulting from your fit? (Remember to include the intercept term.)

(b) **(5 points)** Plot the training data (your axes should be x_1 and x_2 , corresponding to the two coordinates of the inputs, and you should use a different symbol for each point plotted to indicate whether that example had label 1 or 0). Also plot on the same figure the decision boundary fit by logistic regression. (i.e., this should be a straight line showing the boundary separating the region where $h(x) > 0.5$ from where $h(x) \leq 0.5$.)

4. **(25 points) Gaussian Discrmimant Analysis**
In this problem, we will implement GDA for separating out salmons from Alaska and Canada. Each salmon is represented by two attributes x_1 and x_2 depicting growth ring diameters in 1) fresh water, 2) marine water, respectively. File q4x.dat stores the two attribute values with one entry on each row. File q4y.dat contains the target values ($y^{(i)}$'s $\in$ {Alaska, Canada}) on respective rows.

(a) **(6 points)** Implement Gaussian Discriminant Analysis using the closed form equations described in class. Assume that both the classes have the same co-variance matrix i.e. $\Sigma_0 = \Sigma_1 = \Sigma$. Report the values of the means, μ_0 and μ_1, and the co-variance matrix Σ.

(b) **(2 points)** Plot the training data corresponding to the two coordinates of the input features, and you should use a different symbol for each point plotted to indicate whether that example had label Canada or Alaska.

(c) **(3 points)** Describe the equation of the boundary separating the two regions in terms of the parameters μ_0, μ_1 and Σ. Recall that GDA results in a linear separator when the two classes have identical co-variance matrix. Along with the data points plotted in the part above, plot (on the same figure) decision boundary fit by GDA.

(d) **(6 points)** In general, GDA allows each of the target classes to have its own covariance matrix. This results (in general) results in a quadratic boundary separating the two class regions. In this case, the maximum-likelihood estimate of the co-variance matrix Σ_0 can be derived using the equation:

$$\Sigma_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}(x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} \quad (1)$$

And similarly, for Σ_1. The expressions for the means remain the same as before. Implement GDA for the above problem in this more general setting. Report the values of the parameter estimates i.e. μ_0, μ_1, Σ_0, Σ_1.

(e) **(5 points)** Describe the equation for the quadratic boundary separating the two regions in terms of the parameters μ_0, μ_1 and Σ_0, Σ_1. On the graph plotted earlier displaying the data points and the linear separating boundary, also plot the quadratic boundary obtained in the previous step.

(f) **(3 points)** Carefully analyze the linear as well as the quadratic boundaries obtained. Comment on your observations.

[1] Write your own version, and do not call a built-in library function.