Assignment 3

CS6510: Applied Machine Learning
IIT-Hyderabad
Jan-Apr 2019

Max Marks: 40
Due: 14 Apr 2019 11:59 pm

This homework is intended to cover programming exercises in the following topics:

- Regression Methods, Logistic Regression

Instructions

- Please use Google Classroom to upload your submission by the deadline mentioned above. Your submission should comprise of a single file (PDF/ZIP), named `<Your_Roll_No>_Assign3`, with all your solutions.
- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 7 grace days for late submission of assignments. Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the CS6510 Marks and Grace Days document.
- You have to use PYTHON for the programming questions.
- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

Questions: Theory

(No programming required)

1. **Logistic Regression:** ***(10 marks)***

   (a) Plot the sigmoid function $1/(1+e^{-w\mathbf{x}})$ vs $\mathbf{x} \in \mathbb{R}$ or increasing weight $w \in \{1, 5, 100\}$. A qualitative sketch is enough. Use these plots to argue why a solution with large weights can cause logistic regression to overfit. [*2 marks*]

(b) To prevent overfitting, we want the weights to be small. To achieve this, instead of maximum likelihood estimation MLE for logistic regression:

$$\max_{w_0,\cdots,w_d} \prod_{i=1}^{n} P(Y_i|X_i, w_0, \cdots, w_d)$$

we can consider maximum a posterior (MAP) estimation:

$$\max_{w_0,\cdots,w_d} \prod_{i=1}^{n} P(Y_i|X_i, w_0, \cdots, w_d)P(w_0, \cdots, w_d)$$

where $P(w_0, \cdots, w_d)$ is a prior on the weights. Assuming a standard Gaussian prior $\mathcal{N}(0, I)$ for the weight vector (I = Identity matrix), derive the gradient ascent update rules for the weights. [*3 marks*]

(c) One way to extend logistic regression to multi-class (say, K class labels) setting is to consider $(K-1)$ sets of weight vectors and define:

$$P(Y = y_k|X) \propto \exp(w_{k0} + \sum_{i=1}^{d} w_{ki}X_i) \text{ for } k = 1, \cdots, K-1$$

What model does this imply for $P(Y = y_k|X)$? What would be the classification rule in this case? [*3 marks*]

(d) Draw a set of training data with three labels and the decision boundary resulting from a multi-class logistic regression. (The boundary does not need to be quantitatively correct but should qualitatively depict how a typical boundary from multi-class logistic regression would look like.) [*2 marks*]

2. **Kernel Regression and Variants:** ***(7 marks)*** Given a set of n examples, $(\mathbf{x}_i, y_i), i = 1, \cdots, n$, a *linear smoother* is defined as follows. For any $\mathbf{x}$, there exists a vector $l(\mathbf{x}) = (l_1(\mathbf{x}), \cdots, l_n(\mathbf{x}))^T$ such that the estimated output $\hat{y}$ of $\mathbf{x}$ is $\hat{y} = \sum_{i=1}^{n} l_i(\mathbf{x})y_i = l(\mathbf{x})^T Y$ where Y is a $n \times 1$ vector, $Y_i = y_i$. This means that the prediction is a linear function of the training responses (y_is) and it varies slowly and smoothly with change or noise in y_is.

As an example, for linear regression, we assume the data are generated from the model $y_i = \sum_{j=1}^{m} w_j h_j(x_i) + \epsilon_i$, where h are basis functions (such as polynomials: x, x^2, x^3, $\cdots$, etc). The least squares estimate for the coefficient vector $\mathbf{w}$, as we know, is given by $\mathbf{w} = (H^T H)^{-1} H^T Y$ where H is a $n \times m$ matrix, $H_{ij} = h_j(\mathbf{x}_i)$. Given an input x, note that $\hat{y} = h(\mathbf{x})^T \mathbf{w} = h(\mathbf{x})^T (H^T H)^{-1} H^T Y = (H(H^T H)^{-1} h(\mathbf{x}))^T Y$. If $l(\mathbf{x}) = H(H^T H)^{-1} h(\mathbf{x})$, then $\hat{y} = l(\mathbf{x})^T Y$. Therefore, linear regression is a linear smoother.

Now, answer the following questions:

(a) In kernel regression using the kernel $K(\mathbf{x}_i, \mathbf{x}) = \exp \frac{||\mathbf{x}_i - \mathbf{x}||_2^2}{\sigma^2}$, given an input $\mathbf{x}$, what is the estimated output $\hat{y}$? Is kernel regression a linear smoother? [*2.5 marks*]

(b) Suppose we fit a linear regression model, but instead of sum of residual squares $||Hw - Y||_2^2$, we minimized the sum of absolute values of residuals: $||Hw - Y||_1$. Prove that this is not a linear smoother (give a counter-example). (*Hint:* Think about the median for a set of real numbers $(y_1, \cdots, y_n)$ where n is odd, the median y_M minimizes the sum of absolute differences $M = \arg\min_j \sum_{i=1}^{n} |y_j - y_i|$.) [*2.5 marks*]

(c) If we divide the range (a, b) (a and b are real numbers, and $a < b$) into m equally spaced bins denoted by $B_1, \cdots, B_k$. Define the estimated output $\hat{y} = \frac{1}{|B_k|} \sum_{i:\mathbf{x}_i \in B_k} y_i$ for $\mathbf{x} \in B_k$, where $|B_k|$ is the number of points in B_k. In other words, the estimate $\hat{y}$ is a step function obtained by averaging the y_is over each bin. This estimate is called the regressogram. Is this estimate a linear smoother? If yes, give the vector $l(\mathbf{x})$ for a given input $\mathbf{x}$; otherwise, state your reasons. [*2 marks*]

Questions: Programming

3. **Linear Regression:** ***(13 marks)*** We will now implement Linear Regression to predict the age of Abalone (a type of snail). The data set is made available as part of the provided zip archive (linregdata). You can read more about the dataset at the UCI repository link. We are interested in predicting the last column of the data that corresponds to the age of the abalone using all the other attributes.

   (a) The first column in the data denotes the attribute that encodes-female, infant and male as 0, 1 and 2 respectively. The numbers used to represent these values are symbols and therefore should not be ordinal. Transform this attribute into a three column binary representation. For example, represent female as (1, 0, 0), infant as (0, 1, 0) and male as (0, 0, 1). [*0.5 marks*]

   (b) Before performing linear regression, we must first standardize the independent variables, which includes everything except the last attribute (target attribute). Standardizing means subtracting each attribute by its mean and dividing by its standard deviation. Standardization will transform the attributes to possess zero mean and unit standard deviation. You can use this fact to verify the correctness of your code. [*0.5 marks*]

   (c) Implement the following functions: (i) `mylinridgereg(X, Y, λ)` that calculates the linear least squares solution with the ridge regression penalty parameter (λ) and returns the regression weights; (ii) `mylinridgeregeval(X, weights)` that returns a prediction of the target variable given the input variables and regression weights; and (iii) `meansquarederr(T, Tdash)` that computes the mean squared error between the predicted and actual target values. [*2 + 1 + 1 = 4 marks*]

   (d) Partition the dataset into 80% training and 20% testing (Let's call this the partition fraction, in this case 0.2). Now, use your `mylinridgereg` with different λ values to fit the penalized linear model to the training data and predict the target variable for both training and testing data. [*1 mark*]

   (e) Identify the λ with the best performance and examine the weights of the ridge regression model. Which are the most significant attributes? Try removing two or three of the least significant attributes and observe how the mean squared errors change. [*1 mark*]

   (f) We now would like to ask the question: Does the effect of λ on error change for different partitions of the data into training and test sets? To do this, change the partition fraction (a value between 0 and 1, as defined earlier) with at least 4 other values. Repeat the following steps 25 times for each partition fraction:

   - Randomly divide data into training and test sets.
   - Standardize the training input variables.

- Standardize the testing input variables using the means and standard deviations from the training set.
- Follow step (d) for each such partition.

For each partition fraction, plot a figure with λ on the x-axis, and MSE on the y-axis. For each figure, include 2 graphs - one for the training MSE and one for the test MSE. (You should then have 5 figures in total, with 2 plots on each figure.) [*3 marks*]

(g) Do the above figures give you clarity? Also, plot two more figures. In the first graph, plot the minimum average mean squared testing error versus the partition fraction values. In the second graph, plot the λ value that produced the minimum average mean squared testing error versus the partition fraction. [*1 mark*]

(h) How good is your model? So far, we have been looking at only the mean squared error. We might also be interested in understanding the contribution of each prediction towards the error. Maybe the error is due to a few samples with large errors and others have tiny errors. One way to visualize this information is to a plot of predicted versus actual values. Use the best choice for the training fraction and λ, make two graphs corresponding to the training and testing set. The X and Y axes in these graphs will correspond to the predicted and actual target values respectively. If the model is good, then all the points will be close to a 45-degree line through the plot. [*2 marks*]

Include all the plots and your observations in your submission.

4. **Kaggle - Taxi Fare Prediction:** ***(10 marks)*** The next task of this assignment is to work on a (completed) Kaggle challenge on taxi fare prediction. As part of this task, please visit https://www.kaggle.com/c/new-york-city-taxi-fare-prediction to know more about this problem, and download the data. (You now know how to download data from Kaggle.)

   You are allowed to use any machine learning library of your choice: scikitlearn, pandas, Weka (we recommend `scikitlearn`), and any regression method too. Use `train.csv` to train your classifier. Predict the cuisine on the data in `test.csv`, and report your best 2 scores in your report. (We will also upload your codes randomly to confirm the scores.)
   Deliverables:

   - Code
   - Brief report (PDF) with top-2 scores of your methods, and a brief description of the methods that resulted in the top 2 scores.
   - Your report should also include your analysis of why your best 2 methods performed better than others you tried.