Student Name: Minh Phan
Course: AMATH 352
Email address: minhphan@uw.edu

Homework 7

---

1 Introduction and overview of the problem

Politics has always been a melting pot of ideologies, power and fame. Confronted by the vast array of political forms, political scientists have tempted to try to classify, developing models and to bring an "analytic order" to the wide range of data we have on these politicians, varying from classification based on number of rulers (one or many) to by key institutions. Especially in the modern United States' political scene where a politician's party matters more than anything, We want to draw a line and cluster politicians into these boxes of party labels (Democratic, Republic, or Independent) in order to understand more thoroughly their beliefs and actions, as a reflection of their parties themselves. This classification will again assist people in voting decisions, whether they are for their local committees or state-wide senators and U.S. presidents [2].

2 Theoretical background and description of algorithm

Computationally, classification is the "process of predicting the class given data points" . In traditional classification problems, one's task is to develop a model or an approximated mapping function (f) which categorize input variables (which can be represented as a vector $\underline{x}$ denoting a set of values pointing to different attributes) into discrete output variables (classes) [1]. In the scope of this report, I will define some of the terms used in here to help comprehend the content.

- Mean squared error

  Mean squared error (MSE) indicates how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (called "errors" or "residuals") and squaring them to remove negative signs, which gives more weight to larger differences. The lower the MSE, the better the regression line [3].

  $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

  in which y_i is real/observed value (data point) and $\hat{y}_i$ is predicted value from regression/function.

- QR Decomposition

  According to lecture 17, computing QR factorization (or decomposition) on a matrix A is analogous to performing Gram-Schmidt process on the columns of A. We create an orthonormal matrix Q from new orthonormal vectors obtained from the Gram-Schmidt process on the columns. The other matrix R includes entries that when multiplies with Q will reproduce the original matrix A.

Here are the main steps of the algorithm I implemented in this report:

1. Split data into training and test sets

   I split the dataset into a training set and a testing set. Normally the number of records in the training set would be bigger than testing sets' to capture to the best ability abnormalities and outliers. I make sure that the labels on the y-set are discrete.

2. Depending on the number of features I put into the linear function of the classifier model, I construct a matrix that represents the set of linear equations $f(\underline{x}^{(i)}) = a_0 + a_1x_1^{(i)} + a_2x_2^{(i)} + ... + a_Nx_N^{(i)} = y$, where y is the discrete label, $x_j^{(i)}$ indicates the value of the j^{th} feature of the i^{th} data point in the set, and $a_0, a_1, a_2, ..., a_N$ are the weights (coefficients) that we will calculate later. The set of equation in matrix-vector form would be like this:

$$\begin{bmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_N^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_N^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(M)} & x_2^{(M)} & \cdots & x_N^{(M)} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(M)} \end{bmatrix}$$

Call the first matrix A.

3. Perform QR factorization on matrix A to solve the equation for the vector of coefficients $[a_0, a_1, ..., a_N]$
4. Fit the calculated coefficients into the model function. Plug in the data needed from x_train and x_test and produce predicted labels for the training and test dataset
5. Check the prediction quality of the classifier model itself on the original training set and the test set by computing the mean squared errors between the predicted and original labels on both

3 Computational Results

N	16
training MSE	0.1333
test MSE	0.2667

Table 1: Training and test mean squared errors for linear functions using 16 features of the set

Table 1 above reports the training and test mean square errors of the classifier trained on the provided training data for all 16 features (columns) of the dataset.

N	2	3	4
training MSE	1.2400	0.3733	0.1333
test MSE	1.2741	0.8000	0.2667

Table 2: Training and test mean squared errors for linear functions using N first features of the set

Table 2 above reports the training and test mean square errors of the classifier trained on the provided training data for 2, 3, 4 first features (columns) of the dataset.

4 Summary and Conclusions

We can see that the number of features is inversely proportional to the mean squared error on both training and test sets; in other words, the more features involved, the less error (and smaller the residuals) there are between the predicted labels and observed labels. However, this observation is only true to only a certain point, which in this case until $N = 4$. The mean squared error outputs for $N = 4$ is exactly the same to the case of $N = 16$. One can explain that since we classify the output using the $sign()$ function of NumPy library, which lead to the round off errors and may result in the identical values output in both cases; however, this reasoning cannot deny that the use of all 16 features in the training is unnecessary, as only the first 4 features seem to be sufficient to compute the labels without using extra power and memory.

References

[1] Sidath Asiri. *Machine Learning Classifiers*. Medium, June 2018. URL: `https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623`.

[2] The Editors of Encyclopaedia Britannica. *Political system - Issues of classification*. Encyclopedia Britannica. URL: `https://www.britannica.com/topic/political-system/Issues-of-classification`.

[3] Stephanie Glen. *Mean Squared Error: Definition and Example*. Statistics How To. URL: `https://www.statisticshowto.com/probability-and-statistics/statistics-definitions/mean-squared-error/`.