CPSC429/529: Machine Learning

Program 5: Multinomial Logistic Regression

1 Program Description

In this programming assignment, you can **form a group of two** to apply multinomial logistic regression model on dataset of customers of a large national retail chain `Table7_11.txt` (Textbook Page 374). Particularly, you need to do the following:

1. Run the example source code (`MLR_Example.py`) on digit recognition `ex3data1.mat`.
2. Modify the example code, and apply on your problem. Specifically,
   (a) Normalize the descriptive features into the range of [-1, 1].
   (b) Print out final weights (Your output values should be similar to the following, not necessary the format).

One-versus-all	θ_0	θ_1	θ_2
Model1	-3.7699179	-4.97313346	2.86557192
Model2	-2.9310945	5.81833501	2.39159111
Model3	-2.34302305	-0.31350085	-7.23082743

   Important: To obtains these final weights, you must use **learning rate = 0.0001**, when invoking the `one_vs_all` method in `main()`. The `MLR_Example.py` uses the learning rate = 1.
   (c) Plot your decision boundaries and normalized descriptive features, similar to program screenshot in Figure 1. Make sure you also need to provide the lengends as I did the plot.
   (d) Use the trained model to predict all the original training set. Print out the prediction accuracy using the `predict_all` method, your prediction accuracy should be 100%.
   (e) Copy the `predict_all` method, and change the name to be `predict_oneinstance`. You need to modify some of the code, and print out the following things:
      i. The predictions of individual models (The values should be [0,1], See an example in lecture slide 55 in `6B_LogisticRegression.pdf`).
      ii. The normalized predictions of individula models (The sum of the predicted should be 1.0. See an example in lecture slide 56 in `6B_LogisticRegression.pdf`).
      iii. The final predicted model class (Should be 1, 2 or 3).

For the testing purpose, you need to use the instance x =(1, 0.10790978, 0.7643608) (*i.e.*, x_0= 1, x_1=0.10790978, x_2= 0.7643608).

This instance is one of two instances staying within the triangle (Figure 1). This instance is interesting because all three models have low prediction values. From the decision boundaries (i.e., the training models), all predicted values using these three models should be **less than 0.5**. Can you figure it out why that is the case?

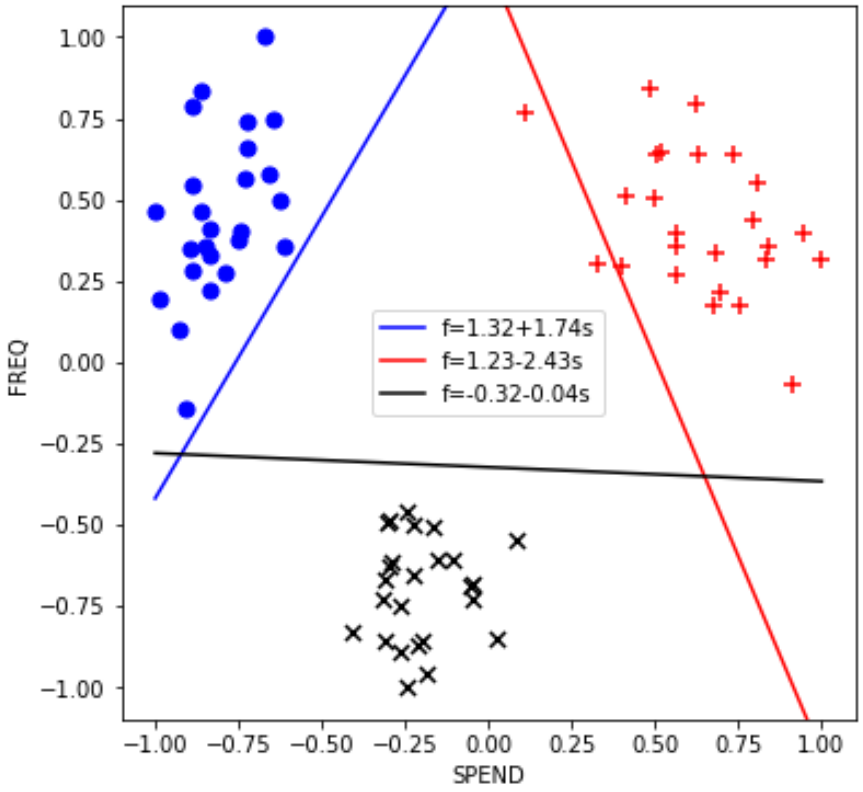

Figure 1: A sample program output on "`Table7_11.txt`"

You should not use `scikit-learner` for this program (other than preprocessing: range normalization). You can use the majority of source code from the example code.

2 Suggested Reading

1. Understand the multinomial logistic regression (MLR) model through watching the videos on the topics of logistic regression. Most importantly, you need to watch the following two videos:

(a) Lecture 6.4 Logistic Regression — Cost Function [Machine Learning — Andrew Ng]

(b) Lecture 6.7 Logistic Regression — MultiClass Classification OneVsAll [Andrew Ng]

You may also read the lecture notes in section of MLR in `6B_LogisticRegression.pdf` (Slides 44-56), and the textbook.

2. Read the explanation of the example code (`MLR_Example.py`) at the following link:

   `https://github.com/jdwittenauer/ipython-notebooks/blob/master/notebooks/ml/ML-Exercise3.ipynb`.

   Note: Pay particular attention to the following methods:

   - `cost` method: This method is to measure the cost function described in Dr. Andrew Ng's lecture notes on Logistic regression model, not same one we used for your previous program.
   - `one_vs_all` method: This method is the key to this model. You will feed the original data and return all the parameters for each of the multi-class models. Again the details should be found in the videos, lecture notes, and the textbook.
   - `predict_vs_all` method: This method is to predict all training set using the parameters (after the models have been trained).

3 Submission

Upload the following items on D2L dropbox, including:

1. The source code (including data file).
2. Program output file (saved as .txt). Your outputs should have the following three things:
   (a) Final weights of the three models (See the above table).
   (b) Prediction accuracy of all training set.
   (c) The predicted values (raw predictions, normalized values, and predicted class) of the given instance.
3. Model plot saved as a .png files. The plots should be similar to the one shown in Figure 1. Label the boundary decisions (equations) for each of the models.