

Project 1 Report

Classification using Linear Regression

Arun Balchandran

Student ID: 1001402679

University of Texas at Arlington

Table of Contents

Problem	3
Introduction.....	4
Data	5
Method	7
Results.....	8
Bibliography	10

Problem

We are given a dataset containing the lengths and widths of the sepals and petals of the Iris plant. We have 150 such readings for 150 different flowers and a classification given indicating the species of the Iris flower for which the readings were taken.

Our problem is that we need to create a model using linear regression that can be used to classify the Iris flowers according to species given their petal and sepal lengths and widths.

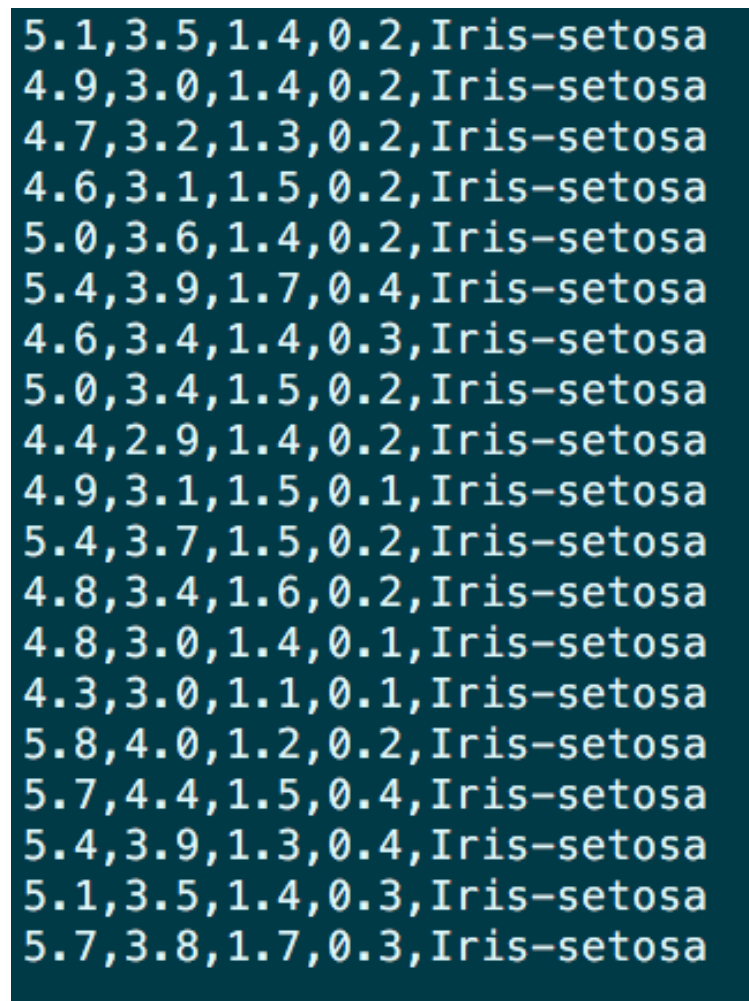
Introduction

The Iris dataset is a data set introduced by British statistician and biologist **Ronald Fisher** [1]. The dataset contains the readings of the lengths and the widths of sepals and petals of the 3 different species of the Iris flower, namely, *Iris virginica*, *Iris setosa* and *Iris versicolor*.

Simple linear classification is used on this dataset to classify it into different classes and cross validation is done on the dataset to find the accuracy of the method being used. Conclusions are then made on the dataset to assess the usefulness of the linear classification method.

Data

The data being used in this assignment is the Iris Flower dataset. This dataset has 4 features, namely the lengths and widths of the sepals and petals of the Iris flowers and their species (classification). There are 150 items rows in this dataset, each measuring the sizes of the features for that given flower. Below is a screenshot containing the first few terms of the dataset.



```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
5.4,3.7,1.5,0.2,Iris-setosa
4.8,3.4,1.6,0.2,Iris-setosa
4.8,3.0,1.4,0.1,Iris-setosa
4.3,3.0,1.1,0.1,Iris-setosa
5.8,4.0,1.2,0.2,Iris-setosa
5.7,4.4,1.5,0.4,Iris-setosa
5.4,3.9,1.3,0.4,Iris-setosa
5.1,3.5,1.4,0.3,Iris-setosa
5.7,3.8,1.7,0.3,Iris-setosa
```

Figure 1. The Iris dataset.

We can plot this data using the Matplotlib library in Python. We assign each axis one of the 4 features and assign the fourth feature to the color variable. The shape of the data points tells us its classification according to its species.

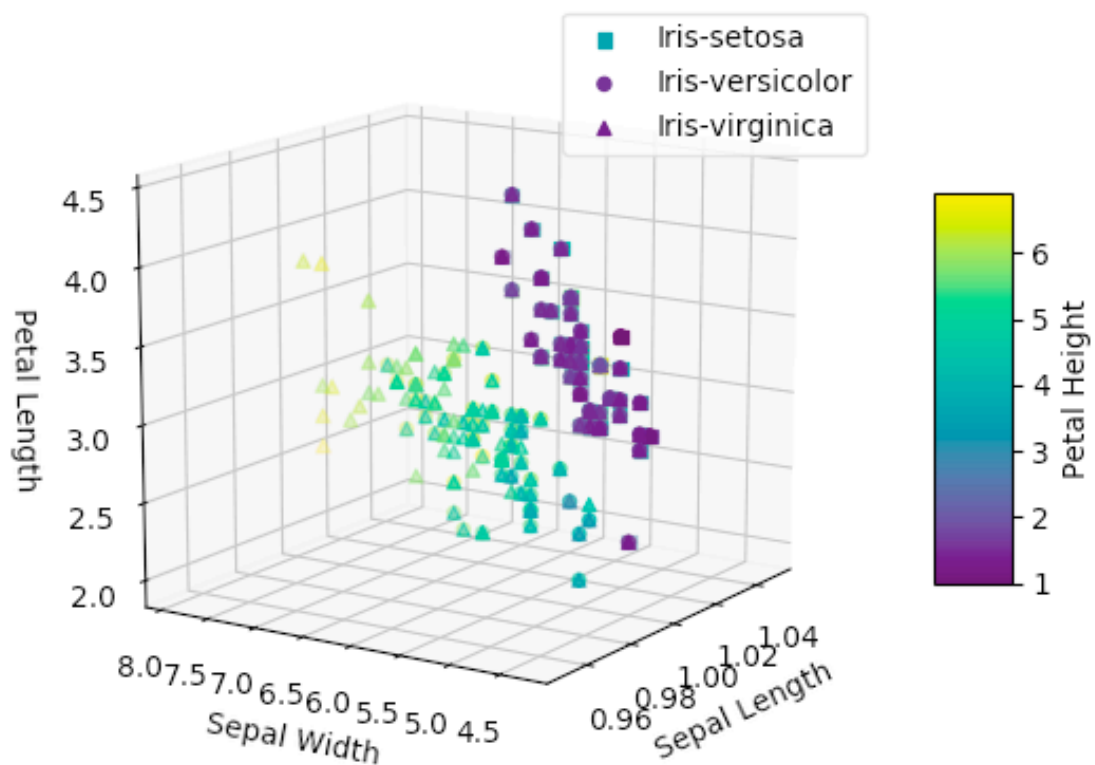


Figure 2. Iris dataset plotted in the 3D plane

Method

The simple linear classifier is used to classify the data into various classes. The classifier is written in Python3. The 'numpy' library was used to provide convenience in terms of matrix operations and the 'Matplotlib' library was used to generate the plot for visualizing the input.

The matrix X contains the training data that is used for training the classifier. This data is the list of feature data for each flower. The vector Y contains the classification data for each flower.

The data is first cleaned using the 'clean_datafile' method and '1s' are appended to each input data row so that we can get a linear classifier with '5' variables after training.

Finally, the 'cross_validation' method is called which calls the 'train_linear_classifier' method to train the training data subset and the testing is done on the test data subset by calling the 'test_linear_classifier' method. The linear classifier finds the β values for the input matrix X by using the formula [1]:

$$\hat{\beta} = (X'X)^{-1}X'y$$

Figure 3. Formula for Beta.

This test method then returns the accuracy for that round of cross validation. The results are reported in the last section.

Results

The cross validation function is called and the β values for each iteration of the validation are displayed along with their accuracy values. The accuracy is calculated using the formula [1]

$$ACC = \frac{TP + TN}{P + N} = \frac{TP + TN}{TP + TN + FP + FN}$$

Figure 4. Formula for Accuracy

The average of all the accuracy values is taken. There is an option for using shuffling or not shuffling, and the average accuracy with shuffle is 0.96666 while the accuracy without shuffle is 0.96000.

The cross validation was performed using a 'k' value of 15 because it is generally seen that higher sizes in the training set result in better performance of the linear classifier. It also makes sense because the size of the dataset is really small and therefore that would impact the performance of linear classifier if we use small 'k' values.


```

[Aruns-MacBook-Air:Project1 arun$ python3 linear_classifier.py
Enter the name of the file to be tested : iris.data.txt
Do you want to shuffle the data? Enter y or n : n
Shuffling not enabled for cross validation
{'Iris-setosa': 1, 'Iris-versicolor': 2, 'Iris-virginica': 3}
num classes 3
[ 1.18717228 -0.11116637 -0.04298249  0.22805741  0.61275981]
Accuracy values for fold 0 are 1.0
[ 1.24867947 -0.12404395 -0.05034347  0.23957077  0.6035162 ]
Accuracy values for fold 1 are 1.0
[ 1.20107874 -0.11153176 -0.04392325  0.22639934  0.61274029]
Accuracy values for fold 2 are 1.0
[ 1.23303844 -0.1146751  -0.05580779  0.22770768  0.62175632]
Accuracy values for fold 3 are 1.0
[ 1.21251193 -0.10937086 -0.04745999  0.21931825  0.62469159]
Accuracy values for fold 4 are 1.0
[ 1.13140073 -0.09283611 -0.05021312  0.23366513  0.58640979]
Accuracy values for fold 5 are 1.0
[ 1.2365465  -0.11078551 -0.05538526  0.22896778  0.60920135]
Accuracy values for fold 6 are 1.0
[ 1.1591208  -0.0967894  -0.05416407  0.23489953  0.59070308]
Accuracy values for fold 7 are 0.8
[ 1.27812501 -0.15042758 -0.02266164  0.26715408  0.56902918]
Accuracy values for fold 8 are 0.9
[ 1.23122697 -0.13506325 -0.02904727  0.26220663  0.55905667]
Accuracy values for fold 9 are 1.0
[ 1.09071433 -0.08338875 -0.04127081  0.19411883  0.65471049]
Accuracy values for fold 10 are 1.0
[ 1.10850624 -0.1018008  -0.03020118  0.22816355  0.59215665]
Accuracy values for fold 11 are 0.9
[ 1.23936223 -0.1136202  -0.04741273  0.21529326  0.6188132 ]
Accuracy values for fold 12 are 1.0
[ 1.24638834 -0.07935734 -0.08042175  0.15357006  0.72417627]
Accuracy values for fold 13 are 0.8
[ 1.14477394 -0.1121241  -0.03152202  0.24731126  0.55666758]
Accuracy values for fold 14 are 1.0
0.9600000000000001

```

Figure 5. Output of the program

Bibliography

- [1] https://en.wikipedia.org/wiki/Iris_flower_data_set
- [2] https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NY_U_notes.pdf
- [3] https://en.wikipedia.org/wiki/Confusion_matrix