CSE474/574 Introduction to Machine Learning
Programming Assignment 1
Report

Group number:
CSE574 Programming Assignment 14
Group Members:
Vinita Venktesh Chappar
Prajit Krisshna Kumar
Harsh Hemendra Shah

Part I - Linear Regression

Problem 1: Linear Regression with Direct Minimization

Here, we were asked to implement the least squared method to estimate the regression parameters. These parameters will be used to calculate the Root mean squared error i.e RMSE. We got the following RMSE values with and without respect to intercept..

OUTPUT:-

```
RMSE without intercept on train data - 138.20
RMSE with intercept on train data - 46.77
RMSE without intercept on test data - 326.76
RMSE with intercept on test data - 60.89
```

As clearly visible in the above results, when we calculate the RMSE with itercept, we have minimum squared loss value.

Problem 2: Using Gradient Descent for Linear Regression Learning

Gradient descent can be used to calculate the weights 'w' in a better and efficient way. Then we can use these weights to minimize the loss function. These were the values we recieved from this method...

OUTPUT:-

```
Gradient Descent Linear Regression RMSE on train data - 47.99
Gradient Descent Linear Regression RMSE on test data - 54.96
```

As we can see, we got the RMSE value as 54.96 on the test data. In the ordinary linear regression, we got the RMSE value for the same test data as 60.86
Hence, we were able to minimize the loss even further using the gradient descent method.

Part II - Linear Classifiers

Problem 3: Using Gradient Descent for Perceptron Learning
Perceptron is similar to the squared loss function. We trained our model to predict the accuracy of this function. We got the following results-

OUTPUT:-

```
Perceptron Accuracy on train data - 0.84
Perceptron Accuracy on test data - 0.84
```

Here, the perceptron predicted our accuracy as 84% for both training and test data.

Problem 4: Using Newton's Method for Logistic Regression Learning
In the Newton's method we compute a Hessian matrix which will give us logistic loss for the given data set. For this, we had to first calculate the logistic loss and gradient vector of this logistic loss. We got the following results..

OUTPUT:-

```
Logistic Regression Accuracy on train data - 0.84
Logistic Regression Accuracy on test data - 0.86
```

The predicted accuracy for logistic loss were 84% and 86% for training and test data respectively. This accuracy prediction is almost similar to the prediction we got using the Perceptron learning method.

Problem 5: Using Stochastic Gradient Descent Method for Training Linear Support Vector Machine
The SVM model is used to learn the weights of the training data using the hinge loss function. Here we trained the model to predict the weights using 200 iterations of the hinge loss function.

OUTPUT:-

```
SVM Accuracy on train data - 0.85
SVM Accuracy on test data - 0.87
```

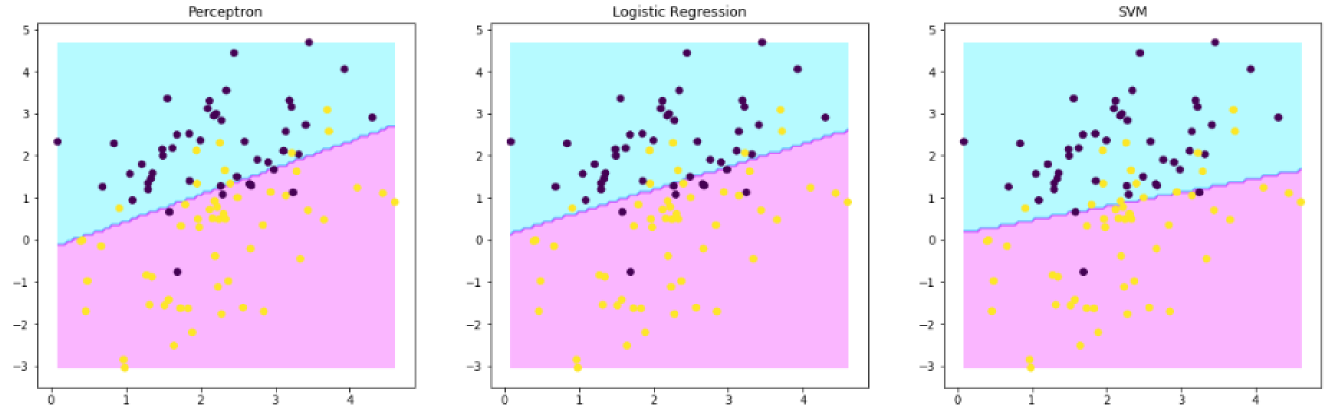
We got the prediction of the accuracy as 85% and 87% on the training and test data respectively, which is more accurate than the what we got using Perceptron and Newton's method. This shows that SVM can give better results and more accuracy for minimizing the loss function.

Problem 6: Comparing Linear Classifiers

Here, we compared the results of the 3 classifiers we implemented above. This is the graph we got..

OUTPUT:-

```
Text(0.5, 1.0, 'SVM')
```



As shown in the figure above, the Perceptron and Logistic Regression(Newton's method) gave us similar results with minor differences. Whereas, the SVM gave us more accuracy for minimizing the loss function. Although we cannot say SVM will give us the most accurate results, we can definitely say that SVM has a better approach for a learning model than other linear classifiers.