Linear Methods for Supervised Learning

CSE 574: Introduction to Machine Learning - PA 1

March 6, 2020

Group # 4

Anjali Bachani
Abhishek Mishra
Adithya L. Narayanan

Contents

List of Figures

Problem 1: Linear Regression with Direct Minimization

In this problem, we implement linear regression with direct minimization, using the closed form equation. Root mean squared error (RMSE) used to assess the predictive accuracy of the model.

We compare the model performance across two cases: with an intercept (or bias) term and without an intercept term added to the predictors.

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

We observe that the error is higher for training and test data without an intercept as compared to that calculated with intercept.

Problem 2: Using Gradient Descent for Linear Regression Learning

In this problem, Gradient Descent, using `scipy.optimize.minimize` is employed to minimize the linear regression error function. Similar to Problem 1, RMSE is used to measure the predictive accuracy of this model.

Gradient Descent Linear Regression RMSE on train data - 48.08
Gradient Descent Linear Regression RMSE on test data - 54.80

Based on the model's predictive accuracy, it can be inferred that Gradient Descent performs better in this scenario.

Problem 3: Using Gradient Descent for Perceptron Learning

For the scope of this problem, perceptron learning functions as a two class classification method that relies on linear regression's error function, which is minimized using gradient descent.

Perceptron Accuracy on train data - 84 %
Perceptron Accuracy on test data - 84 %

Problem 4: Using Newton's Method for Logistic Regression Learning

The functions `logisticObjVal`, `logisticGradient` and `logisticHessian` were implemented to compute the logistic loss, gradient vector of logistic loss and the Hessian matrix of logistic loss respectively, for the given data set. These functions were then used to train the logistic regression model by calling the `scipy.optimize.minimize` method. The function `evaluateLinearModel` from Problem 3 was used to calculate the accuracy of the training and test data. The accuracies obtained are reported as follows:

Logistic Regression Accuracy on train data - 84 %
Logistic Regression Accuracy on test data - 86 %

Problem 5: Using Stochastic Gradient Descent Method for Training Linear Support Vector Machine

Using stochastic gradient descent, a linear Support Vector Machine (SVM) is trained to performed binary classification. The accuracy measures for this implementation were a range, due to the stochastic nature of the algorithm which involves random sampling of obervations to update the weights of predictors.
SVM Accuracy Range on train data: 65-87 %
SVM Accuracy Range on test data: 81-88 %

Problem 6: Comparing Linear Classifiers

Comparing the results obtained from Problems 3, 4 and 5, we can say that logistic regression and SVM classifier give better accuracies in general, however, the results can be a bit sporadic with SVMs due to the stochastic implementation carried out in this study. Even though SVM classifier may sometimes report test accuracy greater than that of logistic regression, the logistic regression classifier yields higher average accuracy over training and test data and should therefore be preferred to build more robust models. The entropy in the performance can be attributed to the random sampling carried out as a part of the implementation that may fail to capture the true nature of the data it is being tested on at all times (a limited number of observations may not always capture the trend of the data).

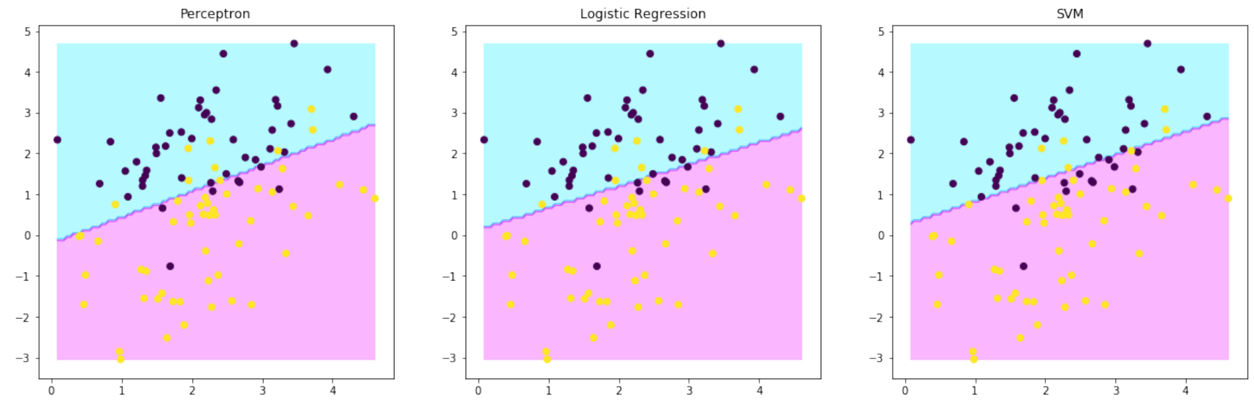

Figure 1: Decision boundaries learnt by each of the classifiers

Observing the decision boundaries in Figure 1, we see that the results from the three classifiers are almost identical. But upon looking more closely, we can see that in this particular case, the SVM classifier has less number of data points on the wrong side of decision boundary. The logistic regression classifier and the perceptron classifier have somewhat equal number of data points on the wrong side of the decision boundary.Iterative runs yield differing decsion boundaries for SVMs, which can again be attributed to the stochastic nature of the process, as described above, wherein each run of training may not necessarily converge with the same decision boundary being learnt.