Akshay Manjunath

Homework 6

EE 559
March 6, 2020

Problem 1

Part(a)

Here I am employing Python and scikit-learn.

Part (b)

Feature	Mean	Standard Deviation
1	1.29653933e+01	8.19560112e-01
2	2.27000000e+00	1.10306417e+00
3	2.37629213e+00	2.73004021e-01
4	1.96494382e+01	3.46517583e+00
5	9.89101124e+01	1.15589748e+01
6	2.27235955e+00	6.14548382e-01
7	2.02943820e+00	9.19718276e-01
8	3.60674157e-01	1.20241309e-01
9	1.57617978e+00	5.41470129e-01
10	5.09123596e+00	2.40437795e+00
11	9.53213483e-01	2.29907577e-01
12	2.56033708e+00	7.23461482e-01
13	7.29707865e+02	3.07221225e+02

The normalizing factor should be calculated from the training data only.This is because in real world problems, the actual test data is not available while we are developing the classifier model. Furthermore, the purpose of labeled test dataset is to estimate performance on actual unknown inputs, which is usually not known beforehand.

So in order to get a good estimate of the model quality, we need to restrict the calculation of the normalization parameters to the training data set only and not peek into the test dataset.

Part (c)

Perceptron Classifier

1. If no initial weight vector is specified, by default weight and intercept are initialized as zeros.

2. In the parameters for the perceptron classifier, we mention tolerance by the parameter tol. This tol is related to the stopping criterion for the perceptron algorithm. The default value is none. If it's not default, the iterations will stop when loss >previous loss - tol where previous loss is from the previous epoch. It defaults to 1e-3 from version 0.21.

   The backup halting condition is specified by max_iter which is the maximum number of epochs. It is a parameter which tells the number of passes over the training data. It's default value is 5. And if tol is not none and the halting condition is not met, then it's default value is 1000. It's default value is 1000 from version 0.21.

Part (d)

For first 2 features, the final weight vectors are :

[[2.56330986 -1.88565639]

[-2.42155116 -1.50489885]

[0.04318568 2.41146442]]

The classification accuracy on training data for first 2 features is 0.7528.

The classification accuracy on test data for first 2 features is 0.79775.

For all 13 features, the final weight vectors are :

[[2.91153722 0.11785352 0.21566131 -1.31355106 0.84179932 -0.52546123 1.86830538 0.93164714 0.66215901 0.75017651 1.34680402 2.11623048 3.27166066]

[-5.43906467 -5.49378737 -2.68218464 3.17346933 -2.9015831 0.62510506 0.87703945 1.613796 -0.04233169 -6.42685327 1.80629497 -2.45759805 -5.4252528]

[0.34452574 2.86474721 -0.48441481 -2.0787894 0.03985424 1.35735009 -2.46093923 -0.79802072 -1.59366381 5.73429637 -3.20441419 -2.78933651 -0.65995704]]

The classification accuracy on training data for all 13 features is 1.0.

The classification accuracy on test data for all 13 features is 0.94382

Part (e)

For first 2 features, the best final weight vectors are :

[[1.69085482 -1.39449524]

[-2.387973 -1.1458107]

[0.433509 -0.71199854]]

The best classification accuracy on training data for first 2 features is 0.83146.

The best classification accuracy on test data for first 2 features is 0.7528.

For all 13 features, the best final weight vectors are :

[[2.46262615 1.18452154 1.29814862 -2.9805194 2.19365343 0.82381886 3.0849811 -0.59467759 -0.15422341 0.18960265 2.28428796 3.66633205 6.04453476]

[-5.29674125 -3.56275473 -3.97195279 3.28744431 -4.99383353 0.90223279 -0.79729725 1.80751975 -0.60141963 -5.5991832 1.71100995 0.23082033 -4.55143929]

[0.53322357 1.18151331 1.37597725 -0.07210525 0.10172525 -0.18037394 -3.20616314 -0.76034103 -1.34879223 1.5692117 -1.43581099 -3.05641518 -1.47148866]]

The best classification accuracy on training data for all 13 features is 1.0.

The best classification accuracy on test data is for all 13 features is 0.95505.

Part (f)

1. Comparing 2 features to 13 features for part (d) we can see that both the training data and test data accuracy increases significantly.The training data accuracy increases from 0.7528 to 1.0 and the test data accuracy increases from 0.79775 to 0.94382.

   This is because as we increase the number of features, it helps the algorithm by giving more data for the model to work with. But it's important to choose the right features only, as sometimes using more features can lead to overfitting.

2. Similarly comparing 2 features to 13 features for part (e) we can see significant improvement in accuracy. The training data accuracy increases from 0.8314 to 1.0 and the test data accuracy increases from 0.7528 to 0.95505.

3. Comparing (d) to (e) for 2 features for each dataset we see that the test accuracy for part (e) is higher than that of part (d). Comparing (d) to (e) for 13 features for each dataset we see that the test accuracies are almost similar.

MSE (pseudo-inverse version) classification

Part (g)

Test data accuracy for first 2 features for unnormalized data is 0.7528.

Test data accuracy for all 13 features for unnormalized data is 0.9775.

Part (h)

Test data accuracy for first 2 features for Standardized data is 0.7528.

Test data accuracy for all 13 features for Standardized data is 0.9775.

Part (i)

The test accuracies for both the first 2 features and all 13 features are same for the unstandardized and standardized data.The pseudo inverse version classification is a tree based algorithm which is scale invariant. So data standardization has no effect on the performance of the MSE classifier.

Part (j)

Comparing the results of h) and e), we can observe that although the test accuracy rates are quite similar for both cases, the accuracy rate of MSE is slightly better on the test data in both cases than the perceptron classifier.

Perceptron aims to minimize the criterion function, i.e., minimize the number of misclassified data points. MSE on the other hand focuses on making a boundary to divide the data into different classes by minimizing the distance between the data points and the boundary. This is however not the general case that MSE always has greater accuracy than perceptron. The vice versa is also possible. It's only in this case that we get this result.