Homework #2

CMPSC 165B - Machine Learning

Kenny Hoang Nguyen
X299187

February 2020

1 Logistic regressor vs SVM

In this task we implemented the SGD with a logistic regressor which we used to classify the data. We implemented the one vs all method for each class and then decided which class to put each element in by looking at the one with highest cross entropy. In other words I ran the binary classifier three times. This resulted in three k-dimensional lines that separated the class poor from median and excellent, median from poor and excellent, and excellent from poor and median. The predictions resulted in three vectors that classifies n-data from the matrix x in testing data. For each of these predictions the row was decided to belong to class poor, median or excellent by looking at which resulted in the highest cross-entropy.

The data was partitioned into five sets that were equal size and then trained on and tested on five times. The nepoch it was run on was 350. The accuracy was calculated with

$$\text{Accuracy} = \frac{\text{Correct predictions}}{\text{total number predictions}} \tag{1}$$

The same five sets were run in SVM which resulted in the accuracy in table 1 and 1. Each of these runs used the time listed in the table 3 and 4

Then for each of the data we got the average accuracy given in 5:

We see that the SVM and SGD has somewhat same performance with regards to accuracy, but SVM has on average some more accurate predictions. However the SGD is almost 1000x more efficient when it comes to training of the data set, so it is in that sense better since it is more effective with about the same accuracy. From the confusion matrix it seems that it in almost all cases predicted that the data belonged to class 1. Which kind of made sense since the correlation between x and y was not really high, meaning that when most of the data was biased to be in class 1 the trained line also did. I also did remove the columns in x that was highly correlated (had correlation coefficient > 0.6), but I still got the same predictions. I did check with my own sample data putting some data samples in each vertices in a triangle in a 2D plane, in that case my classifier managed to classify each sample correctly, so I'm not sure what to do to make it work correctly for the wine data.

Run number	Accuracy SGD	Accuracy SVM
1	76.56%	70.31%
2	84.38%	90.62%
3	76.56%	73.44%
4	71.88%	64.06%
5	73.44%	70.31%

Table 1: The accuracy for the red wine set

Run number	Accuracy SGD	Accuracy SVM
1	72.96%	70.92%
2	75.00%	75.51%
3	72.45%	75.00%
4	77.55%	73.98%
5	76.53%	77.04%

Table 2: The accuracy for the white wine set

Run number	Training SGD	Testing SGD	Training SVM	Testing SVM
1	17.25	0.0013	0.011	0.0011
2	15.85	0.0013	0.0090	0.0010
3	16.89	0.0013	0.0098	0.0010
4	13.99	0.0015	0.010	0.0010
5	15.42	0.002	0.0197	0.002

Table 3: The runtime for the red wine dataset in seconds

Run number	Training SGD	Testing SGD	Training SVM	Testing SVM
1	87.053	0.022	0.103	0.0068
2	90.16	0.010	0.104	0.0064
3	92.91	0.011	0.112	0.0068
4	78.26	0.012	0.097	0.0073
5	107.87	0.012	0.114	0.006

Table 4: The runtime for the white wine dataset in seconds

Data set	Average accuracy
Red wine SGD	75.56%
Red wine SVM	73.75%
White wine SGD	74.90%
White wine SVM	74.49%

Table 5: The average accuracy for each run

Data set	Training	Testing
Red wine SGD	15.88	0.0015
Red wine SVM	0.012	0.0012
White wine SGD	91.25	0.013
White wine SVM	0.11	0.0067

Table 6: The average runtime for each data set