CENG499 hw3 Report

Ali Dogan

January 8, 2021

1 Part 1: Decision Tree

1.1 Information Gain

accuracy : 823 / 864 : 0.952
The tree can be found in trees folder under the name dt_info_gain_wo_pruning.

1.2 Gain Ratio

accuracy : 826 / 864 : 0.956
The tree can be found in trees folder under the name dt_gain_ratio_wo_pruning.

1.3 Average Gini Index

accuracy : 823 / 864 : 0.952
The tree can be found in trees folder under the name dt_avg_gini_wo_pruning.

1.4 Gain Ratio with Chi-squared Pre-pruning

accuracy : 819 / 864 : 0.947
The tree can be found in trees folder under the name dt_chi_prepruning.

1.5 Gain Ratio with Reduced Error Post-pruning

accuracy : 823 / 864 : 0.952
The tree can be found in trees folder under the name dt_reduced_postpruning.

2 Part 2: Support Vector Machine

2.1 First Part

As we see in the plots, the larger the C gets, the classifier chooses a smaller margin. So, the larger C implies that we give more importance(weight) to the classification accuracy than the hyperplane with the largest minimum margin.

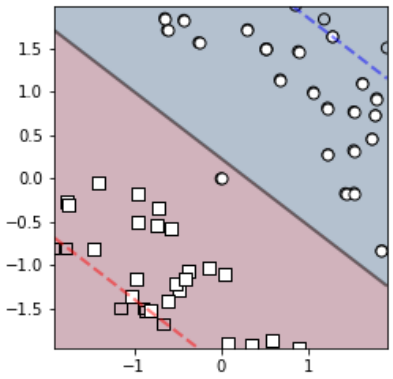

Figure 1: svm with c = 0.01

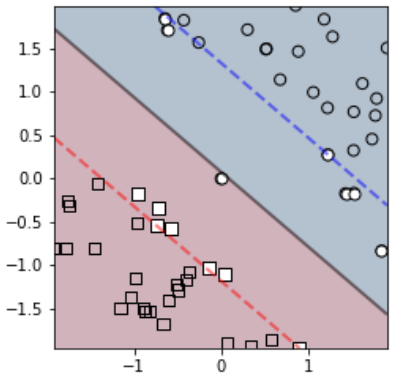

Figure 2: svm with c = 0.1

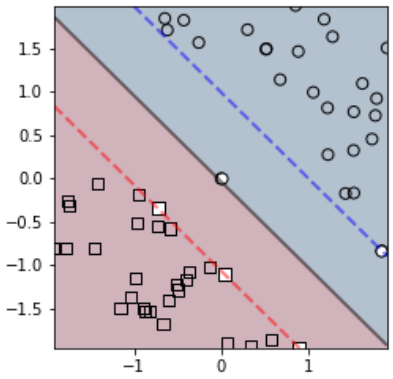

Figure 3: svm with c = 1

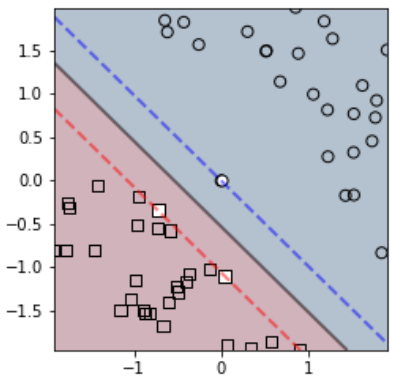

Figure 4: svm with c = 10

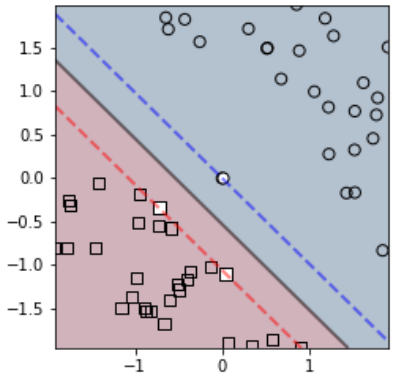

Figure 5: svm with c = 100

2.2 Second Part

Changing the kernel can drastically effect the boundary lines and classification accuracy depending on the pattern in the data. As we see below, data points can not be separated with a linear boundary as the rectangles are clustered in the middle surrounded by circles. Having a linear transformation is not feasible. RBF, in this case works pretty good. Polynomial do not make a good classification due its similarity to linear kernel.

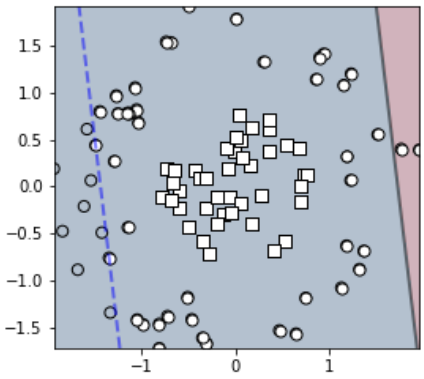

Figure 6: svm with linear kernel

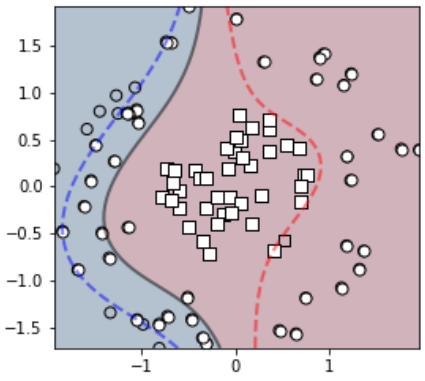

Figure 7: svm with polynomial kernel

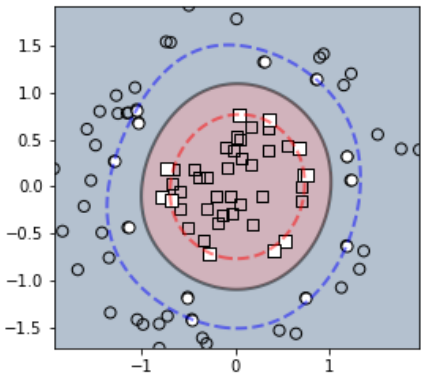

Figure 8: svm with rbf kernel

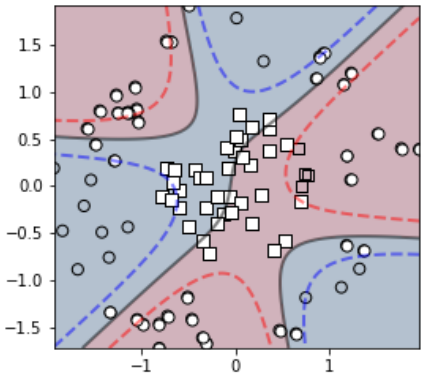

Figure 9: svm with sigmoid kernel

2.3 Third Part

Best hyperparameters obtained from grid search are as follows:
kernel : RBF
C : 100
gamma : 0.01
Test Accuracy : 0.80

gamma	C				
	0.01	0.1	1	10	100
-	0.64	0.64	0.7	**0.71**	0.71

Table 1: Linear kernel

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.54	0.54	0.54	0.56	0.63
0.0001	0.54	0.54	0.63	0.73	0.67
0.001	0.54	0.54	0.71	0.71	0.73
0.01	0.54	0.54	0.56	0.62	**0.74**
0.1	0.55	0.54	0.69	0.74	0.71
1	0.54	0.54	0.71	0.71	0.71

Table 2: RBF kernel

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.54	0.54	0.54	0.54	0.54
0.0001	0.54	0.54	0.54	0.54	0.56
0.001	0.54	0.56	0.61	0.69	0.73
0.01	0.69	0.73	0.73	0.73	0.73
0.1	0.73	0.73	**0.73**	0.73	0.73
1	0.73	0.73	0.73	0.73	0.73

Table 3: Polynomial kernel

gamma	C				
	0.01	0.1	1	10	100
0.00001	0.54	0.54	0.54	0.54	0.61
0.0001	0.54	0.54	0.54	0.54	0.47
0.001	0.54	0.54	0.54	0.54	0.54
0.01	0.54	0.54	0.54	0.61	**0.64**
0.1	0.54	0.54	0.57	0.54	0.47
1	0.49	0.54	0.54	0.54	0.54

Table 4: Sigmoid kernel

2.4 Fourth part

2.4.1 Without handling the imbalance problem

Accuracy : 0.83

		Predicted	
		0	1
Actual	0	1	190
	1	0	949

The accuracy is not a good performance metric on its own. We need to look at other metrics as well, especially when we have biased data set such as having imbalanced set. We see in confusion metric that, basically the classifier predicts everything as 1 and gets a 0.83 accuracy due to imbalance problem. Such metrics can be f1 score, calculated with precision and recall scores which are obtained from confusion matrix.

2.4.2 Oversampling the minority class

Accuracy : 0.8

		Predicted	
		0	1
Actual	0	73	118
	1	115	834

Now we have a more balanced confusion matrix. Although the majority of predictions is still in 1, the false predictions (115 and 118) are distributed among 1's and 0's.

2.4.3 Undersampling the majority class

Accuracy : 0.59

		Predicted	
		0	1
Actual	0	119	72
	1	400	549

Undersampling the data caused a very nice ratio in the predictins, i.e., the predictions are distributed among 1's and 0's almost equally. However, the accuracy is the lowest among others but this is not a surprise if we have almost equal number of 1 and 0 predictions in this test set.

2.4.4 Setting the class_weight to balanced

Accuracy : 0.76

		Predicted	
		0	1
Actual	0	87	104
	1	173	776

The distributions are similar to the oversampling with a satisfied accuracy.