COMP 3105 - Assignment 2

Alex Nedev(101195595), Jacob Lane(101220538)

October 22, 2023

Question 1

Part d

Table 1: Training accuracies with different hyper-parameters

		BinDev			Hinge	
λ	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
0.001	0.9993	0.5767	0.8712	0.9982	0.5936	0.8636
0.01	0.998	0.5728	0.8724	0.9969	0.5936	0.8668
0.1	0.9941	0.5733	0.8754	0.9916	0.5843	0.8651
1.0	0.9853	0.5737	0.8637	0.9817	0.6012	0.8605

Table 2: Test accuracies with different hyper-parameters

		BinDev			Hinge	
λ	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
0.001	0.99008	0.53913	0.88042	0.99307	0.5768	0.87252
0.01	0.98739	0.54388	0.88108	0.99197	0.58187	0.87419
0.1	0.98458	0.54289	0.88113	0.9866	0.57149	0.87243
1.0	0.97699	0.54383	0.86727	0.97646	0.57919	0.86839

Part e

The accuracies that were obtained above make sense for the different models of data that we can play with. The linear models match up very nicely because we're essentially trying to split up the data set as well as we can while using a linear function/straight line. This can be done rather easily for model 1 because of the way the data is generated. However, model 2 is quite difficult to split up properly using a straight line because linear functions are not really the greatest choice when it comes to trying to split up points that are generated in a circle. Thus is makes sense to see lower accuracies for the second model. The third model's wavy point generation can feature some success when trying to split it up using linear functions though. If we line up our line closer to the off diagonal(top left corner to bottom right corner), then we can properly identify a good majority of our data points, leading to the higher accuracy that we see in our tales.

We observe this same trend for both the regularized binomial deviance loss and the regularized hinge loss. Both the binomial deviance loss and the hinge loss can feature very similary accuracies if the margin is the right value. Working with the data of the question, the margin must have been large enough such that both types of losses were able to feature similar enough accuracies.

Finally, the accuracies seem to be rather similar when comparing the test accuracies to their training counterparts across the board. This is nice because it means that we haven't overfit or underfit our data anywhere, likely indicating that the functions for the previous parts of question 1 were implemented correctly.

Question 2

Part d

Table 1: Training accuracies with different hyper-parameters

		BinDev			Hinge	
Kernel	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
Linear	0.999	0.543	0.865	0.967	0.564	0.823
Poly(d=2)	1.	1.	0.877	0.949	0.591	0.523
Poly(d=3)	1.	1.	1.	0.958	0.603	0.5
Gauss($\sigma = 1$)	1.	1.	1.	0.971	0.632	0.826
Gauss($\sigma = 0.5$)	1.	1.	1.	0.978	0.876	0.931

Table 2: Test accuracies with different hyper-parameters

		BinDev			Hinge	
Kernel	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
Linear	0.9885	0.5304	0.883	0.9562	0.509	0.8251
Poly(d=2)	0.9858	0.9805	0.8804	0.9411	0.5999	0.5202
Poly(d=3)	0.9767	0.9695	0.9944	0.9513	0.5996	0.5
Gauss($\sigma = 1$)	0.9756	0.9694	0.9981	0.9536	0.6138	0.8313
Gauss($\sigma = 0.5$)	0.9578	0.9639	0.9989	0.9539	0.8487	0.936

Part e

The accuracies for BinDev are what I would expect, very good training data and corresponding very good test accuracies, The linear model with model 2 for both test and training accuracies is very bad because there is no good linear model for a circle of data points

Hinge has alot more interesting data though, model 1 has very promissing accuracies with every kernel and model in the mid to high 90's, this shows to me that my current code is accurate, however it is with both model 2 and model 3 that the acuracy becomes much worse, with model 2 once again being the most inaccurate, except for gauss with 1.0 parameter, because there are no good linear and poly kernels for hinge function. Finally, there are no good Poly kernels for data set 3, there is a clear rough linear line that you could use with a mild innaccuracy ad obvious gauss kernel options, but there is no Poly kernel.

Question 3

Part d

λ	Linear	Poly(d=3)	Gauss($\sigma = 1.0$)
0.001	0.955	0.967	0.128
1.0	0.955	0.966	0.125

When trying to select the best performing hyper-parameter and kernel function, we do it by evaluating which combination of hyper-parameters and kernels have the highest accuracy. In the table above, the hyper-parameter $\lambda = 0.001$ and the Polynomial kernel function with distance 3 produce the highest accuracy out of all the combinations here, and thus we selected them for our chosen "best" hyper parameter and Kernel function combination.