E0-270 Machine Learning Assignment-2

Name: Prateek Yadav
Date: 12 March 2017

Question-1

a

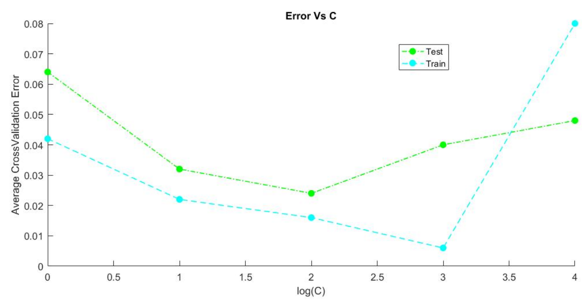


Figure 1:

Table 1:

C	Average Training Set Error	Average Test Set Error
1	0.113	0.131
10	0.059	0.112
100	0.033	0.112
1000	0.021	0.117
10000	0.089	0.134

b *Linear Kernel*

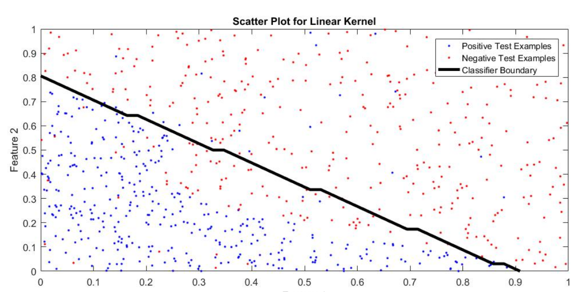


Figure 2: Scatter Plot for Linear Kernel

Table 2:

C	Average Training Set Error	Average Test Set Error
1	0.106	0.116
10	0.107	0.112
100	0.106	0.108
1000	0.109	0.115
10000	0.107	0.120

Training Error using C that gives minimum Validation error is 0.1
Test Error using C that gives minimum Validation error is 0.126

Degree two Polynomial Kernel

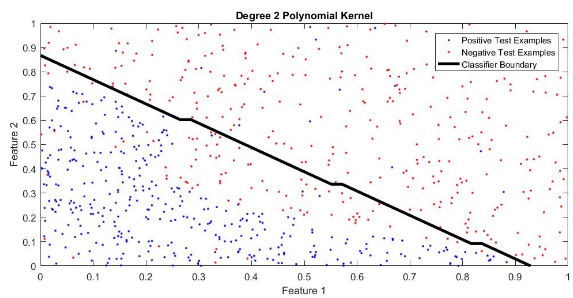


Figure 3: Scatter Plot for Degree two polynomial Kernel

Table 3:

C	Average Training Set Error	Average Test Set Error
1	0.110	0.103
10	0.111	0.121
100	0.129	0.140
1000	0.131	0.155
10000	0.144	0.126

Training Error using C that gives minimum Validation error is 0.109.
Test Error using C that gives minimum Validation error is 0.138

Degree three Polynomial Kernel

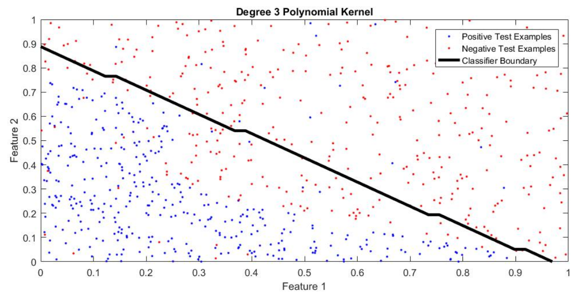


Figure 4: Scatter Plot for Degree three polynomial Kernel

Table 4:

C	Average Training Set Error	Average Test Set Error
1	0.110	0.113
10	0.113	0.117
100	0.105	0.119
1000	0.115	0.117
10000	0.108	0.125

Training Error using C that gives minimum Validation error is 0.109
Test Error using C that gives minimum Validation error is 0.151

Degree RBF Kernel

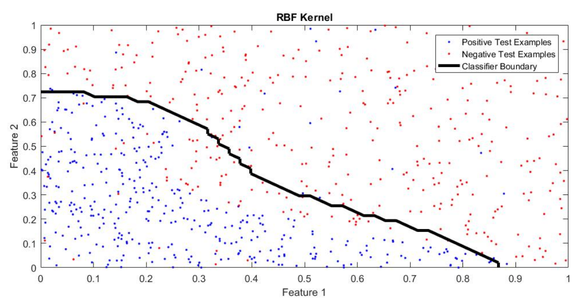


Figure 5: Scatter Plot for RBF Kernel

Table 5:

	$\sigma = 1/32$	$\sigma = 1/4$	$\sigma = 1$	$\sigma = 4$	$\sigma = 32$
C=1	0.15	0.10	0.116	0.133	0.570
C=10	0.15	0.104	0.116	0.109	0.114
C=100	0.147	0.134	0.116	0.109	0.122
C=1000	0.138	0.173	0.135	0.12	0.117
C=10000	0.146	0.2	0.154	0.115	0.113

Training Error using C that gives minimum Validation error is 0.095
Test Error using C that gives minimum Validation error is 0.115

c For SVM using linear kernel the best test accuracy is 88.61% usinf the value of C = 10 while for Logistic Regression the best test accuracy was 90.01% using $\lambda = 0.001$. Logistic Regression performed better on the given dataset as compared to SVM with linear kernel. Generally we hope that SVM will perform better on test set as SVM with linear kernel tries to find out a decision boundary while maximizing the geometric margins. In this case as the data is not linearly seperable So SVM also tries to minimize ξ for all given constraints which in turn deteriorates the performance on test set and hence logistic regression performs better.

Question-2

c Average Squared error on training Set is 101.3872% and Average squared error on test set is 26.56%

d

Table 6:

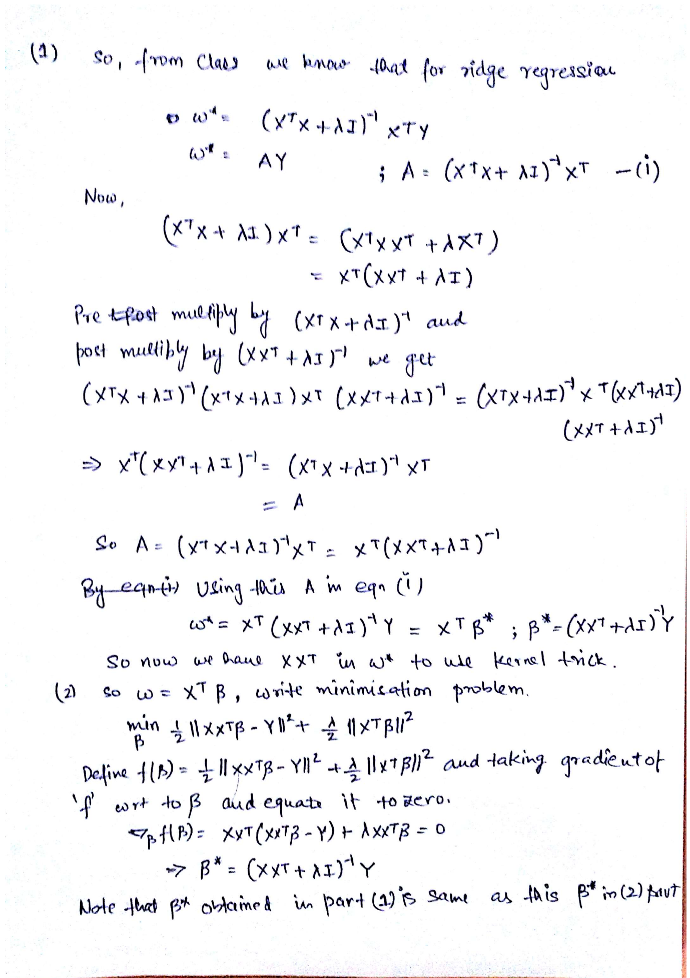
λ	**FOLD1**	**FOLD2**	**FOLD3**	**FOLD4**	**FOLD5**	**Mean**
0.01	85.774	113.915	106.854	99.625	102.167	101.667
0.1	85.764	113.954	106.862	99.652	102.462	101.739
1	85.759	113.934	106.841	99.631	102.698	101.773
10	86.751 8	113.978	106.831	99.745	102.548	101.971
100	86.548	114.745	107.954	99.364	103.265	102.375

Table 7:

λ	Training Set Error	Test Set Error
0.01	101.384	26.856
0.1	101.384	26.824
1	101.384	26.843
10	101.389	26.681
100	101.563	25.259

The value $\lambda = 10$ (green row) indicates the value of lambda selected through cross validation while the cyan one indicated λ at which minimum test error is obtained. Usually cross Validation gives the right value of λ but in this case we are getting a different value.
The performance obtained from Linear least squared learner is 26.861 while the best performance obtained on test set error using ridge regression is 26.681. So we can see that ridge regression gives a slightly better performance than linear least squared regression and this is mainly due to regularisation effect of λ.

e



(1) So, from Class we know that for ridge regression

$$\omega^* = (X^TX+\lambda I)^{-1}X^TY$$

$$\omega^* = AY \quad ; \; A = (X^TX+\lambda I)^{-1}X^T \quad -(i)$$

Now,

$$(X^TX+\lambda I)X^T = (X^TXX^T+\lambda X^T)$$
$$= X^T(XX^T+\lambda I)$$

Pre ~~+post~~ multiply by $(X^TX+\lambda I)^{-1}$ and post multiply by $(XX^T+\lambda I)^{-1}$ we get

$$(X^TX+\lambda I)^{-1}(X^TX+\lambda I)X^T(XX^T+\lambda I)^{-1} = (X^TX+\lambda I)^{-1}X^T(XX^T+\lambda I)(XX^T+\lambda I)^{-1}$$

$$\Rightarrow X^T(XX^T+\lambda I)^{-1} = (X^TX+\lambda I)^{-1}X^T$$
$$= A$$

So $A = (X^TX+\lambda I)^{-1}X^T = X^T(XX^T+\lambda I)^{-1}$

~~By eqn (i)~~ Using this A in eqn (i)

$$\omega^* = X^T(XX^T+\lambda I)^{-1}Y = X^T\beta^* \quad ; \; \beta^* = (XX^T+\lambda I)^{-1}Y$$

So now we have XX^T in ω^* to use kernel trick.

(2) so $\omega = X^T\beta$, write minimisation problem.

$$\min_{\beta} \frac{1}{2}\|XX^T\beta - Y\|^2 + \frac{\lambda}{2}\|X^T\beta\|^2$$

Define $f(\beta) = \frac{1}{2}\|XX^T\beta - Y\|^2 + \frac{\lambda}{2}\|X^T\beta\|^2$ and taking gradient of 'f' wrt to β and equate it to zero.

$$\nabla_\beta f(\beta) = XX^T(XX^T\beta - Y) + \lambda XX^T\beta = 0$$

$$\Rightarrow \beta^* = (XX^T+\lambda I)^{-1}Y$$

Note that β^* obtained in part (1) is same as this β^* in (2) part

Figure 6:

f

Table 8:

λ	**FOLD1**	**FOLD2**	**FOLD3**	**FOLD4**	**FOLD5**	**Mean**
0.01	56.734	79.614	73.184	62.816	67.094	67.888
0.1	56.649	79.628	73.178	62.819	67.035	67.861
1	56.639	79.568	73.249	62.716	67.037	67.842
10	56.873	80.230	74.848	62.644	67.637	68.446
100	61.648	87.024	82.864	67.424	71.441	74.082

Table 9:

λ	Training Set Error	Test Set Error
0.01	67.103	15.291
0.1	67.103	15.258
1	67.109	14.918
10	67.579	12.705
100	72.667	10.003

The value $\lambda = 10$ (green row) indicates the value of lambda selected through cross validation while the cyan one indicated λ at which minimum test error is obtained. Usually cross Validation gives the right value of λ but in this case we are getting a different value.
The performance obtained from ridge regression is 26.681 while the best performance obtained on test set error using poly-3 ridge regression is 14.918. So we can see that poly-3 ridge regression gives significantly better results on test set error compared to linear ridge regression. So as Poly-3 ridge performs significantly better than linear ridge it is a strong evidence that the data we are trying to model is not linear and most probably cubic.

Question-3

b

Table 10:

C	**FOLD1**	**FOLD2**	**FOLD3**	**FOLD4**	**FOLD5**	**Mean**
0.1	95.168	129.648	121.145	104.624	112.348	112.586
1	89.731	124.349	115.452	106.846	105.397	108.355
100	89.921	124.653	115.568	106.689	105.412	108.448

Table 11:

C	Training Set Error	Test Set Error
0.01	110.138	18.451
1	108.617	21.346
100	108.841	21.624

The cyan coloured row indicates the value of C selected through cross validation while the green one indicated C at which minimum test error is obtained. Usually cross Validation gives the right value of C but in this case we are getting a different value.
The test set error for Linear Ridge regression is 26.681, while for Linear least squared regression is 26.861 and for Linear SVR is 21.346.
So we can see that SVR gives better results on test set error as compared to

other two. The main reason of this is that the objective function of SVR find a hyperplane such that most of the points lie within a specified margin from that plane.

c

Table 12:

C	**FOLD1**	**FOLD2**	**FOLD3**	**FOLD4**	**FOLD5**	**Mean**
0.1	74.125	105.769	98.245	85.248	89.317	90.541
1	67.967	97.921	91.357	76.861	82.897	83.401
100	67.657	97.413	90.762	76.530	82.621	82.996

Table 13:

C	Training Set Error	Test Set Error
0.01	90.148	6.341
1	83.241	0.684
100	82.986	0.697

The cyan coloured row indicates the value of C selected through cross validation while the green one indicated C at which minimum test error is obtained. Usually cross Validation gives the right value of C but in this case we are getting a different value.
The test set error for Cubic Ridge regression is 14.918, while for Linear SVR is 21.346 and for Cubic SVR is 0.697.
So we can see that Cubic SVR gives significantly better results on test set error as compared to other two. By this we can say that the data is distributed close to cubic. As SVR keep most points within an ϵ-margin from the predictions it outperforms the cubic ridge regression model.

Question-4

a

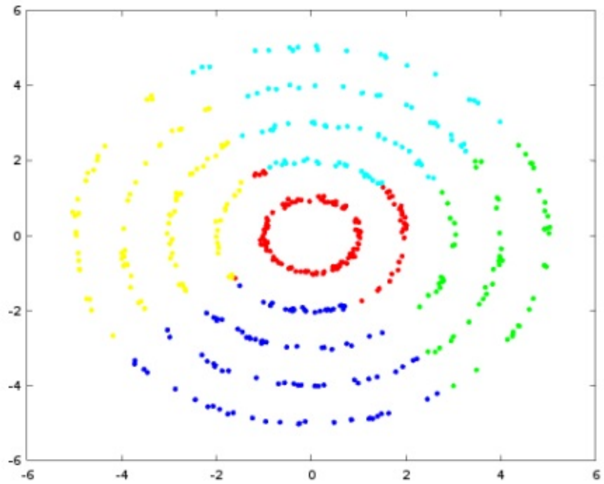


Figure 7: k-Means Clustering

b The results are not satisfactory because the kernelized k-Means uses euclidean norm as the distance measure which forces the cluster to be hyper-spheres. A better clustering would be when concentric circles are get assigned to different clusters, however that would imply that all our centroids lie at the center of these circles since we are using euclidean norm distance which cannot lead to the most natural clustering.

c

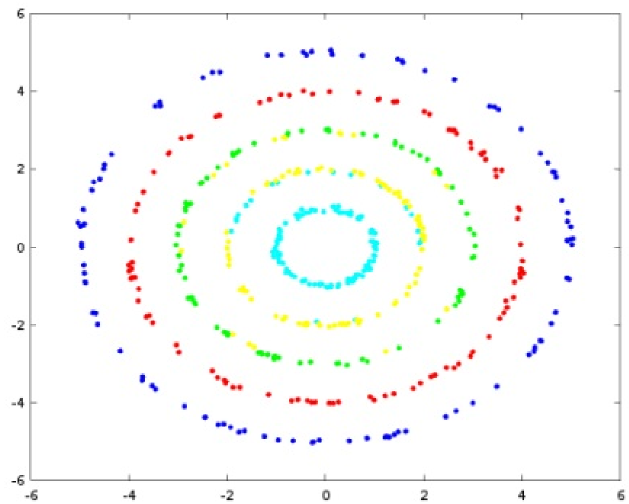


Figure 8: Kernelized k-Means

d RAND Index for k-Means Clustering is 0.7067 and RAND index for Kernelized k-Means clustering is 0.931463