E0 270 Machine Learning *Due:* Feb 7, 2012

Assignment 2 - Report

sachin nagargoje *04-04-00-10-41-11-1-08449*

1 Solution: Problem 1

Let $f\colon \chi \mapsto \mathbb{R}$ be any arbitrary real-valued function that is used for labeling new instances from χ. then,

Since $y = g(x) + \eta$

$$E_{(x,y)\sim D}[(f(x) - y)^2] = E[(f(x) - (g(x) + \eta))^2]$$

Let $k = f(x) - g(x)$, then

$$E[(k - \eta)^2] = E[k^2 + \eta^2 - 2k\eta]$$
$$= E[k^2] + E[\eta^2] - 2E[k]E[\eta]$$

Since η follows Standard Normal Distribution ,

$$E[\eta] = 0$$
$$var(\eta) = 1$$

So,

$$E[\eta^2] = var(\eta) + E[\eta]^2$$
$$= 1 + 0$$
$$= 1$$

Thus ,

$$E[(k - \eta)^2] = E[k^2] + 1 - 0$$
$$= E[k^2] + 1$$

Since, $E[k^2]$ is expectation of a positive quantity,

$$E[k^2] \geq 0$$

and,

$$\begin{aligned} E[(g(x)-y)^2] &= E[(g(x)-g(x)-\eta)^2] \\ &= E[\eta^2] \\ &= 1 \end{aligned}$$

So we got ,
error with $g(x)$ as classifier = 1.
and error with any arbitrary function $f(x)$ as classifier $= 1 + E[k^2]$,
where $k = f(x) - g(x)$
and also $E[k^2]$ is positive quantity.
Thus we conclude that,

$$E_{(x,y)\sim D}[(g(x)-y)^2] \leq E_{(x,y)\sim D}[(f(x)-y)^2] \forall f : \chi \mapsto \mathbb{R}$$

2 Solution: Problem 2

Part (a)

Refer to Code 1

Part (b)

Refer to Code 2

Part (c)

On running the code on given data, following error were found :

Training Error : 101.3872
Test Error : 26.8586

Part (d)
Refer to code 3
Test Error

Folds	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
1	85.7615	85.7628	85.7780	86.0910	92.4212
2	113.9051	113.9046	113.9033	114.1163	121.1080
3	106.7523	106.7523	106.7560	107.0459	115.0829
4	99.7846	99.7746	99.6801	99.1158	103.6544
5	102.3173	102.3238	102.3902	103.1771	112.0226

Train Error

Folds	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
1	105.3216	105.3216	105.3245	105.5627	112.5263
2	98.2991	98.2991	98.3021	98.5515	105.6139
3	100.0530	100.0530	100.0561	100.3096	107.3352
4	101.8518	101.8518	101.8551	102.1285	109.6258
5	101.2350	101.2350	101.2378	101.4685	108.1266

Average Train Error

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
101.3521	101.3521	101.3551	101.6042	108.6456

Average Test Error

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
101.7041	101.7036	101.7015	101.9092	108.8578

As we can observe here, $\lambda = 1.0$ gives minimum average cross validation error.
Using this value of λ with complete train data, we got following errors :
Test Error : 26.7510
Train Error : 101.3891

Test Error using Complete Data

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
26.8575	26.8476	26.7510	26.0140	26.8059

Train Error using Complete Data

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
101.3872	101.3872	101.3891	101.5543	107.1628

As we can see here,cross-validation selects $\lambda = 1.0$ which gives minimum average test error, but this value of λ does not give minimum error when run on the test set using complete training data.

Comparing to linear least square regression model , we see that training and test errors are almost same.

Part (e)
Refer to Code 5 and 6

Test Error

Folds	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
1	53.3263	53.3083	53.2558	54.8039	62.0508
2	76.6655	76.6489	76.6283	78.6544	87.9578
3	70.9982	71.0392	71.4627	74.5822	84.7555
4	61.0854	61.0556	60.9238	62.4248	69.5314
5	65.4179	65.4269	65.5327	66.5064	71.8632

Train Error

Folds	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
1	68.0576	68.0590	68.1586	70.0791	77.8143
2	62.2223	62.2236	62.3253	64.2349	71.7455
3	63.6660	63.6671	63.7477	65.4025	72.6768
4	66.0990	66.1004	66.2049	68.1582	75.8957
5	65.0791	65.0801	65.1611	66.9815	75.2170

Average Train Error

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
65.0248	65.0261	65.1195	66.9712	74.6699

Average Test Error

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
65.4987	65.4958	65.5607	67.3943	75.2317

As we can observe here, $\lambda = 0.10$ gives minimum average cross validation error.
Using this value of λ with complete train data, we got following errors :
Test Error : 16.1780
Train Error : 65.0730

Test Error using Complete Data

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
16.2563	16.1780	15.5365	13.6111	14.0447

Train Error using Complete Data

$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 1.0$	$\lambda = 10.0$	$\lambda = 100.0$
65.0722	65.0730	65.1357	66.6520	73.5195

As we can see here,cross-validation selects $\lambda = 0.10$ which gives minimum average test error, but this value of λ does not give minimum error when run on the test set using complete training data.

Comparing to linear least square regression model , we see that training and test errors are considerably smaller in this case.

3 Solution: Problem 3

Part (a)
Refer to Code 8

Part (b)
not attempted

Part (c)
not attempted

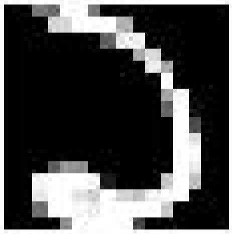
4 Solution: Problem 4

Refer to code 10

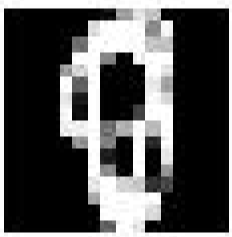
	k = 1	k = 9	k = 99
Training Error	0	0.0291	0.0968
Test Error	0.0792	0.1081	0.2361

Here $k = 1$ has lowest test error. Misclassified images are as follows :

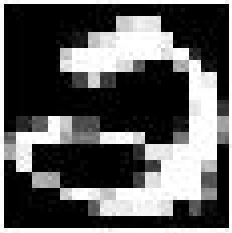
1. 3 instead of 2



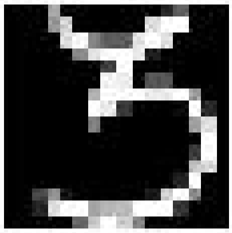
2. 9 instead of 0



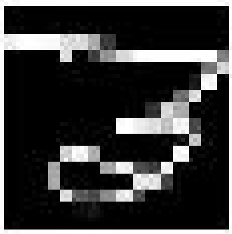
3. 0 instead of 2



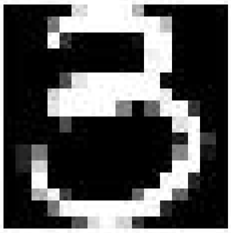
4. 5 instead of 3



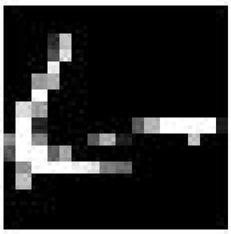
5. 7 instead of 3



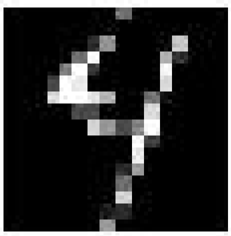
6. 8 instead of 3



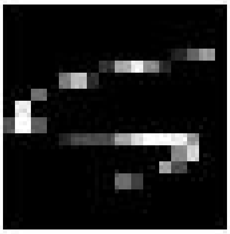
7. 2 instead of 6



8. 7 instead of 4



9. 4 instead of 5



10. 2 instead of 6