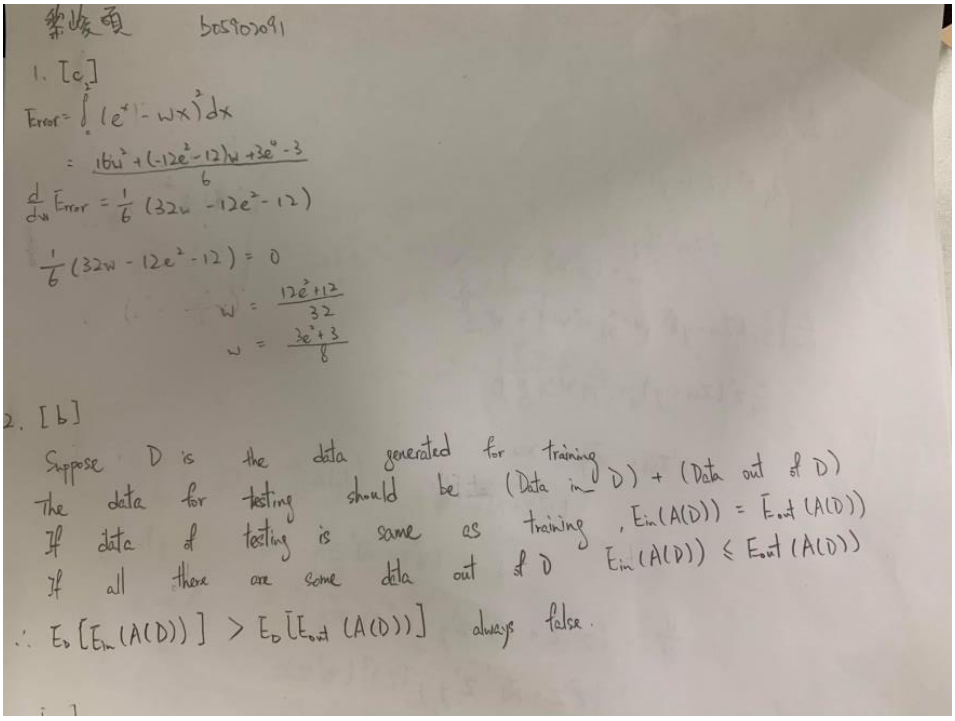


黎峻碩 b05902091

1. [c]

$$\text{Error} = \int_0^6 (e^x - wx)^2 dx$$

$$= \frac{16w^2 + (-12e^2 - 12)w + 3e^6 - 3}{6}$$

$$\frac{d}{dw}\text{Error} = \frac{1}{6}(32w - 12e^2 - 12)$$

$$\frac{1}{6}(32w - 12e^2 - 12) = 0$$

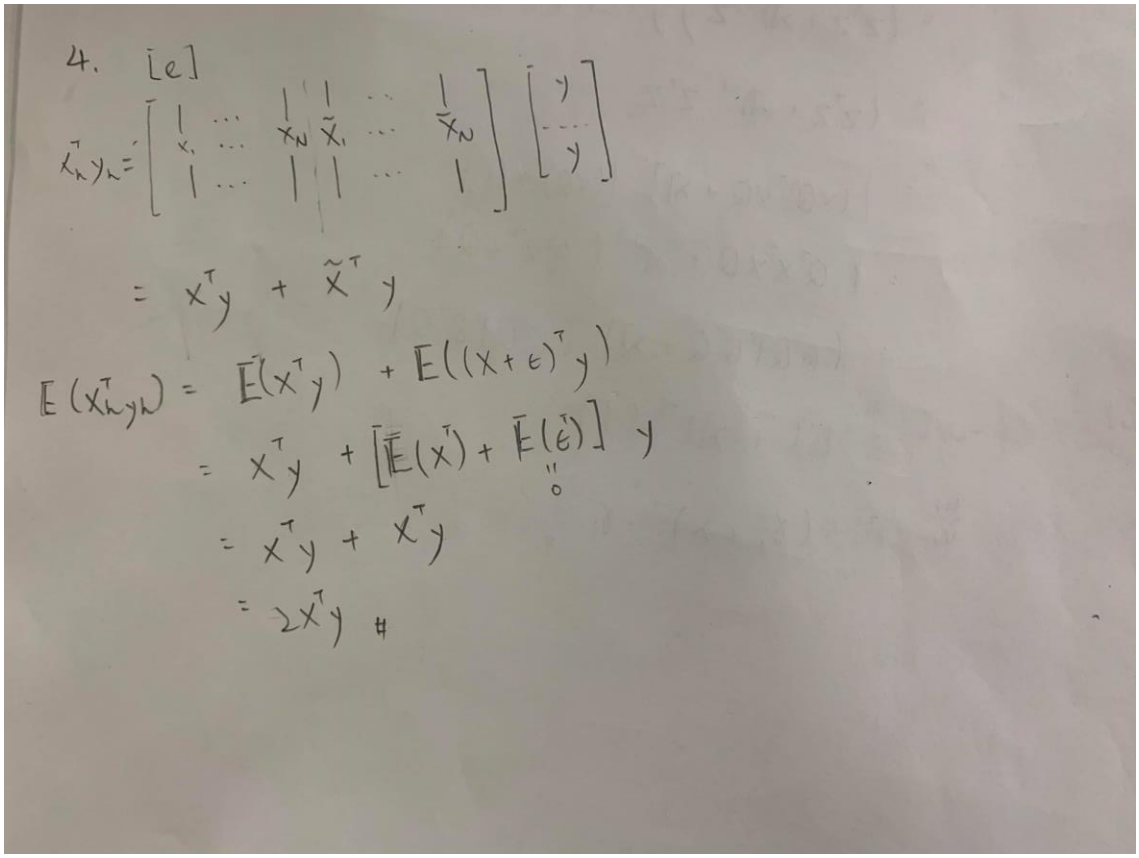
$$w = \frac{12e^2 + 12}{32}$$

$$w = \frac{3e^2 + 3}{8}$$

2. [b]

Suppose D is the data generated for training
The data for testing should be (Data in D) + (Data out of D)
If data of testing is same as training, $E_{in}(A(D)) = E_{out}(A(D))$
If all there are some data out of D $E_{in}(A(D)) \leq E_{out}(A(D))$

$\therefore E_D[E_{in}(A(D))] > E_D[E_{out}(A(D))]$ always false.



4. [e]

$$X_N^T y_N = \begin{bmatrix} 1 & \cdots & 1 & 1 & \cdots & 1 \\ x_1 & \cdots & x_N & \tilde{x}_1 & \cdots & \tilde{x}_N \\ 1 & \cdots & 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y \\ \cdots \\ y \end{bmatrix}$$

$$= X^T y + \tilde{X}^T y$$

$$\mathbb{E}(X_N^T y_N) = \mathbb{E}(X^T y) + \mathbb{E}((X+\epsilon)^T y)$$

$$= X^T y + [\mathbb{E}(X^T) + \mathbb{E}(\epsilon^T)]\, y$$

$$= X^T y + X^T y$$

$$= 2X^T y \quad \#$$

5. [d]

$$\frac{d}{dw}\frac{1}{N}\|Zw-y\|^2 = 0$$

$$\frac{2}{N}Z^T(Zw-y) = 0$$

$$Z^TZw - Z^Ty = 0$$

$$w = (Z^TZ)^{-1}Z^Ty = v$$

$$\frac{d}{dw}\left[\frac{1}{N}\|Zw-y\|^2 + \frac{\lambda}{N}w^Tw\right] = 0$$

$$\frac{2}{N}Z^T(Zw-y) + \frac{2\lambda}{N}w = 0$$

$$Z^TZw - Z^Ty + \lambda w = 0$$

$$(Z^TZ+\lambda I)w = Z^Ty$$

$$w = (Z^TZ+\lambda I)^{-1}Z^Ty = u$$

$$\frac{u}{v} = (Z^TZ+\lambda I)^{-1}Z^Ty\left[(Z^TZ)^{-1}Z^Ty\right]^{-1}$$

$$= (Z^TZ+\lambda I)^{-1}Z^Ty\,y^{-1}(Z^T)^{-1}(Z^TZ)$$

$$= (Z^TZ+\lambda I)^{-1}Z^TZ$$

$$= \left[(XQ)^TXQ+\lambda I\right]^{-1}\left[(XQ)^TXQ\right]$$

$$= (Q^TX^TXQ+\lambda I)^{-1}(Q^TX^TXQ)$$

$$= (Q^TQ\Gamma Q^TQ+\lambda I)^{-1}(Q^TQ\Gamma Q^TQ)$$

$$= (\Gamma+\lambda I)^{-1}(\Gamma)$$

$$\frac{w_i}{v_i} = (\gamma_i+\lambda)^{-1}\cdot\gamma_i$$

6. [a]

$$\frac{d}{dw}\left(\frac{1}{N}\sum_{n=1}^{N}(w\cdot x_n-y_n)^2+\frac{\lambda}{N}w^2\right) = 0$$

$$\frac{2}{N}\sum_{n=1}^{N}x_n(w\cdot x_n-y_n)+\frac{2\lambda}{N}w = 0$$

$$\sum_{n=1}^{N}x_n^2w - \sum_{n=1}^{N}x_ny_n + \lambda w = 0$$

$$w\left(\sum_{n=1}^{N}x_n^2+\lambda\right) = \sum_{n=1}^{N}x_ny_n$$

$$w = \frac{\sum_{n=1}^{N}x_ny_n}{\sum_{n=1}^{N}x_n^2+\lambda}$$

$$C = w^2 = \left(\frac{\sum_{n=1}^{N}x_ny_n}{\sum_{n=1}^{N}x_n^2+\lambda}\right)^2$$

7. [d]

$$\frac{d}{dy}\left(\frac{1}{N}\sum_{n=1}^{N}(y-y_n)^2+\frac{2K}{N}\Omega(y)\right) = 0$$

$$\frac{1}{N}\sum_{n=1}^{N}2(y-y_n)+\frac{2K}{N}\Omega'(y) = 0$$

When $K\to\infty$ $\quad\frac{(\sum_{n=1}^{N}y_n)+K}{N+2K} \approx \frac{1}{2}$

$$\Omega'\left(\frac{1}{2}\right) = 0$$

$$\Omega(y) = (y-0.5)^2$$

8. [1]

8. [b]

$$\frac{1}{N}\sum_{n=1}^{N}(\tilde{w}^T\Gamma^{-1}x - y_n)^2 + \frac{\lambda}{N}(\tilde{w}^T\tilde{w})$$

Sub $\tilde{w}^T\Gamma^{-1}$ into w

$$\frac{1}{N}\sum_{n=1}^{N}(wx - y_n)^2 + \frac{\lambda}{N}[(\Gamma w)^T(\Gamma w)]$$

$$= \frac{1}{N}\sum_{n=1}^{N}(wx - y_n)^2 + \frac{\lambda}{N}(w^T\Gamma^T\Gamma w) \rightarrow \Omega(w)$$

9. [b]

Comparing $\frac{1}{N}\sum_{n=1}^{N}(w^Tx_n - y_n)^2 + \frac{\lambda}{N}\sum_{i=0}^{d}\beta_i w_i^2$ with $\frac{1}{N+K}\left(\sum_{n=1}^{N}(w^Tx_n - y_n)^2 + \sum_{n=1}^{K}(w^T\tilde{x}_k - \tilde{y}_k)^2\right)$

$\frac{1}{N}\|Xw - y\|^2 + \frac{\lambda}{N}\|\sqrt{B}w\|^2$ $\qquad$ $\frac{1}{N+K}\left(\|Xw - y\|^2 + \|\tilde{X}w - \tilde{y}\|^2\right)$

$$\lambda\|\sqrt{B}w\|^2 = \|\tilde{X}w - \tilde{y}\|^2 \Rightarrow \tilde{y} = 0$$

$$\lambda\|\sqrt{B}w\|^2 = \|\tilde{X}w\|^2$$

$$\tilde{X} = \sqrt{\lambda}\cdot\sqrt{B}$$

10. [e]

∴ Majority always choose the class with more instance

there are same number of positive and negative examples

When doing E_{loocv} (Majority), the one that is picken up would let the number of some sign element become $N-1$, which cause the validation always false

11. [c]

The error will only occur between the positive and negative elements that are the nearest to 0 when they are picken up. ∴ the upper bound is $\frac{2}{N}$.

12. [e]

	E (linear)	E (constant)
$\sqrt{4+9\sqrt{6}}$	$\frac{1}{3}[(5.765)^2 + (1.481)^2 + 2^2] = 11.34$	$\frac{1}{3}[(1)^2 + (2)^2 + (1)^2] = 2$
$\sqrt{16+81\sqrt{6}}$	$\frac{1}{3}[(1.031)^2 + (1.68)^2 + 2^2] = 1.84$	2
$\sqrt{9+4\sqrt{6}}$	$\frac{1}{3}[(8.984)^2 + (11.682)^2 + 2^2] = 91.13$	2
$\sqrt{36+16\sqrt{6}}$	$\frac{1}{3}[(2.116)^2 + (1.028)^2 + 2^2] = 3.18$	2
$\sqrt{81+36\sqrt{6}}$	$\frac{1}{3}[(1.199)^2 + (0.75)^2 + 2^2] = 2.00$	2

14. [c]

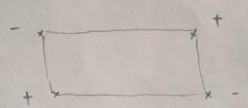
$$\text{Ho} = 1 - \frac{m_H(N)}{N} = 1 - \frac{14}{16} = \frac{2}{16}$$

14. [c]

Probability of not fitting data $= 1 - \frac{m_H(N)}{2^N} = 1 - \frac{14}{16} = \frac{2}{16}$

Every time the situation occurs, there will be one point false

$\therefore$ Expectation $= \frac{1}{4} \times \frac{2}{16} = \frac{2}{64}$ //



15. [a]

$$p\epsilon_+ + (1-p)\epsilon_- = 1-p$$

$$p\epsilon_+ + \epsilon_- - p\epsilon_- + p = 1$$

$$p(\epsilon_+ - \epsilon_- + 1) = 1 - \epsilon_-$$

$$p = \frac{1-\epsilon_-}{\epsilon_+ - \epsilon_- + 1}$$ //

16

expand.py -> convert the input into a correct format

```python
import numpy as np
import random

x_in = []
d = 6

with open('hw4_train.dat') as f1:
    with open('hw4_train_processed.dat', "w") as f2:
        line = f1.readlines()
        for l in line:
            x_in = np.fromstring(l, dtype =float, sep='    ')
            p=2
            f2.write(str(x_in[d])+"    1:1.0"+"    ")
            for i in range(0, d):
                f2.write(str(p)+":"+str(x_in[i])+"    ")
                p += 1
            for i in range(0, d):
                for j in range(i, d):
                    f2.write(str(p)+":"+str(x_in[i]*x_in[j]))
                    f2.write("    ")
                    p += 1

            f2.write("\n")

x_in = []
d = 6

with open('hw4_test.dat') as f1:
        with open('hw4_test_processed.dat', "w") as f2:
                line = f1.readlines()
                for l in line:
                        x_in = np.fromstring(l, dtype =float, sep='    ')
                        p=2
                        f2.write(str(x_in[d])+"    1:1.0"+"    ")
                        for i in range(0, d):
                                f2.write(str(p)+":"+str(x_in[i])+"    ")
                                p += 1
                        for i in range(0, d):
                                for j in range(i, d):
                                        f2.write(str(p)+":"+str(x_in[i]*x_in[j]))
                                        f2.write("    ")
                                        p+=1
                        f2.write("\n")
```

```
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ python expand.py
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ make -f Makefile16
./train -s 0 -c 5000 -e 0.000001 -q hw4_train_processed.dat 16a.model
./train -s 0 -c 50 -e 0.000001 -q hw4_train_processed.dat 16b.model
./train -s 0 -c 0.5 -e 0.000001 -q hw4_train_processed.dat 16c.model
./train -s 0 -c 0.005 -e 0.000001 -q hw4_train_processed.dat 16d.model
./train -s 0 -c 0.00005 -e 0.000001 -q hw4_train_processed.dat 16e.model
./predict -b 0 hw4_test_processed.dat 16a.model 16a.output
Accuracy = 86.6667% (260/300)
./predict -b 0 hw4_test_processed.dat 16b.model 16b.output
Accuracy = 87% (261/300)
./predict -b 0 hw4_test_processed.dat 16c.model 16c.output
Accuracy = 80.6667% (242/300)
./predict -b 0 hw4_test_processed.dat 16d.model 16d.output
Accuracy = 74.3333% (223/300)
./predict -b 0 hw4_test_processed.dat 16e.model 16e.output
Accuracy = 51.6667% (155/300)
```

17.

```
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ make -f Makefile17
./train -s 0 -c 5000 -e 0.000001 -q hw4_train_processed.dat 17a.model
./train -s 0 -c 50 -e 0.000001 -q hw4_train_processed.dat 17b.model
./train -s 0 -c 0.5 -e 0.000001 -q hw4_train_processed.dat 17c.model
./train -s 0 -c 0.005 -e 0.000001 -q hw4_train_processed.dat 17d.model
./train -s 0 -c 0.00005 -e 0.000001 -q hw4_train_processed.dat 17e.model
./predict -b 0 hw4_train_processed.dat 17a.model 17a.output
Accuracy = 91% (182/200)
./predict -b 0 hw4_train_processed.dat 17b.model 17b.output
Accuracy = 90% (180/200)
./predict -b 0 hw4_train_processed.dat 17c.model 17c.output
Accuracy = 87% (174/200)
./predict -b 0 hw4_train_processed.dat 17d.model 17d.output
Accuracy = 80.5% (161/200)
./predict -b 0 hw4_train_processed.dat 17e.model 17e.output
Accuracy = 46.5% (93/200)
```

18.

split_data.py -> separate D to training and validation parts

```python
import numpy as np
import random

x_in = []
#k = 1
d = 6
data_no = 0
with open('hw4_train.dat') as f1:
    with open('hw4_train_splitTest.dat', "w") as f3:
        with open('hw4_train_splitTrain.dat', "w") as f2:
            line = f1.readlines()
            for l in line:
                x_in = np.fromstring(l, dtype =float, sep='    ')
                if (data_no < 120):
                    p=2
                    f2.write(str(x_in[d])+"    1:1.0"+"    ")
                    for i in range(0, d):
                        f2.write(str(p)+":"+str(x_in[i])+"    ")
                        p += 1
                    for i in range(0, d):
                        for j in range(i, d):
                            f2.write(str(p)+":"+str(x_in[i]*x_in[j]))
                            f2.write("    ")
                            p += 1

                    f2.write("\n")
                else:
                    p=2
                    f3.write(str(x_in[d])+"    1:1.0"+"    ")
                    for i in range(0, d):
                        f3.write(str(p)+":"+str(x_in[i])+"    ")
                        p += 1
                    for i in range(0, d):
                        for j in range(i, d):
                            f3.write(str(p)+":"+str(x_in[i]*x_in[j]))
                            f3.write("    ")
                            p += 1

                    f3.write("\n")
    data_no += 1
```

```
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ python split_data.py
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ make -f Makefile18
./train -s 0 -c 5000 -e 0.000001 -q hw4_train_splitTrain.dat 18a.model
./train -s 0 -c 50 -e 0.000001 -q hw4_train_splitTrain.dat 18b.model
./train -s 0 -c 0.5 -e 0.000001 -q hw4_train_splitTrain.dat 18c.model
./train -s 0 -c 0.005 -e 0.000001 -q hw4_train_splitTrain.dat 18d.model
./train -s 0 -c 0.00005 -e 0.000001 -q hw4_train_splitTrain.dat 18e.model
./predict -b 0 hw4_train_splitTest.dat 18a.model 18a.output
Accuracy = 80% (64/80)
./predict -b 0 hw4_train_splitTest.dat 18b.model 18b.output
Accuracy = 86.25% (69/80)
./predict -b 0 hw4_train_splitTest.dat 18c.model 18c.output
Accuracy = 76.25% (61/80)
./predict -b 0 hw4_train_splitTest.dat 18d.model 18d.output
Accuracy = 73.75% (59/80)
./predict -b 0 hw4_train_splitTest.dat 18e.model 18e.output
Accuracy = 42.5% (34/80)
./predict -b 0 hw4_test_processed.dat 18b.model 18.output
Accuracy = 85.6667% (257/300)
```

19.

```
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ make -f Makefile19
./train -s 0 -c 50 -e 0.000001 -q hw4_train_processed.dat 19b.model
./predict -b 0 hw4_test_processed.dat 19b.model 19.output
Accuracy = 87% (261/300)
```

20.

Ecv_cal.py -> calculate the Ecv result from the predict results

```python
import numpy as np
import sys

ecv = 0
with open(sys.argv[1]) as f1:
    line = f1.readlines()
    for l in line:
        percent = l.split(" ")[2]
        ecv += 1 - 0.01*float(percent[:-1])

    ecv /= 5
    print("Ecv = "+str(ecv))

```

```
steven@steven-VirtualBox:~/Downloads/liblinear-2.42$ make -f Makefile20
cat hw4_train_fold2.dat hw4_train_fold3.dat hw4_train_fold4.dat hw4_train_fold5.dat > fold2345.dat
cat hw4_train_fold1.dat hw4_train_fold3.dat hw4_train_fold4.dat hw4_train_fold5.dat > fold1345.dat
cat hw4_train_fold1.dat hw4_train_fold2.dat hw4_train_fold4.dat hw4_train_fold5.dat > fold1245.dat
cat hw4_train_fold1.dat hw4_train_fold2.dat hw4_train_fold3.dat hw4_train_fold5.dat > fold1235.dat
cat hw4_train_fold1.dat hw4_train_fold2.dat hw4_train_fold3.dat hw4_train_fold4.dat > fold1234.dat
./train -s 0 -c 5000 -e 0.000001 -q fold2345.dat 2345-4.model
./train -s 0 -c 5000 -e 0.000001 -q fold1345.dat 1345-4.model
./train -s 0 -c 5000 -e 0.000001 -q fold1245.dat 1245-4.model
./train -s 0 -c 5000 -e 0.000001 -q fold1235.dat 1235-4.model
./train -s 0 -c 5000 -e 0.000001 -q fold1234.dat 1234-4.model
./train -s 0 -c 50 -e 0.000001 -q fold2345.dat 2345-2.model
./train -s 0 -c 50 -e 0.000001 -q fold1345.dat 1345-2.model
./train -s 0 -c 50 -e 0.000001 -q fold1245.dat 1245-2.model
./train -s 0 -c 50 -e 0.000001 -q fold1235.dat 1235-2.model
./train -s 0 -c 50 -e 0.000001 -q fold1234.dat 1234-2.model
./train -s 0 -c 0.5 -e 0.000001 -q fold2345.dat 2345-0.model
./train -s 0 -c 0.5 -e 0.000001 -q fold1345.dat 1345-0.model
./train -s 0 -c 0.5 -e 0.000001 -q fold1245.dat 1245-0.model
./train -s 0 -c 0.5 -e 0.000001 -q fold1235.dat 1235-0.model
./train -s 0 -c 0.5 -e 0.000001 -q fold1234.dat 1234-0.model
./train -s 0 -c 0.005 -e 0.000001 -q fold2345.dat 2345+2.model
./train -s 0 -c 0.005 -e 0.000001 -q fold1345.dat 1345+2.model
./train -s 0 -c 0.005 -e 0.000001 -q fold1245.dat 1245+2.model
./train -s 0 -c 0.005 -e 0.000001 -q fold1235.dat 1235+2.model
./train -s 0 -c 0.005 -e 0.000001 -q fold1234.dat 1234+2.model
./train -s 0 -c 0.00005 -e 0.000001 -q fold2345.dat 2345+4.model
./train -s 0 -c 0.00005 -e 0.000001 -q fold1345.dat 1345+4.model
./train -s 0 -c 0.00005 -e 0.000001 -q fold1245.dat 1245+4.model
./train -s 0 -c 0.00005 -e 0.000001 -q fold1235.dat 1235+4.model
./train -s 0 -c 0.00005 -e 0.000001 -q fold1234.dat 1234+4.model
```

```
echo "lambda = -4";
lambda = -4
./predict -b 0 hw4_train_fold1.dat 2345-4.model 20.output > 1.txt
./predict -b 0 hw4_train_fold2.dat 1345-4.model 20.output > 2.txt
./predict -b 0 hw4_train_fold3.dat 1245-4.model 20.output > 3.txt
./predict -b 0 hw4_train_fold4.dat 1235-4.model 20.output > 4.txt
./predict -b 0 hw4_train_fold5.dat 1234-4.model 20.output > 5.txt
cat 1.txt 2.txt 3.txt 4.txt 5.txt > 20-4.txt
python Ecv_cal.py 20-4.txt
Ecv = 0.14499999999999996
echo "lambda = -2";
lambda = -2
./predict -b 0 hw4_train_fold1.dat 2345-2.model 20.output > 1.txt
./predict -b 0 hw4_train_fold2.dat 1345-2.model 20.output > 2.txt
./predict -b 0 hw4_train_fold3.dat 1245-2.model 20.output > 3.txt
./predict -b 0 hw4_train_fold4.dat 1235-2.model 20.output > 4.txt
./predict -b 0 hw4_train_fold5.dat 1234-2.model 20.output > 5.txt
cat 1.txt 2.txt 3.txt 4.txt 5.txt > 20-2.txt
python Ecv_cal.py 20-2.txt
Ecv = 0.11999999999999997
echo "lambda = 0";
lambda = 0
./predict -b 0 hw4_train_fold1.dat 2345-0.model 20.output > 1.txt
./predict -b 0 hw4_train_fold2.dat 1345-0.model 20.output > 2.txt
./predict -b 0 hw4_train_fold3.dat 1245-0.model 20.output > 3.txt
./predict -b 0 hw4_train_fold4.dat 1235-0.model 20.output > 4.txt
./predict -b 0 hw4_train_fold5.dat 1234-0.model 20.output > 5.txt
cat 1.txt 2.txt 3.txt 4.txt 5.txt > 20-0.txt
python Ecv_cal.py 20-0.txt
Ecv = 0.15499999999999997
```

```
echo "lambda = 2";
lambda = 2
./predict -b 0 hw4_train_fold1.dat 2345+2.model 20.output > 1.txt
./predict -b 0 hw4_train_fold2.dat 1345+2.model 20.output > 2.txt
./predict -b 0 hw4_train_fold3.dat 1245+2.model 20.output > 3.txt
./predict -b 0 hw4_train_fold4.dat 1235+2.model 20.output > 4.txt
./predict -b 0 hw4_train_fold5.dat 1234+2.model 20.output > 5.txt
cat 1.txt 2.txt 3.txt 4.txt 5.txt > 20+2.txt
python Ecv_cal.py 20+2.txt
Ecv = 0.18
echo "lambda = 4";
lambda = 4
./predict -b 0 hw4_train_fold1.dat 2345+4.model 20.output > 1.txt
./predict -b 0 hw4_train_fold2.dat 1345+4.model 20.output > 2.txt
./predict -b 0 hw4_train_fold3.dat 1245+4.model 20.output > 3.txt
./predict -b 0 hw4_train_fold4.dat 1235+4.model 20.output > 4.txt
./predict -b 0 hw4_train_fold5.dat 1234+4.model 20.output > 5.txt
cat 1.txt 2.txt 3.txt 4.txt 5.txt > 20+4.txt
python Ecv_cal.py 20+4.txt
Ecv = 0.5199999999999999
rm 1.txt 2.txt 3.txt 4.txt 5.txt
```