1. c

$$\int_0^2 (wx - e^x)^2 \cdot \frac{1}{2}\,dx$$

$$= \frac{1}{2}\left[\int_0^2 (wx)^2 dx - 2\int_0^2 wxe^x dx + \int_0^2 e^{2x} dx\right]$$

$$= \frac{4}{3}w^2 - (e^2+1)w + \frac{e^4-1}{4}$$

$$\frac{\partial}{\partial w}\left(\frac{4}{3}w^2 - (e^2+1)w + \frac{e^4+1}{4}\right) = 0$$

$$w = \frac{3}{8}(e^2+1) \#$$

3. d.

$$X_h^T X_h = X^T X + \tilde{X}^T \tilde{X} = X^T X + X^T X + \varepsilon^T X + X^T \varepsilon + \varepsilon^T \varepsilon$$

$$E[X_h^T X_h] = 2X^T X + E[\varepsilon^T \varepsilon]$$

$$= 2X^T X + N\sigma^2 I \#$$

$$E[\varepsilon^T \varepsilon] = E\left(\begin{bmatrix} \varepsilon_1 & \cdots & \varepsilon_N \end{bmatrix}\begin{bmatrix} - \varepsilon_1^T - \\ \vdots \\ - \varepsilon_N^T - \end{bmatrix}\right)$$

$$= E[\varepsilon_1^T \varepsilon_1] \cdots E[\varepsilon_N^T \varepsilon_N]$$

4. e

$$E[X_h^T y] = E[X^T y + \tilde{X}^T y]$$

$$= E[X^T y] + E[\tilde{X}^T y]$$

$$= 2X^T y + E[\varepsilon^T y]$$

$$\because E[\varepsilon^T y] = 0$$

$$\therefore E[X_h^T y] = 2X^T y \#$$

2. b.

E_{in} 不會比 E_{out} 大.

5. d.

$$\nabla E(W_{lin}) = \frac{2}{N}(Z^T Z W - Z^T y) = 0$$

$$W_{lin} = (Z^T Z)^{-1} Z^T y$$

$$\nabla E(W_{reg}) = \frac{2}{N}(Z^T Z W - Z^T y) + \frac{2\lambda}{N} W = 0$$

$$W_{reg} = (Z^T Z + \lambda I)^{-1} Z y$$

$$Z = XQ, \quad X^T X = Q \Gamma Q^T, \quad Q^T Q = Q Q^T = I.$$

$$W_{lin} = ((XQ)^T XQ)(XQ)^T y$$

$$= (Q^T X^T X Q)(XQ)^T y$$

$$= (Q^T Q \Gamma Q^T Q)(XQ)^T y$$

$$= \Gamma Q^T X^T y$$

$$W_{reg} = (\Gamma + \lambda I)^{-1} Q^T X^T y$$

$$\frac{u_i}{v_i} = \frac{\frac{1}{\gamma_i + \lambda}}{\frac{1}{\gamma_i}} = \frac{\gamma_i}{\gamma_i + \lambda}$$ #

6. a.

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

$$w = \frac{X^T y}{X^T X + \lambda}$$ #

8. b.

$\Phi(x) = \Gamma^{-1} x$

$\tilde{w}^T \Phi(x) = w^T x$

$\rightarrow \tilde{w}^T \Gamma^{-1} x = w^T x$

$w^T = \tilde{w}^T \Gamma^{-1}$

$\tilde{w}^T = w^T \Gamma$

$\Omega(w) = \tilde{w}^T \tilde{w} = w^T \Gamma (w^T \Gamma)^T = w^T \Gamma^2 w$ #

9. b.

$E_{in1} = \frac{1}{N} \Sigma (w^T x_n - y_n)^2 + \frac{\lambda}{N} \Sigma \beta w^2$

$\nabla E_{in1} = \frac{2}{N} (X^T X w - X^T y) + \frac{2\lambda}{N} \beta w = 0$

$w = (X^T X + \lambda \beta)^{-1} X^T y$

$E_{in2} = \frac{1}{N+K} \left(\Sigma (w^T x - y)^2 + \Sigma (w^T \tilde{x} - \tilde{y})^2 \right)$

$\nabla E_{in2} = (X^T X w - X^T y) + (\tilde{X}^T \tilde{X} w - \tilde{X}^T \tilde{y}) = 0.$

$w = (X^T X + \tilde{X}^T \tilde{X})^{-1} (X^T y + \tilde{X}^T \tilde{y})$

$\lambda \beta = \tilde{X}^T \tilde{X}$. $\tilde{X} = \sqrt{\lambda \beta}$ #

$\tilde{x}^T \tilde{y} = 0$. $\tilde{y} = 0$ #

10. e

2N examples

$E_{loocv} = \frac{1}{2N} \sum_{n=1}^{2N} \cdot 1$

$= \frac{1}{2N} \cdot 2N$

$= 1$ #

11.

O O X X

1 2 3 4

移掉 2 or 3 $\Rightarrow e_n = 1$

移掉 1 or 4 $\Rightarrow e_n = 0$. $E_{loocv} = \frac{2}{N}$ #

12.

$h = w_0$

$(3,0)$ $e_n = 1$

$(\rho,2)$ $e_n = 1$

$(-3,0)$ $e_n = 4$

$E_{loocv} = 2$.

$h = w_0 + w_1 x$

$(3,0)$ $e_n = (3w_1 + w_0)^2$

$(\rho,2)$ $e_n = 4$

$(-3,0)$ $e_n = (-3w_1' + w_0')^2$

$(-3w_1' + w_0')^2 + (3w_1 + w_0)^2 = 2$

$3w_1 + w_0 = 0$
$\rho w_1 + w_0 = 2$ $w_1 = \frac{2}{\rho+3}$, $w_0 = \frac{6}{\rho+3}$

$-3w_1' + w_0' = 0$
$\rho w_1' + w_0' = 0$ $w_1' = \frac{2}{\rho-3}$, $w_0' = \frac{-6}{\rho-3}$

$(\frac{-12}{\rho-3})^2 + (\frac{12}{\rho+3})^2 = 2$.

$\rho = \sqrt{81 + 36\sqrt{6}}$ #

15. a.

$$p \cdot E_+ + (1-p) E_- = 1-p$$

$$E_+ p + E_- - E_- p = 1-p$$

$$(E_+ - E_- + 1) p = 1 - E_-$$

$$p = \frac{1-E_-}{E_+ - E_- + 1} \#$$

7. d.

$$E_{in} = \frac{1}{N} \Sigma (y-y_n)^2 + \frac{2k}{N} \Omega(y)$$

$$\nabla E_{in} = \frac{2}{N} \Sigma (y-y_n) + \frac{2k}{N} \nabla \Omega(y) = 0$$

由選項推測，令 $\nabla \Omega(y) = 2y + b$

$$\nabla E_{in} = \frac{2}{N} (Ny - \Sigma y_n) + \frac{2k}{N} (2y+b) = 0$$

$$\begin{cases} 2y - 2\dfrac{\Sigma y_n}{N} = -\dfrac{2k}{N}(2y+b) = 0 \\ y = \dfrac{\Sigma y_n + k}{N + 2k} \end{cases}$$

$b = -1$

$\because \nabla \Omega(y) = 2y - 1$

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

13. d.

16 – 20

b a e d a

Function

```
def split_data_label(data):
    label = data[:, -1]
    data = data[:, :-1]
    return data, label


def split_train_val(data, train_percentage):
    num = int(len(data)*train_percentage)
    train_data = data[:num]
    val_data = data[num:]
    return train_data, val_data


def transform_data(data, d):
    new_data = []
    for i in range(len(data)):
        tmp = []
        tmp.append(1)
        for j in range(len(data[i])):
            tmp.append(data[i][j])
        for j in range(len(data[i])):
            num = d - j
            for m in range(num):
                xx = data[i][j]*data[i][j+m]
                tmp.append(xx)
        new_data.append(tmp)
    new_data = np.array(new_data)
    return new_data
```

Read data

```
all_train_data = np.loadtxt("./hw4_train.dat.txt")
train_data, train_label = split_data_label(all_train_da
ta)
train_data = transform_data(train_data, 6)

all_test_data = np.loadtxt("./hw4_test.dat.txt")
```

```python
test_data, test_label = split_data_label(all_test_data)
test_data = transform_data(test_data, 6)
```

```python
# 16
lambdas = [0.0001, 0.01, 1, 100, 10000]
best_acc = 0.0
best_lambda = 0
for lambda_ in lambdas:
    C = 1 / (2*lambda_)
    m = train(train_label, train_data, "-s 0 -c %f -
e 0.000001" % C)
    p_label, p_acc, p_val = predict(test_label, test_da
ta, m)
    ACC, MSE, SCC = evaluations(test_label, p_label)
    if ACC >= best_acc:
        best_acc = ACC
        best_lambda = 1 / (2*C)

print("best_lambda:", best_lambda)
```

```python
# 17
lambdas = [0.0001, 0.01, 1, 100, 10000]
best_acc = 0.0
best_lambda = 0
for lambda_ in lambdas:
    C = 1 / (2*lambda_)
    m = train(train_label, train_data, "-s 0 -c %f -
e 0.000001" % C)
    p_label, p_acc, p_val = predict(train_label, train_
data, m)
    ACC, MSE, SCC = evaluations(train_label, p_label)
    if ACC >= best_acc:
        best_acc = ACC
        best_lambda = 1 / (2*C)

print("best_lambda:", best_lambda)
```

```python
# 18, 19
```

```
train_data, val_data = split_train_val(train_data, 0.6)
train_label, val_label = split_train_val(train_label, 0.6)
lambdas = [0.0001, 0.01, 1, 100, 10000]
best_acc = 0.0
best_lambda = 0
for lambda_ in lambdas:
    C = 1 / (2*lambda_)
    m = train(train_label, train_data, "-s 0 -c %f -e 0.000001" % C)
    p_label, p_acc, p_val = predict(val_label, val_data, m)
    ACC, MSE, SCC = evaluations(val_label, p_label)
    if ACC >= best_acc:
        best_acc = ACC
        best_lambda = 1 / (2*C)

C = 1 / (2*best_lambda)
m = train(train_label, train_data, "-s 0 -c %f -e 0.000001" % C)
p_label, p_acc, p_val = predict(test_label, test_data, m)
ACC, MSE, SCC = evaluations(test_label, p_label)
print("18 best_E:", 100 - ACC)

train_data = np.concatenate([train_data, val_data])
train_label = np.concatenate([train_label, val_label])
C = 1 / (2*best_lambda)
m = train(train_label, train_data, "-s 0 -c %f -e 0.000001" % C)
p_label, p_acc, p_val = predict(test_label, test_data, m)
ACC, MSE, SCC = evaluations(test_label, p_label)

print("19 best_E:", 100 - ACC)
```

```
num = int(len(all_train_data) / 5)
all_acc = []
```

```
for i in range(5):
    val_data = train_data[num*i:num*(i+1)]
    val_label = train_label[num*i:num*(i+1)]
    nums = np.arange(num*i, num*(i+1))
    data = train_data
    label = train_label
    data = np.delete(data, nums, 0)
    label = np.delete(label, nums, 0)
    lambdas = [0.0001, 0.01, 1, 100, 10000]
    best_acc = 0.0
    best_lambda = 0
    for lambda_ in lambdas:
        C = 1 / (2*lambda_)
        m = train(label, data, "-s 0 -c %f -
e 0.000001" % C)
        p_label, p_acc, p_val = predict(val_label, val_
data, m)
        ACC, MSE, SCC = evaluations(val_label, p_label)
        if ACC >= best_acc:
            best_acc = ACC
            best_lambda = 1 / (2*C)
    all_acc.append(best_acc)

best_E = (100 - np.mean(all_acc)) / 100
print(best_E)
print(all_acc)
```