

大家好，这篇是有关台大机器学习课程作业四的详解，题目同Coursera。

我的github地址：

<https://github.com/Doraemonzzz>

个人主页：

<http://doraemonzzz.com/>

作业地址：

<https://www.csie.ntu.edu.tw/~htlin/course/ml15fall/>

参考资料：

<https://blog.csdn.net/a1015553840/article/details/51085129>

<http://www.vynguyen.net/category/study/machine-learning/page/6/>

<http://book.caltech.edu/bookforum/index.php>

<http://beader.me/mlnotebook/>

Problem 1

我的理解是deterministic noise是由 \mathcal{H} 和 f “复杂度之差”产生的，所以如果选择使用 \mathcal{H}' ，那么“复杂度之差”会增加，所以deterministic noise会增加。

Problem 2

这题比较简单， $\mathcal{H}(Q, 0, Q_0) = \sum_{q=0}^{Q_0-1} w_q L_q(x) = \mathcal{H}(Q_0 - 1)$ ，所以

$$\mathcal{H}(10, 0, 3) \cap \mathcal{H}(10, 0, 4) = \mathcal{H}(2) \cap \mathcal{H}(3) = \mathcal{H}(2)$$

Problem 3

首先计算 $E_{\text{in}}(w)$, $E_{\text{aug}}(w)$

$$\begin{aligned} E_{\text{in}}(w) &= \|Xw - y\|^2 \\ &= (Xw - y)^T (Xw - y) \\ &= w^T X^T X w - 2y^T Xw + y^T y \end{aligned}$$

$$\begin{aligned} E_{\text{aug}}(w) &= E_{\text{in}}(w) + \frac{\lambda}{N} w^T w \\ &= w^T X^T X w - 2y^T Xw + y^T y + \frac{\lambda}{N} w^T w \end{aligned}$$

接着关于 w 求梯度

$$\begin{aligned}\nabla E_{\text{in}}(w) &= 2X^T Xw - 2X^T y \\ \nabla E_{\text{aug}}(w) &= \nabla E_{\text{in}}(w) + 2\frac{\lambda}{N}w = 2X^T Xw - 2X^T y + 2\frac{\lambda}{N}w\end{aligned}$$

由梯度下降法可知更新规则为

$$\begin{aligned}w(t+1) &= w(t) - \eta \nabla E_{\text{aug}}(w(t)) \\ &= w(t) - \eta(\nabla E_{\text{in}}(w(t)) + 2\frac{\lambda}{N}w(t)) \\ &= (1 - 2\eta\frac{\lambda}{N})w(t) - \eta \nabla E_{\text{in}}(w)\end{aligned}$$

所以

$$\alpha = 1 - 2\eta\frac{\lambda}{N}, \beta = -\eta$$

Problem 4

首先回顾 w_{reg} 的公式, 记 $u = Z^T y$

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

假设 $Z \in \mathbb{R}^{N \times M}$, 所以 $Z^T Z \in \mathbb{R}^{M \times M}$ 。

由于 $Z^T Z$ 为半正定对称矩阵, 所以 $Z^T Z$ 正交相似于对角阵, 且特征值非负, 令记 P 为正交相似矩阵, $Z^T Z$ 的特征值为 $k_1, \dots, k_M (k_i \geq 0)$, 所以

$$P^T Z^T Z P = \text{diag}\{k_1, k_2, \dots, k_M\}$$

从而

$$\begin{aligned}P^T (Z^T Z + \lambda I) P &= \text{diag}\{k_1 + \lambda, k_2 + \lambda, \dots, k_M + \lambda\} \\ (Z^T Z + \lambda I) &= P \text{diag}\{k_1 + \lambda, k_2 + \lambda, \dots, k_M + \lambda\} P^T \\ (Z^T Z + \lambda I)^{-1} &= P^T \text{diag}\{(k_1 + \lambda)^{-1}, (k_2 + \lambda)^{-1}, \dots, (k_M + \lambda)^{-1}\} P \\ (Z^T Z + \lambda I)^{-2} &= P^T \text{diag}\{(k_1 + \lambda)^{-2}, (k_2 + \lambda)^{-2}, \dots, (k_M + \lambda)^{-2}\} P\end{aligned}$$

带入 $w_{\text{reg}}^T w_{\text{reg}}$ 的定义可得

$$w_{\text{reg}}^T w_{\text{reg}} = u^T P^T \text{diag}\{(k_1 + \lambda)^{-2}, (k_2 + \lambda)^{-2}, \dots, (k_M + \lambda)^{-2}\} P u$$

记 $v = Pu = (v_1, \dots, v_M)$, 注意 P, u 为常量, 所以 v 也为常量, 从而

$$w_{\text{reg}}^T w_{\text{reg}} = v^T \text{diag}\{(k_1 + \lambda)^{-2}, (k_2 + \lambda)^{-2}, \dots, (k_M + \lambda)^{-2}\} v = \sum_{i=1}^M (k_i + \lambda)^{-2} v_i^2$$

因为 $k_i \geq 0$, 所以 $w_{\text{reg}}^T w_{\text{reg}} = \|w_{\text{reg}}\|^2$ 关于 $\lambda (\lambda \geq 0)$ 递减, 从而 $\|w_{\text{reg}}\|$ 关于 $\lambda (\lambda \geq 0)$ 递减。

Problem 5

题目的意思是利用平方误差计算leave-one-out cross-validation, 对于此题来说, 有三个点 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$, 首先根据其中两个点 (不妨设为 $(x_1, y_1), (x_2, y_2)$) 训练模型 $h(x)$, 然后计算 $(h(x_3) - y_3)^2$, 下面具体看下题目。

首先看 $h_0(x) = b_0$, 如果有两个点 $(x_1, y_1), (x_2, y_2)$, 那么

$$E_{\text{in}} = (b_0 - y_1)^2 + (b_0 - y_2)^2 = 2b_0^2 - 2(y_1 + y_2)b_0 + (y_1^2 + y_2^2)$$

由二次函数性质可知, 当 $b_0 = \frac{y_1 + y_2}{2}$ 时, E_{in} 最小。

所以如果点为 $(-1, 0), (\rho, 1)$, 那么 $b_0 = \frac{1}{2}$, 从而误差为 $(\frac{1}{2} - 0)^2 = \frac{1}{4}$; 如果点为 $(-1, 0), (1, 0)$, 那么 $b_0 = 0$, 从而误差为 $(1 - 0)^2 = 1$; 如果点为 $(\rho, 1), (1, 0)$, 那么 $b_0 = \frac{1}{2}$, 从而误差为 $(\frac{1}{2} - 0)^2 = \frac{1}{4}$ 。因此总误差为

$$\frac{1}{3} \left(1 + \frac{1}{4} + \frac{1}{4} \right)$$

再来看下 $h_1(x) = a_1 x + b_1$, 如果有两个点 $(x_1, y_1), (x_2, y_2)$, 显然直线过这两点时 E_{in} 最小, 从而

$$\begin{aligned} a_1 x_1 + b_1 &= y_1 \\ a_1 x_2 + b_1 &= y_2 \end{aligned}$$

解得

$$a_1 = \frac{y_1 - y_2}{x_1 - x_2}, b_1 = \frac{x_1 y_2 - x_2 y_1}{x_1 - x_2}$$

所以如果点为 $(-1, 0), (\rho, 1)$, 那么 $a_1 = \frac{1}{1+\rho}, b_1 = \frac{1}{1+\rho}$, 从而误差为 $(\frac{1}{1+\rho} + \frac{1}{1+\rho} - 0)^2 = \frac{4}{(1+\rho)^2}$; 如果点为 $(-1, 0), (1, 0)$, 那么 $a_1 = 0, b_1 = 0$, 从而误差为 $(0 - 1)^2 = 1$; 如果点为 $(\rho, 1), (1, 0)$, 那么 $a_1 = \frac{1}{\rho-1}, b_1 = -\frac{1}{\rho-1}$, 从而误差为 $(-\frac{1}{\rho-1} - \frac{1}{\rho-1} - 0)^2 = \frac{4}{(1-\rho)^2}$ 。因此总误差为

$$\frac{1}{3} \left(\frac{4}{(1+\rho)^2} + 1 + \frac{4}{(1-\rho)^2} \right)$$

由题设可知

$$\begin{aligned}
\frac{1}{3} \left(\frac{4}{(1+\rho)^2} + 1 + \frac{4}{(1-\rho)^2} \right) &= \frac{1}{3} \left(1 + \frac{1}{4} + \frac{1}{4} \right) \\
\frac{4}{(1+\rho)^2} + \frac{4}{(1-\rho)^2} &= \frac{1}{2} \\
8[(1-\rho)^2 + (1+\rho)^2] &= (1-\rho)^2(1+\rho)^2 \\
8(2+2\rho^2) &= (1-\rho^2)^2 \\
16+16\rho^2 &= \rho^4 - 2\rho^2 + 1 \\
\rho^4 - 18\rho^2 - 15 &= 0 \\
\rho^2 &= 9 + 4\sqrt{6} \\
\rho &= \sqrt{9 + 4\sqrt{6}}
\end{aligned}$$

Problem 6

要使得至少有一个人收到的5次预测都是正确的，需要给 $2^5 = 32$ 个人写信，方法如下，第一天告诉一半的人A队胜利，告诉另一半人B队胜利，那么第一天必然有16封信是正确的，对这16个人重复此操作，到第五天肯定有人收到的5封信都是正确的。从这个过程中可以看出在第五场比赛之前，一共需要寄送的数量为

$$2^5 + 2^4 + 2^3 + 2^2 = 4(2^4 - 1) = 60$$

Problem 7

由上题可知，加上第五场以及第六场，一共要寄送

$$2^5 + 2^4 + 2^3 + 2^2 + 2 + 1 = 2^6 - 1 = 63$$

所以一共要花费 $63 \times 10 = 630$ ，如果别人花了1000元，那么一共可以赚 $1000 - 630 = 370$

Problem 8

因为这题 $a(x)$ 是确定的，所以此处只有一个模型， $M = 1$

Problem 9

回顾公式可知

$$\mathbb{P}(|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon) \leq 2Me^{-2\epsilon^2 N}$$

这里 $\epsilon = 0.01$, $M = 1$, $N = 10000$ ，带入可得

$$P \leq 0.271$$

Problem 10

我们获得 $g(x)$ 的过程中实际上参考了 $a(x)$ ，这是一种data snooping，所以为了好的效果，应该同时使用 $g(x), a(x)$ ，根据两个判别函数来做决定。

Problem 11

只要代公式即可，首先做以下记号

$$X = [x_1, \dots, x_N]^T, y = [y_1, \dots, y_N]^T$$

所以

$$\begin{aligned} X' &= [x_1, \dots, x_N, \tilde{x}_1, \dots, \tilde{x}_K]^T = \begin{bmatrix} X \\ \tilde{X} \end{bmatrix} \\ y' &= [y_1, \dots, y_N, \tilde{y}_1, \dots, \tilde{y}_K]^T = \begin{bmatrix} y \\ \tilde{y} \end{bmatrix} \end{aligned}$$

所以原问题可化为

$$\min_w \frac{1}{N+K} \|X' w - y'\|^2$$

所以最优解为

$$\begin{aligned} w &= ((X')^T X')^{-1} (X')^T y' \\ &= \left(\begin{bmatrix} X \\ \tilde{X} \end{bmatrix}^T \begin{bmatrix} X \\ \tilde{X} \end{bmatrix} \right)^{-1} \begin{bmatrix} X \\ \tilde{X} \end{bmatrix}^T \begin{bmatrix} y \\ \tilde{y} \end{bmatrix} \\ &= (X^T X + \tilde{X}^T \tilde{X})^{-1} (X^T y + \tilde{X}^T \tilde{y}) \end{aligned}$$

Problem 12

这题的目地是为了把岭回归转换成一般的线性回归，回顾岭回归的公式

$$w_{\text{reg}} = (X^T X + \lambda I)^{-1} X^T y$$

对比上题的公式，我们令

$$\tilde{X} = \sqrt{\lambda} I, y = 0$$

即可。

Problem 13

这部分只要根据公式计算即可，注意这里是0-1误差

```

# -*- coding: utf-8 -*-
"""
Created on Sat Mar 16 09:53:46 2019

@author: qinzhen
"""

import numpy as np
from numpy.linalg import inv
import matplotlib.pyplot as plt

#### Problem 13
#预处理函数
def preprocess(data):
    """
    数据预处理
    """
    #获取维度
    n, d = data.shape
    #分离x
    X = data[:, :-1]
    #添加偏置项1
    X = np.c_[np.ones(n), X]
    #分离y
    y = data[:, -1]

    return X, y

#计算w
def w(X, y, Lambda):
    """
    计算岭回归的结果
    """
    d = X.shape[1]
    w = inv(X.T.dot(X) + Lambda * np.eye(d)).dot(X.T).dot(y)
    return w

#计算误差
def E(X, y, w):
    return np.mean(np.sign(X.dot(w)) != y)

#读取数据
train = np.genfromtxt('hw4_train.dat')
test = np.genfromtxt('hw4_test.dat')
X_train, y_train = preprocess(train)
X_test, y_test = preprocess(test)

#计算
w1 = w(X_train, y_train, 11.26)
Ein = E(X_train, y_train, w1)
Eout = E(X_test, y_test, w1)
print("Ein = {}".format(Ein))
print("Eout = {}".format(Eout))

```

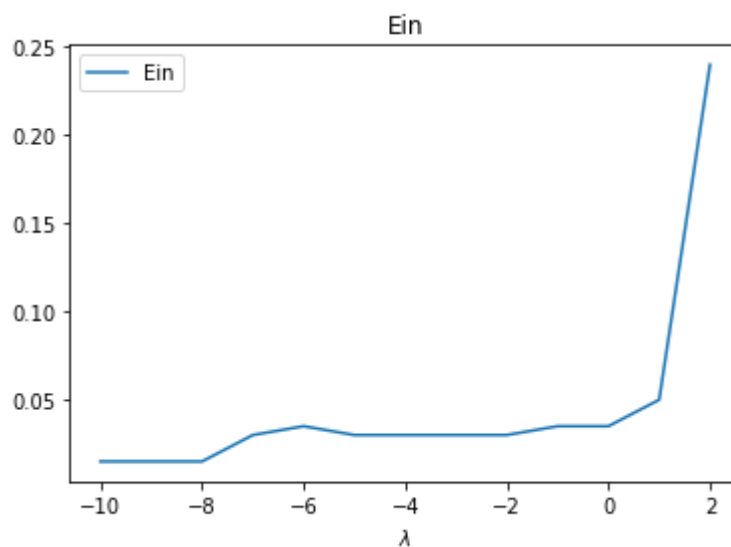
```
Ein = 0.055
Eout = 0.052
```

Problem 14

```
#### Problem 14
Lambda = range(2, -11, -1)
Ein=[]
Eout=[]
for l in Lambda:
    l1 = 10 ** (l)
    w1 = w(X_train, y_train, l1)
    e_in = E(X_train, y_train, w1)
    e_out = E(X_test, y_test, w1)
    Ein.append(e_in)
    Eout.append(e_out)

plt.plot(Lambda, Ein, label='Ein')
plt.xlabel('$\lambda$')
plt.title('Ein')
plt.legend()
plt.show()

Lambda1 = Lambda[np.argmin(Ein)]
Eout1 = Eout[np.argmin(Ein)]
print("最小Ein对应的Lambda为log10({}), 对应的Eout为{}".format(Lambda1, Eout1))
```



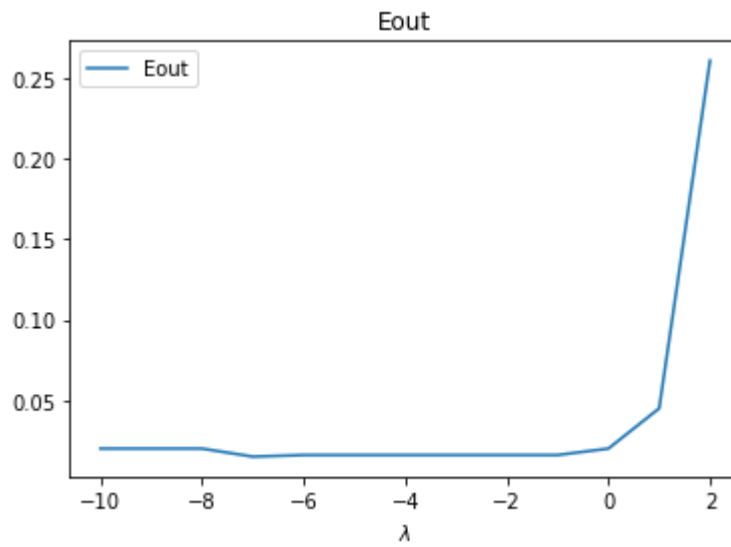
最小Ein对应的Lambda为log10(-8)，对应的Eout为0.02

Problem 15

Problem 15

```
plt.plot(Lambda, Eout, label='Eout')
plt.xlabel('$\lambda$')
plt.title('Eout')
plt.legend()
plt.show()

Lambda2 = Lambda[np.argmin(Eout)]
Eout2 = Eout[np.argmin(Eout)]
print("最小Eout对应的Lambda为log10({}), 对应的Eout为{}".format(Lambda2, Eout2))
```



最小Eout对应的Lambda为log10(-7), 对应的Eout为0.015

Problem 16

这题是将数据拆成训练集和验证集，步骤和上题基本一致，选择 E_{in} 最小的 λ ：

Problem 16

```
X_train_ = X_train[:, :120]
y_train_ = y_train[:, :120]
X_val = X_train[:, 120:]
y_val = y_train[:, 120:]

Etrain = []
Eout = []
Eval = []
for l in Lambda:
    l1 = 10 ** l
    w1 = w(X_train_, y_train_, l1)
    e_train = E(X_train_, y_train_, w1)
    e_out = E(X_test, y_test, w1)
    e_val = E(X_val, y_val, w1)
    Etrain.append(e_train)
```



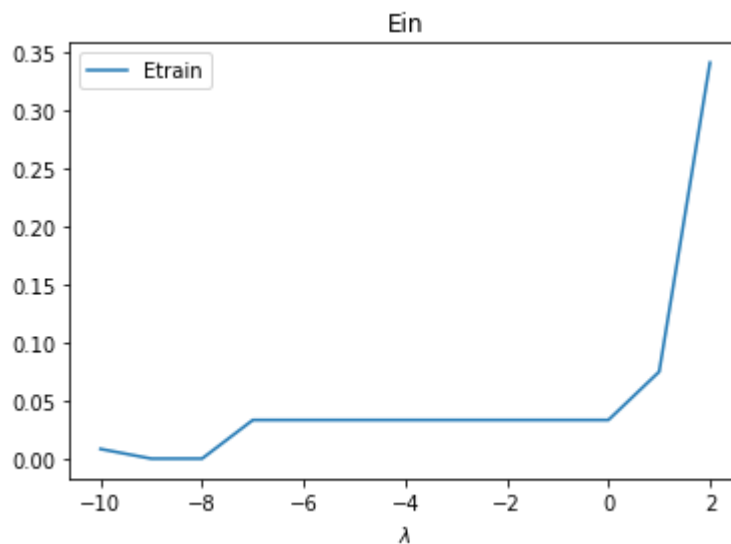
```

Eout.append(e_out)
Eval.append(e_val)

plt.plot(Lambda, Etrain, label='Etrain')
plt.xlabel('$\lambda$')
plt.title('Ein')
plt.legend()
plt.show()

Lambda3 = Lambda[np.argmin(Etrain)]
Eout3 = Eout[np.argmin(Etrain)]
print("最小Etrain对应的Lambda为log10({}), 对应的Eout为{}".format(Lambda3, Eout3))

```



最小Etrain对应的Lambda为log10(-8), 对应的Eout为0.025

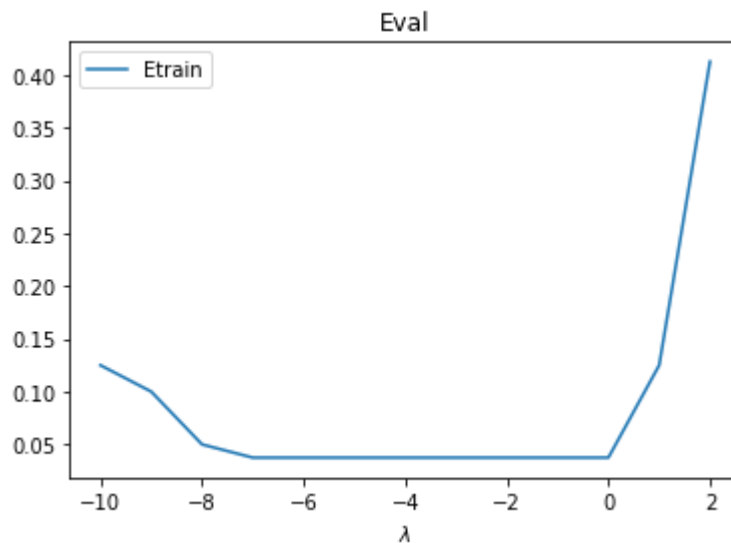
Problem 17

```

#### Problem 17
plt.plot(Lambda, Eval, label='Etrain')
plt.xlabel('$\lambda$')
plt.title('Eval')
plt.legend()
plt.show()

Lambda4 = Lambda[np.argmin(Eval)]
Eout4 = Eout[np.argmin(Eval)]
print("最小Eval对应的Lambda为log10({}), 对应的Eout为{}".format(Lambda4, Eout4))

```



最小Eval对应的Lambda为 $\log_{10}(0)$ ，对应的Eout为0.028

Problem 18

选择最优的 λ 进行训练：

```
#### Problem 18
l = Lambda[np.argmin(Eval)]
l1 = 10 ** l
w1 = w(X_train, y_train, l1)
Ein = E(X_train, y_train, w1)
Eout = E(X_test, y_test, w1)
print("Ein = {}".format(Ein))
print("Eout = {}".format(Eout))
```

```
Ein = 0.035
Eout = 0.02
```

Problem 19

将数据拆成5各部分，计算 E_{cv} ：

```
####Problem 19
#准备数据
K = 5
m = X_train.shape[0] // K
data=[]
#将数据拆分成训练集以及验证集
for i in range(K):
    xtrain = np.r_[X_train[: i*m], X_train[(i+1)*m: ]]
    ytrain = np.r_[y_train[: i*m], y_train[(i+1)*m: ]]
```

```

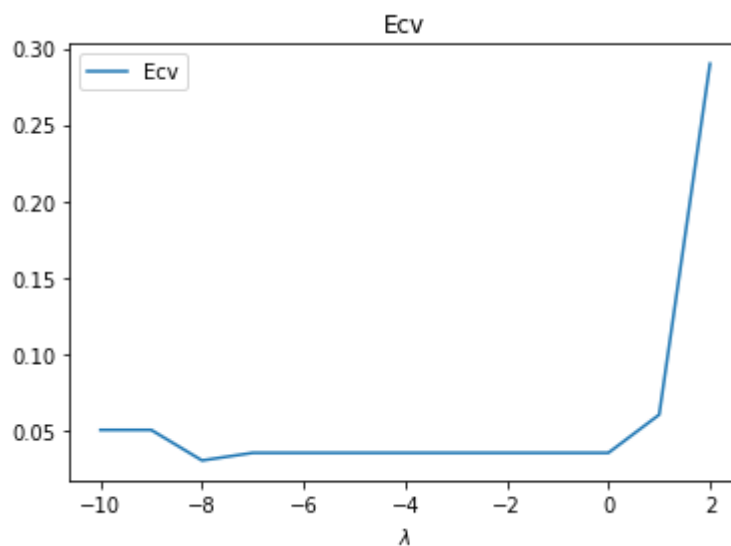
xval = X_train[i*m: (i+1)*m]
yval = y_train[i*m: (i+1)*m]
data.append([Xtrain, ytrain, xval, yval])

Ecv = []
for l in Lambda:
    l1 = 10 ** l
    ecv = 0
    for d in data:
        Xtrain, ytrain, xval, yval = d
        w1 = w(Xtrain, ytrain, l1)
        ecv += E(xval, yval, w1)
    ecv /= K
    Ecv.append(ecv)

plt.plot(Lambda, Ecv, label='Ecv')
plt.xlabel('$\lambda$')
plt.title('Ecv')
plt.legend()
plt.show()

Lambda5 = Lambda[np.argmin(Ecv)]
print("最小Ecv对应的Lambda为log10({})".format(Lambda5))

```



最小Ecv对应的Lambda为log10(-8)

Problem 20

计算 $\log_{10}(\lambda) = -8$ 时的 $E_{\text{in}}, E_{\text{out}}$

```
####Problem20
l1 = 10 ** Lambda5
w1 = w(X_train, y_train, l1)
Ein = E(X_train, y_train, w1)
Eout = E(X_test, y_test, w1)
print("Ein = {}".format(Ein))
print("Eout = {}".format(Eout))
```

```
Ein = 0.015
Eout = 0.02
```

以下两题为附加题

Problem 21

这题是需要最小化

$$E(w) = \|Xw - y\|^2 + \lambda w^T \Gamma^T \Gamma w = w^T X^T X w - 2y^T X w + y^T y + \lambda w^T \Gamma^T \Gamma w$$

求梯度可得

$$\nabla E(w) = 2X^T X w - 2X^T y + 2\lambda \Gamma^T \Gamma w$$

令梯度为0，解得

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

对比11题的公式

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

比较可得

$$\tilde{X} = \sqrt{\lambda} \Gamma, \tilde{y} = 0$$

Problem 22

这题是需要最小化

$$\begin{aligned} E(w) &= \|Xw - y\|^2 + \lambda \|w - w_{\text{hint}}\|^2 \\ &= w^T X^T X w - 2y^T X w + y^T y + \lambda (w - w_{\text{hint}})^T (w - w_{\text{hint}}) \\ &= w^T X^T X w - 2y^T X w + y^T y + \lambda (w^T w - 2w_{\text{hint}}^T w + w_{\text{hint}}^T w_{\text{hint}}) \end{aligned}$$

求梯度可得

$$\nabla E(w) = 2X^T X w - 2X^T y + 2\lambda w - 2\lambda w_{\text{hint}}$$

令梯度为0，解得

$$w = (X^T X + \lambda I)^{-1} (X^T y + \lambda w_{\text{hint}})$$

对比11题的公式

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

比较可得

$$\tilde{X} = \sqrt{\lambda} I, \tilde{y} = \sqrt{\lambda} w_{\text{hint}}$$