HW4

程式Python語言（Jupiter notebook），並使用了LIBLINEAR，如下

import numpy as np

from sklearn.preprocessing import PolynomialFeatures

def preprocess(data):

    #維度

    n, d = data.shape

    #剝離X

    X = data[:, :-1]

    #偏置項1

    #X = np.c\_[np.ones(n), X]

    #剝離y

    y = data[:, -1]

    return X, y

#將網頁上的資料下載到桌面，存儲到同一個目錄下，打開命名

train = np.genfromtxt('hw4\_train.dat.txt')

test = np.genfromtxt('hw4\_test.dat.txt')

X\_train1, y\_train = preprocess(train)

X\_test1, y\_test = preprocess(test)

#X\_test

#y\_test

reg = PolynomialFeatures(degree=3)

X\_test=reg.fit\_transform(X\_test1)

reg = PolynomialFeatures(degree=3)

X\_train=reg.fit\_transform(X\_train1)

X\_test

f = open("hw4\_train.dat.txt","r")   #设置文件对象

for i in range(120):

    print(f.readline().strip())

import liblinear

from liblinear.liblinearutil import \*

import scipy

#Problem 12

#cheating means use data from "Eout"

#c=1/(2\*lamda)

#lamda=0.0001

m=train(y\_test,X\_test,'-s 0 -c 5000 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_test,X\_test,m)

#lamda=0.01

m=train(y\_test,X\_test,'-s 0 -c 50 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_test,X\_test,m)

#lamda=1

m=train(y\_test,X\_test,'-s 0 -c 0.5 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_test,X\_test,m)

#lamda=100

m=train(y\_test,X\_test,'-s 0 -c 0.005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_test,X\_test,m)

#lamda=10000

m=train(y\_test,X\_test,'-s 0 -c 0.00005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_test,X\_test,m)

#Problem 13

#lamda=0.0001

m=train(y\_train,X\_train,'-s 0 -c 5000 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=0.01

m=train(y\_train,X\_train,'-s 0 -c 50 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=1

m=train(y\_train,X\_train,'-s 0 -c 0.5 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=100

m=train(y\_train,X\_train,'-s 0 -c 0.005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=10000

m=train(y\_train,X\_train,'-s 0 -c 0.00005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#Problem 14

#120 train 和 80val

np.genfromtxt('hw4\_train.dat.txt',skip\_header=0, skip\_footer=119)

train1 = np.genfromtxt('hw4\_train.dat1.txt')#前120筆

X\_train, y\_train = preprocess(train1)

#lamda=0.0001

m=train(y\_train,X\_train,'-s 0 -c 5000 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=0.01

m=train(y\_train,X\_train,'-s 0 -c 50 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=1

m=train(y\_train,X\_train,'-s 0 -c 0.5 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=100

m=train(y\_train,X\_train,'-s 0 -c 0.005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=10000

m=train(y\_train,X\_train,'-s 0 -c 0.00005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#120 train 和 80val

np.genfromtxt('hw4\_train.dat.txt',skip\_header=120, skip\_footer=0)

train1 = np.genfromtxt('hw4\_train.dat2.txt')#後80筆

X\_train, y\_train = preprocess(train1)

#lamda=0.0001

m=train(y\_train,X\_train,'-s 0 -c 5000 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=0.01

m=train(y\_train,X\_train,'-s 0 -c 50 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=1

m=train(y\_train,X\_train,'-s 0 -c 0.5 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=100

m=train(y\_train,X\_train,'-s 0 -c 0.005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=10000

m=train(y\_train,X\_train,'-s 0 -c 0.00005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=0.01

m=train(y\_test,X\_test,'-s 0 -c 0.05 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_test,X\_test,m)

#Problem 15

#convert from Dtrain to D

#lamda=0.0001

m=train(y\_train,X\_train,'-s 0 -c 5000 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=0.01

m=train(y\_train,X\_train,'-s 0 -c 50 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=1

m=train(y\_train,X\_train,'-s 0 -c 0.5 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=100

m=train(y\_train,X\_train,'-s 0 -c 0.005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=10000

m=train(y\_train,X\_train,'-s 0 -c 0.00005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#Problem 16

#for lamda=0.0001,0.01,1,100,10000

#40+40+40+40+40=5\*40=200

train1=np.genfromtxt('hw4\_train.dat.txt',skip\_header=0, skip\_footer=160)

X\_train1, y\_train = preprocess(train1)

reg = PolynomialFeatures(degree=3)

X\_train=reg.fit\_transform(X\_train1)

#train2=np.genfromtxt('hw4\_train.dat.txt',skip\_header=40, skip\_footer=120)

#X\_train, y\_train = preprocess(train2)

#train3=np.genfromtxt('hw4\_train.dat.txt',skip\_header=80, skip\_footer=80)

#X\_train, y\_train = preprocess(train3)

#train4=np.genfromtxt('hw4\_train.dat.txt',skip\_header=120, skip\_footer=40)

#X\_train, y\_train = preprocess(train4)

#train5=np.genfromtxt('hw4\_train.dat.txt',skip\_header=160, skip\_footer=0)

#X\_train, y\_train = preprocess(train5)

#lamda=0.0001

m=train(y\_train,X\_train,'-s 0 -c 5000 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=0.01

m=train(y\_train,X\_train,'-s 0 -c 50 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=1

m=train(y\_train,X\_train,'-s 0 -c 0.5 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=100

m=train(y\_train,X\_train,'-s 0 -c 0.005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)

#lamda=10000

m=train(y\_train,X\_train,'-s 0 -c 0.00005 -e 0.000001')

p\_labs,p\_acc,p\_vals=predict(y\_train,X\_train,m)