In [1]:

```
import numpy as np
from liblinearutil import *
```

In [2]:

```
def readfile(filename):
    X = []
    Y = []
    for lines in open(filename).readlines():
        temp = lines.strip().split()
        x = [1]
        for i in range(6):
            x.append(float(temp[i]))
        for i in range(6):
            for j in range(6):
                if i<=j:
                    x.append(float(temp[i])*float(temp[j]))
        X.append(x)
        Y.append(float(temp[-1]))
#     X = np.asarray(X)
#     Y = np.asarray(Y)
    return X, Y
```

In [3]:

```
trainfile = "../../hw4_train.dat.txt"
testfile = "../../hw4_test.dat.txt"
X, Y = readfile(trainfile)
Xt, Yt = readfile(testfile)
```

In [4]:

```
##16, 17
l = [10**(-4), 10**(-2), 10**0, 10**2, 10**4]
max_out_acc = 0
l_out = -1
max_in_acc = 0
l_in = -1
for i in range(len(l)):
    C = 1/(2*l[i])
    param = '-s 0 -c ' + str(C) + ' -e 0.000001'
    m = train(Y, X, param)
    p_out_labels, p_out_acc, p_out_vals = predict(Yt, Xt, m)
    if p_out_acc[0] >= max_out_acc:
        max_out_acc = p_out_acc[0]
        l_out = i
    p_in_labels, p_in_acc, p_in_vals = predict(Y, X, m)
    if p_in_acc[0] >= max_in_acc:
        max_in_acc = p_in_acc[0]
        l_in = i
print("16.: ", np.log10(l[l_out]))
print("17.: ", np.log10(l[l_in]))
```

```
Accuracy = 86.6667% (260/300) (classification)
Accuracy = 91% (182/200) (classification)
Accuracy = 87% (261/300) (classification)
Accuracy = 90% (180/200) (classification)
Accuracy = 80.6667% (242/300) (classification)
Accuracy = 87% (174/200) (classification)
Accuracy = 74.3333% (223/300) (classification)
Accuracy = 80.5% (161/200) (classification)
Accuracy = 51.6667% (155/300) (classification)
Accuracy = 46.5% (93/200) (classification)
16.:  -2.0
17.:  -4.0
```

In [5]:

```
##18
l = [10**(-4), 10**(-2), 10**0, 10**2, 10**4]
X_train = X[:120]
Y_train = Y[:120]
X_val = X[120:]
Y_val = Y[120:]
max_val_acc = 0
l_val = -1
for i in range(len(l)):
    C = 1/(2*l[i])
    param = '-s 0 -c ' + str(C) + ' -e 0.000001'
    m = train(Y_train, X_train, param)
    p_val_labels, p_val_acc, p_val_vals = predict(Y_val, X_val, m)
    if p_val_acc[0] >= max_val_acc:
        max_val_acc = p_val_acc[0]
        l_val = i
        best_m = m
p_labels, p_acc, p_vals = predict(Yt, Xt, best_m)
print("18.: ", (100-p_acc[0])*0.01)
```

```
Accuracy = 80% (64/80) (classification)
Accuracy = 86.25% (69/80) (classification)
Accuracy = 76.25% (61/80) (classification)
Accuracy = 73.75% (59/80) (classification)
Accuracy = 42.5% (34/80) (classification)
Accuracy = 85.6667% (257/300) (classification)
18.:  0.14333333333333328
```

In [6]:

```
##19
##retrain by lambda*
best_C = 1/(2*l[l_val])
param = '-s 0 -c ' + str(best_C) + ' -e 0.000001'
m = train(Y, X, param)
p_labels, p_acc, p_vals = predict(Yt, Xt, m)
print("19.: ", (100-p_acc[0])*0.01)
```

```
Accuracy = 87% (261/300) (classification)
19.:  0.13
```

In [7]:

```
##20
l = [10**(-4), 10**(-2), 10**0, 10**2, 10**4]
min_Ecv = 10
for i in range(len(l)):
    Ecv = 0
    C = 1/(2*l[i])
    param = '-s 0 -c ' + str(C) + ' -e 0.000001'
    for j in range(5):
        X_train = X[:40*j] + X[40*(j+1):]
        Y_train = Y[:40*j] + Y[40*(j+1):]
        X_val = X[40*j:40*(j+1)]
        Y_val = Y[40*j:40*(j+1)]
#         print(np.shape(X_train), ":", np.shape(X_val))
        m = train(Y_train, X_train, param)
        p_val_labels, p_val_acc, p_val_vals = predict(Y_val, X_val,
m)
        Ecv += (100-p_val_acc[0])*0.01
    Ecv /= 5
    if Ecv <= min_Ecv:
        min_Ecv = Ecv
print("20.: ", min_Ecv)
```

```
Accuracy = 87.5% (35/40) (classification)
Accuracy = 77.5% (31/40) (classification)
Accuracy = 95% (38/40) (classification)
Accuracy = 77.5% (31/40) (classification)
Accuracy = 90% (36/40) (classification)
Accuracy = 85% (34/40) (classification)
Accuracy = 80% (32/40) (classification)
Accuracy = 95% (38/40) (classification)
Accuracy = 85% (34/40) (classification)
Accuracy = 95% (38/40) (classification)
Accuracy = 80% (32/40) (classification)
Accuracy = 90% (36/40) (classification)
Accuracy = 90% (36/40) (classification)
Accuracy = 80% (32/40) (classification)
Accuracy = 82.5% (33/40) (classification)
Accuracy = 77.5% (31/40) (classification)
Accuracy = 92.5% (37/40) (classification)
Accuracy = 85% (34/40) (classification)
Accuracy = 75% (30/40) (classification)
Accuracy = 80% (32/40) (classification)
Accuracy = 42.5% (17/40) (classification)
Accuracy = 65% (26/40) (classification)
Accuracy = 47.5% (19/40) (classification)
Accuracy = 40% (16/40) (classification)
Accuracy = 45% (18/40) (classification)
20.:  0.12
```

In [ ]: