

```

from libsvm.svmutil import *
import numpy as np
import random

train_y, train_x = svm_read_problem('train.txt')
test_y, test_x = svm_read_problem('test.txt')

#11
train_y_11 = [-1 if x != 1 else 1 for x in train_y]
prob11 = svm_problem(train_y_11, train_x)
param11 = svm_parameter('-s 0 -t 0 -c 1')
model_11 = svm_train(prob11, param11)

sv = np.asarray(model_11.get_SV())
sv_coef = np.asarray(model_11.get_sv_coef())
sv_indices = np.asarray(model_11.get_sv_indices())
rho = model_11.rho[0]

w = np.zeros(len(train_x[0]))
b = -rho
# weight vector w = sum over i ( coefsi * xi )
for index, coef in zip(sv_indices, sv_coef):
    #print(index, coef)
    ai = coef[0]
    for key in train_x[index-1]:
        #print(key)
        w[key-1] = w[key-1] + (ai * train_x[index-1][key])

print(((w**2).sum())**(1/2))
# 6.309673609961579

```

```

#12, 13
sv = []
err = []
param12 = svm_parameter('-s 0 -t 1 -d 2 -g 1 -r 1 -c 1')
for i in range(2,7):
    train_y_12 = [-1 if x != i else 1 for x in train_y]
    prob12 = svm_problem(train_y_12, train_x)
    model_12 = svm_train(prob12, param12)
    sv.append(len(model_12.get_sv_coef()))
    p_label, p_acc, p_val = svm_predict(train_y_12, train_x, model_12, '-q')
    err.append(1-p_acc[0]/100)
print(err)
print(err.index(max(err))+2)
print(sv)
print((min(sv)))

# [0.011333333333333306, 0.006761904761904747, 0.009619047619047638, 0.0148571
42857142902, 0.01123809523809527]
# 5
# [588, 368, 499, 642, 503]
# 368

#14
train_y_14 = [-1 if x != 7 else 1 for x in train_y]
test_y_14 = [-1 if x != 7 else 1 for x in test_y]
err = []
clist = [0.01, 0.1, 1, 10, 100]
for i in range(2,7):
    param14 = svm_parameter(f'-s 0 -t 2 -g 1 -c {clist[i-2]}')
    prob14 = svm_problem(train_y_14, train_x)
    model_14 = svm_train(prob14, param14)
    p_label, p_acc, p_val = svm_predict(test_y_14, test_x, model_14, '-q')
    err.append(1-p_acc[0]/100)
print(err)
print(clist[err.index(min(err))])

# [0.04519999999999991, 0.04519999999999991, 0.01419999999999999, 0.0040000000
000000036, 0.00539999999999996]
# 10

```

```

#15
train_y_14 = [-1 if x != 7 else 1 for x in train_y]
test_y_14 = [-1 if x != 7 else 1 for x in test_y]
err = []
glist = [0.1, 1, 10, 100, 1000]
for i in range(2,7):
    param14 = svm_parameter(f'-s 0 -t 2 -c 0.1 -g {glist[i-2]}')
    prob14 = svm_problem(train_y_14, train_x)
    model_14 = svm_train(prob14, param14)
    p_label, p_acc, p_val = svm_predict(test_y_14, test_x, model_14, '-q')
    err.append(1-p_acc[0]/100)
print(err)
print(glist[err.index(min(err))])

# [0.04519999999999991, 0.04519999999999991, 0.040200000000000014, 0.045199999
99999991, 0.04519999999999991]
# 10

```

```

## 16
train_y_14 = [-1 if x != 7 else 1 for x in train_y]
test_y_14 = [-1 if x != 7 else 1 for x in test_y]
glist = [0.1, 1, 10, 100, 1000]
best_g = [0,0,0,0,0]

for _ in range(500):
    print(_)
    random.seed(_+100)
    draw_val = random.sample(range(0, len(train_y)), 200)
    draw_train = [i for i in range(0,len(train_y)) if i not in draw_val]

    data_train_x = [train_x[i] for i in draw_train]
    data_val_x = [train_x[i] for i in draw_val]

    data_train_y = [train_y_14[i] for i in draw_train]
    data_val_y = [train_y_14[i] for i in draw_val]
    err = []
    for i in range(5):
        param16 = svm_parameter(f'-s 0 -t 2 -c 0.1 -g {glist[i]}')
        prob16 = svm_problem(data_train_y, data_train_x)
        model_16 = svm_train(prob16, param16)
        p_label, p_acc, p_val = svm_predict(data_val_y, data_val_x, model_1
6, '-q')
        err.append(1-p_acc[0]/100)
    best_g[err.index(min(err))] += 1
    print(_, best_g)
print(best_g)
# [279, 0, 221, 0, 0]
# 0.1

```