

```

import pandas as pd
import numpy as np
from liblinear.liblinearutil import *
import itertools
dtrain = pd.read_table('hw4_train.dat', header=None)
dtest = pd.read_table('hw4_test.dat', header=None)
tmp = dtrain.copy()
#tmp = tmp.iloc[:,1:4]

def _transform_withoutconst(x):
    cnt=0
    x_copy = x.copy()
    tt = x.iloc[:,10].copy()
    options = [i for i in range(11)]#11
    comb = list(itertools.product(options, repeat=4))
    unique_combs = [tuple(sorted(c)) for c in comb]
    u = list(set(unique_combs))
    #print(u)
    d = 10

    for i in u:
        tmp = tt.copy()
        for j in i:
            tmp *= x.iloc[:, j]
        x_copy[cnt] = tmp
        cnt+=1
    #print(x_copy)
    return x_copy

y_train = dtrain[10].tolist()
dtrain.iloc[:, 10] = 1
#print(dtrain)
dtrain = _transform_withoutconst(dtrain)

y_test = dtest[10].tolist()
dtest.iloc[:, 10] = 1
dtest = _transform_withoutconst(dtest)
#print(dtrain)
dtrain = dtrain.values.tolist()
dtest = dtest.values.tolist()
#print(dtest)

```

```

# 12
lamda_list = [-6, -3, 0, 3, 6]
best_lamda = -1
err = 1000000000
for ln_lamda in lamda_list:
    lamda = pow(10, ln_lamda)
    option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
    model_12 = train(y_train, dtrain, option)
    p_label, p_acc, p_val = predict(y_test, dtest, model_12)
    if 1-p_acc[0]/100 <= err:
        err = 1-p_acc[0]/100
        best_lamda = ln_lamda
print(best_lamda)
#3

```

```

#20
lamda = pow(10, 0)
option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
model_12 = train(y_train, dtrain, option)
www = model_12.get_decfun()
count = len([x for x in www[0] if abs(x) <= 0.000001])
print(count)
#0

```

```

# 13
lamda_list = [-6, -3, 0, 3, 6]
best_lamda = -1
err = 1000000000
for ln_lamda in lamda_list:
    lamda = pow(10, ln_lamda)
    option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
    model_13 = train(y_train, dtrain, option)
    p_label, p_acc, p_val = predict(y_train, dtrain, model_13)

    if 1-p_acc[0]/100<= err:
        err = 1-p_acc[0]/100
        best_lamda = ln_lamda
print(best_lamda)
#0

```

```

# 18
lamda_list = [-6, -3, 0, 3, 6]
best_lamda = -1
err = 1000000000
for ln_lamda in lamda_list:
    lamda = pow(10, ln_lamda)
    option = '-s 6 -c '+str(1/lamda)+' -e 0.000001'
    model_18 = train(y_train, dtrain, option)
    p_label, p_acc, p_val = predict(y_test, dtest, model_18)
    if 1-p_acc[0]/100 <= err:
        err = 1-p_acc[0]/100
        best_lamda = ln_lamda
print(best_lamda)
#0

```

```

#19
lamda = pow(10, 0)
option = '-s 6 -c '+str(1/lamda)+' -e 0.000001'
model_18 = train(y_train, dtrain, option)
www = model_18.get_decfun()
count = len([x for x in www[0] if abs(x) <= 0.000001])
print(count)
#959

```

#14

```
import random
import pandas as pd
import numpy as np
from liblinear.liblinearutil import *

dtrain = pd.read_table('hw4_train.dat', header=None)
dtest = pd.read_table('hw4_test.dat', header=None)
y_train = dtrain[10]
y_test = dtest[10]
dtrain.iloc[:, 10] = 1
dtest.iloc[:, 10] = 1

dtrain = _transform_withoutconst(dtrain)
dtest = _transform_withoutconst(dtest)
#print(dtrain)
dtrain[len(dtrain.columns)] = y_train
dtest[len(dtest.columns)] = y_test
y_test = dtest[len(dtest.columns)-1].tolist()
x_test = dtest.iloc[:,0:len(dtest.columns)-1].values.tolist()
#print(dtrain)

cnt = [0,0,0,0,0]
for _ in range(256):
    random.seed(_)
    draw_train = random.sample(range(0, 200), 120)
    #print(draw_train)
    draw_val = [i for i in range(0,200) if i not in draw_train]

    data_train = dtrain.iloc[draw_train, :]
    data_val = dtrain.iloc[draw_val, :]

    y_train = data_train[len(data_train.columns)-1].tolist()
    y_val = data_val[len(data_val.columns)-1].tolist()

    x_train = data_train.iloc[:,0:len(data_train.columns)-1].values.tolist()
    x_val = data_val.iloc[:,0:len(data_val.columns)-1].values.tolist()
    #print(x_train)
    lamda_list = [-6, -3, 0, 3, 6]
    best_lamda = -1
    err = 1000000000
    for ln_lamda in lamda_list:
        lamda = pow(10, ln_lamda)
        option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
        model_14 = train(y_train, x_train, option)
        p_label, p_acc, p_val = predict(y_val, x_val, model_14, '-q')
        diff_count = 0
        for i in range(len(p_label)):
            if p_label[i] != y_val[i]:
                diff_count += 1
```

```
    if diff_count/len(y_val) <= err:
        err = diff_count/len(y_val)
        best_lamda = ln_lamda
    cnt[lamda_list.index(best_lamda)] += 1
print(cnt)
print(lamda_list[cnt.index(max(cnt))])
#3
```

```

#15
import random
import pandas as pd
import numpy as np
from liblinear.liblinearutil import *

dtrain = pd.read_table('hw4_train.dat', header=None)
dtest = pd.read_table('hw4_test.dat', header=None)

y_train = dtrain[10]
y_test = dtest[10]
dtrain.iloc[:, 10] = 1
dtest.iloc[:, 10] = 1

dtrain = _transform_withoutconst(dtrain)
dtest = _transform_withoutconst(dtest)

dtrain[len(dtrain.columns)] = y_train
dtest[len(dtest.columns)] = y_test

y_test = dtest[len(dtest.columns)-1].tolist()
x_test = dtest.iloc[:,0:len(dtest.columns)-1].values.tolist()
#print(dtrain)

out_err = 0
for _ in range(256):
    random.seed(_)
    draw_train = random.sample(range(0, 200), 120)
    #print(draw_train)
    draw_val = [i for i in range(0,200) if i not in draw_train]

    data_train = dtrain.iloc[draw_train, :]
    data_val = dtrain.iloc[draw_val, :]

    y_train = data_train[len(data_train.columns)-1].tolist()
    y_val = data_val[len(data_val.columns)-1].tolist()

    x_train = data_train.iloc[:,0:len(data_train.columns)-1].values.tolist()
    x_val = data_val.iloc[:,0:len(data_val.columns)-1].values.tolist()
    #print(x_train)
    lamda_list = [-6, -3, 0, 3, 6]
    best_lamda = -1
    err = 1000000000
    for ln_lamda in lamda_list:
        lamda = pow(10, ln_lamda)
        option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
        model_14 = train(y_train, x_train, option)
        p_label, p_acc, p_val = predict(y_val, x_val, model_14, '-q')
        diff_count = 0
        for i in range(len(p_label)):

```

```

        if p_label[i] != y_val[i]:
            diff_count += 1
    if diff_count/len(y_val) <= err:
        err = diff_count/len(y_val)
        best_lamda = ln_lamda

lamda = pow(10, best_lamda)
option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
model_15 = train(y_train, x_train, option)
p_label, p_acc, p_val = predict(y_test, x_test, model_15, '-q')
diff_count = 0
for i in range(len(p_label)):
    if p_label[i] != y_test[i]:
        diff_count += 1
out_err += diff_count/len(y_test)

print(out_err/256)
#0.16913281250000015

```

```

#16
import random
import pandas as pd
import numpy as np
from liblinear.liblinearutil import *

dtrain = pd.read_table('hw4_train.dat', header=None)
dtest = pd.read_table('hw4_test.dat', header=None)

y_train = dtrain[10]
y_test = dtest[10]
dtrain.iloc[:, 10] = 1
dtest.iloc[:, 10] = 1

dtrain = _transform_withoutconst(dtrain)
dtest = _transform_withoutconst(dtest)

dtrain[len(dtrain.columns)] = y_train
dtest[len(dtest.columns)] = y_test

y_test = dtest[len(dtest.columns)-1].tolist()
x_test = dtest.iloc[:,0:len(dtest.columns)-1].values.tolist()
y_train_val = dtrain[len(dtrain.columns)-1].tolist()
x_train_val = dtrain.iloc[:,0:len(dtrain.columns)-1].values.tolist()
#print(dtrain)

out_err = 0
for _ in range(256):
    random.seed(_+100)
    draw_train = random.sample(range(0, 200), 120)
    #print(draw_train)
    draw_val = [i for i in range(0,200) if i not in draw_train]

    data_train = dtrain.iloc[draw_train, :]
    data_val = dtrain.iloc[draw_val, :]

    y_train = data_train[len(data_train.columns)-1].tolist()
    y_val = data_val[len(data_val.columns)-1].tolist()

    x_train = data_train.iloc[:,0:len(data_train.columns)-1].values.tolist()
    x_val = data_val.iloc[:,0:len(data_val.columns)-1].values.tolist()
    #print(x_train)
    lamda_list = [-6, -3, 0, 3, 6]
    best_lamda = -1
    err = 1000000000
    for ln_lamda in lamda_list:
        lamda = pow(10, ln_lamda)
        option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
        model_14 = train(y_train, x_train, option)
        p_label, p_acc, p_val = predict(y_val, x_val, model_14, '-q')

```



```

diff_count = 0
for i in range(len(p_label)):
    if p_label[i] != y_val[i]:
        diff_count += 1
if diff_count/len(y_val) <= err:
    err = diff_count/len(y_val)
    best_lamda = ln_lamda

lamda = pow(10, best_lamda)
option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
model_15 = train(y_train_val, x_train_val, option)
p_label, p_acc, p_val = predict(y_test, x_test, model_15, '-q')
diff_count = 0
for i in range(len(p_label)):
    if p_label[i] != y_test[i]:
        diff_count += 1
out_err += diff_count/len(y_test)

print(out_err/256)
#0.14941406250000025

```

```

#17
import random
import pandas as pd
import numpy as np
from liblinear.liblinearutil import *

dtrain = pd.read_table('hw4_train.dat', header=None)
dtest = pd.read_table('hw4_test.dat', header=None)
y_train = dtrain[10]
y_test = dtest[10]
dtrain.iloc[:, 10] = 1
dtest.iloc[:, 10] = 1

dtrain = _transform_withoutconst(dtrain)
dtest = _transform_withoutconst(dtest)

dtrain[len(dtrain.columns)] = y_train
dtest[len(dtest.columns)] = y_test
y_test = dtest[len(dtest.columns)-1].tolist()
x_test = dtest.iloc[:,0:len(dtest.columns)-1].values.tolist()
#print(dtrain)
cv_err = 0
v = 5
for _ in range(256):
    random.seed(_)
    draw_train = random.sample(range(0, 200), 200)
    fold = [draw_train[0:40],draw_train[40:80],draw_train[80:120],draw_train[1
20:160],draw_train[160:200]]

    #print(x_train)
    lamda_list = [-6, -3, 0, 3, 6]
    best_lamda = -1
    err = 1000000000
    for ln_lamda in lamda_list:

        lamda = pow(10, ln_lamda)
        option = '-s 0 -c '+str(1/2/lamda)+' -e 0.000001'
        diff_count = 0
        for f in range(5):
            rest_f = [i for i in range(200) if i not in fold[f]]
            #print(fold[f])
            data_train = dtrain.iloc[rest_f, :]
            data_val = dtrain.iloc[fold[f],:]

            y_train = data_train[len(data_train.columns)-1].tolist()
            y_val = data_val[len(data_val.columns)-1].tolist()
            x_train = data_train.iloc[:,0:len(data_train.columns)-1].values.to
list()
            x_val = data_val.iloc[:,0:len(data_val.columns)-1].values.tolist()

```

```

model_17 = train(y_train, x_train, option)
p_label, p_acc, p_val = predict(y_val, x_val, model_17, '-q')

for j in range(len(p_label)):
    if p_label[j] != y_val[j]:
        diff_count += 1

if diff_count/len(y_val)/v <= err:
    err = diff_count/len(y_val)/v
    best_lamda = ln_lamda
cv_err += err

print(cv_err/256)
#0.12968750000000004

```

In [ ]:

1	
---	--