# Problem1 a

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

c1 = SVC(kernel='linear', C=1)

c100 = SVC(kernel='linear', C=100)

with open("train\_x.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows)

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("train\_y.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

c1.fit(tdata, tlabel)

c100.fit(tdata, tlabel)

c1predict = c1.predict(tdata)

c100predict = c100.predict(tdata)

accuracy = accuracy\_score(tlabel, c1predict)

accuracy2 = accuracy\_score(tlabel, c100predict)

print('c = 1 accuracy : ', accuracy)

print('c = 100 accuracy : ', accuracy2)

plotSVMBoundaries(tdata, tlabel, c1)

plotSVMBoundaries(tdata, tlabel, c100)

#Problem1 b

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

c1 = SVC(kernel='linear', C=1)

c100 = SVC(kernel='linear', C=100)

with open("train\_x.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows)

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("train\_y.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

c1.fit(tdata, tlabel)

c100.fit(tdata, tlabel)

c1predict = c1.predict(tdata)

c100predict = c100.predict(tdata)

accuracy = accuracy\_score(tlabel, c1predict)

accuracy2 = accuracy\_score(tlabel, c100predict)

print('for c = 100')

print("the weights w0, w1, w2: (", c100.intercept\_[0], c100.coef\_[0][0], c100.coef\_[0][1], ")")

print("the decision boundary equation : ", c100.intercept\_[0], "+", c100.coef\_[0][0], "x1 +", c100.coef\_[0][1], "x2 +", "= 0")

plotSVMBoundaries(tdata, tlabel, c100, c100.support\_vectors\_)

#Problem 1 C

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

c1 = SVC(kernel='linear', C=1)

c100 = SVC(kernel='linear', C=100)

with open("train\_x.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows)

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("train\_y.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

c1.fit(tdata, tlabel)

c100.fit(tdata, tlabel)

c1predict = c1.predict(tdata)

c100predict = c100.predict(tdata)

accuracy = accuracy\_score(tlabel, c1predict)

accuracy2 = accuracy\_score(tlabel, c100predict)

print('for c = 100')

print("the weights w0, w1, w2: (", c100.intercept\_[0], c100.coef\_[0][0], c100.coef\_[0][1], ")")

print("the decision boundary equation : ", c100.intercept\_[0], "+", c100.coef\_[0][0], "x1 +", c100.coef\_[0][1], "x2 +", "= 0")

print('\n')

g = [0, 0, 0]

i = 0

for point in c100.support\_vectors\_:

g[i] = point[1] \* c100.coef\_[0][1] + point[0]\*c100.coef\_[0][0] + c100.intercept\_[0]

i = i + 1

print(g)

print("𝑔(𝑥) for each support vector: ", g)

plotSVMBoundaries(tdata, tlabel, c100, c100.support\_vectors\_)

#Problem1 d

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

c50 = SVC(kernel='rbf', C=50, gamma='auto')

c5000 = SVC(kernel='rbf', C=5000, gamma='auto')

with open("train\_x.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows)

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("train\_y.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

c50.fit(tdata, tlabel)

c5000.fit(tdata, tlabel)

c50predict = c50.predict(tdata)

c5000predict = c5000.predict(tdata)

accuracy = accuracy\_score(tlabel, c50predict)

accuracy2 = accuracy\_score(tlabel, c5000predict)

print('c=50 Gaussian (RBF) Kernel : ', accuracy)

print('c=5000 Gaussian (RBF) Kernel : ', accuracy2)

plotSVMBoundaries(tdata, tlabel, c50)

plotSVMBoundaries(tdata, tlabel, c5000)

Problem1 e

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

gama10 = SVC(kernel='rbf', gamma=10)

gama50 = SVC(kernel='rbf', gamma=50)

gama5000 = SVC(kernel='rbf', gamma=500)

with open("train\_x.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows)

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("train\_y.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

gama10.fit(tdata, tlabel)

gama50.fit(tdata, tlabel)

gama5000.fit(tdata, tlabel)

gama10predict = gama10.predict(tdata)

gama50predict = gama50.predict(tdata)

gama5000predict = gama5000.predict(tdata)

accuracy = accuracy\_score(tlabel, gama10predict)

accuracy2 = accuracy\_score(tlabel, gama50predict)

accuracy3 = accuracy\_score(tlabel, gama5000predict)

print('Gaussian Kernel gama=10 : ', accuracy)

print('Gaussian Kernel gama=50 : ', accuracy2)

print('Gaussian Kernel gama=5000 :', accuracy3)

plotSVMBoundaries(tdata, tlabel, gama10)

plotSVMBoundaries(tdata, tlabel, gama50)

plotSVMBoundaries(tdata, tlabel, gama5000)

Problem2 a

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

cross = StratifiedKFold(n\_splits=5, shuffle=True)

gama1 = SVC(kernel='rbf', C=1, gamma=1)

with open("feature\_train.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows[0:2])

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("label\_train.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

accuracy = cross\_val\_score(gama1, tdata, tlabel, cv=cross)

accmean = sum(accuracy)/5

print("average cross- validation accuracy", accmean)

Problem2b

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

cross = StratifiedKFold(n\_splits=5, shuffle=True)

c = np.logspace(-3, 3, 100)

gamma = np.logspace(-3, 3, 100)

acc = np.zeros((100, 100))

dev = np.zeros((100, 100))

with open("feature\_train.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows[0:2])

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("label\_train.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

for i in range(len(c)):

for j in range(len(gamma)):

cgama = SVC(kernel='rbf', C=c[i], gamma=gamma[j])

accuracy = cross\_val\_score(cgama, tdata, tlabel, cv=cross)

acc[i][j] = np.mean(accuracy)

dev[i][j] = np.std(accuracy)

print('\n')

bestc = c[np.where(acc == np.max(acc))[0]]

bestgama = gamma[np.where(acc == np.max(acc))[1]]

bestdev = dev[np.where(acc == np.max(acc))]

bestacc = np.max(acc)

print("best values of [γ, C] ", bestgama, bestc)

print('\n')

print("mean cross-validation accuracy and standard deviation ", bestacc, bestdev)

image = plt.imshow(acc)

plt.colorbar()

plt.show()

Problem 2c

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

c = np.logspace(-3, 3, 50)

gamma = np.logspace(-3, 3, 50)

for i in range(20):

acc = np.zeros((50, 50))

dev = np.zeros((50, 50))

cross = StratifiedKFold(n\_splits=5, shuffle=True)

with open("feature\_train.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows[0:2])

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("label\_train.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

for i in range(20):

for j in range(len(c)):

for k in range(len(gamma)):

cgama = SVC(kernel='rbf', C=c[j], gamma=gamma[k])

accuracy = cross\_val\_score(cgama, tdata, tlabel, cv=cross)

acc[j][k] = np.mean(accuracy)

dev[j][k] = np.std(accuracy)

bestc = c[np.where(acc == np.max(acc))[0][0]]

bestgama = gamma[np.where(acc == np.max(acc))[1][0]]

finalacc = acc[np.where(acc == np.max(acc))[0][0], np.where(acc == np.max(acc))[1][0]]

finaldev = dev[np.where(acc == np.max(acc))[0][0], np.where(acc == np.max(acc))[1][0]]

print("T =", i, "[γ, C] : ", [round(bestgama, 8), round(bestc, 8)])

print("T =", i, "accuracy and standard deviation : ", round(finalacc, 8), " ", round(finaldev, 8))

print("\n")

Problem 2d

import math

import numpy as np

from sklearn.preprocessing import StandardScaler

import pandas as pd

np.set\_printoptions(threshold=np.inf)

# np.set\_printoptions(precision=9)

from sklearn import preprocessing

from libsvm.svm import \*

from libsvm.svmutil import \*

from plotSVMBoundaries import \*

from sklearn.svm import \*

from sklearn.metrics import \*

from sklearn.model\_selection import StratifiedKFold

from sklearn.model\_selection import cross\_val\_score

import matplotlib.pyplot as plt

set = SVC(kernel='rbf', gamma=0.06866488, C=244.20530945)

with open("feature\_train.csv") as datatemp:

train = list()

reader = datatemp.readlines()

for rows in reader:

rows = rows.split(',')

train.append(rows[0:2])

train = [list(map(float, rows2)) for rows2 in train]

tdata = np.array(train)

with open("label\_train.csv") as datatemp2:

train2 = list()

reader2 = datatemp2.readlines()

for rows2 in reader2:

rows2 = rows2.split(',')

train2.append(rows2)

train2 = [list(map(float, rows3)) for rows3 in train2]

tlabel = np.array(train2)

tlabel = np.hstack(tlabel.T)

with open("feature\_test.csv") as datatemp3:

train3 = list()

reader3 = datatemp3.readlines()

for rows3 in reader3:

rows3 = rows3.split(',')

train3.append(rows3[0:2])

train3 = [list(map(float, rows4)) for rows4 in train3]

testdata = np.array(train3)

with open("label\_test.csv") as datatemp4:

train4 = list()

reader4 = datatemp4.readlines()

for rows4 in reader4:

rows4 = rows4.split(',')

train4.append(rows4)

train4 = [list(map(float, rows5)) for rows5 in train4]

testlabel = np.array(train4)

testlabel = np.hstack(testlabel.T)

set.fit(testdata, testlabel)

setpredict = set.predict(testdata)

accuracy = accuracy\_score(testlabel, setpredict)

print("accuracy of test set:", accuracy)

plotSVMBoundaries(testdata, testlabel, set)