HW6_edgarsp2

November 11, 2019

1 STAT 542 / CS 598: Homework 6

Fall 2019, by Edgar Pino

Due: Monday, Nov 11 by 11:59 PM Pacific Time

[2]:
```
%matplotlib inline
```

[24]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.optimize import minimize
from cvxopt import matrix, solvers
import math
```

2 Question 1 [50 Points] Linearly Separable SVM using Quadratic Programming

[4]:
```
np.random.seed(1)
```

[5]:
```
COLOR_LABELS = ["darkorange","deepskyblue"]
```

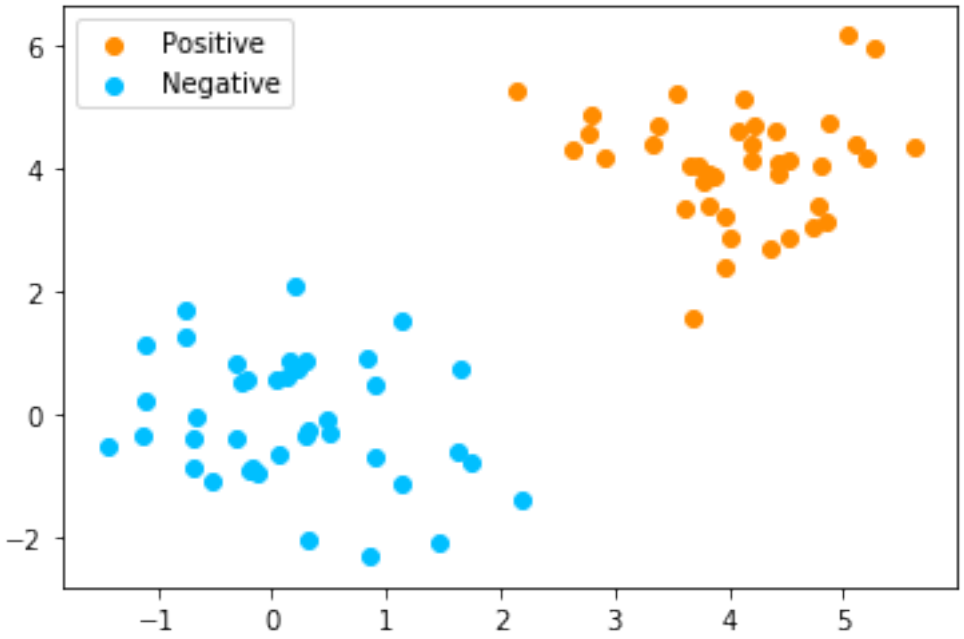
[6]:
```
N = 40
P = 2
```

[7]:
```
xpos = np.random.normal(0, 1,(N,P))
xneg = np.random.normal(4, 1,(N,P))
```

[8]:
```
x = np.concatenate((xpos, xneg), axis=0)
y = np.concatenate((np.ones((N,)), -np.ones((N,))), axis=0)
```

[9]:
```
def plot_data(x, y):
    unique = np.unique(y)
    for li in range(len(unique)):
        items = x[y == unique[li]]
        label = 'Positive' if unique[li] == -1 else 'Negative'
        plt.scatter(items[:, 0], items[:, 1], c = COLOR_LABELS[li], label=label)
    plt.legend()
```

[10]:
```
plot_data(x,y)
```



2.0.1 Fit

[42]:
```
def fit(x, y):
    NUM = x.shape[0]
    DIM = x.shape[1]
    K = y[:, None] * x
    K = np.dot(K, K.T)
    P = matrix(K)
    q = matrix(-np.ones((NUM, 1)))
    G = matrix(-np.eye(NUM))
    h = matrix(np.zeros(NUM))
    A = matrix(y.reshape(1, -1))
    b = matrix(np.zeros(1))

    return solvers.qp(P, q, G, h, A, b)
```

[43]:
```
def plot_separator(ax, w, b):
    slope = -w[0] / w[1]
    intercept = -b / w[1]
    x = np.arange(0, 6)
    ax.plot(x, x * slope + intercept, 'k-')
    ax.plot(x, x * slope + intercept + 2.75, '-')
    ax.plot(x, x * slope + intercept - 2.75, '-')
```

[44]:
```
def plot_data_with_labels(x, y, ax):
    unique = np.unique(y)
```

```
    for li in range(len(unique)):
        x_sub = x[y == unique[li]]
        label = 'Positive' if unique[li] == 1 else 'Negative'
        ax.scatter(x_sub[:, 0], x_sub[:, 1], c = COLOR_LABELS[li], label=label)
```

[45]:
```
sol = fit(x, y)
alphas = np.array(sol['x'])
```

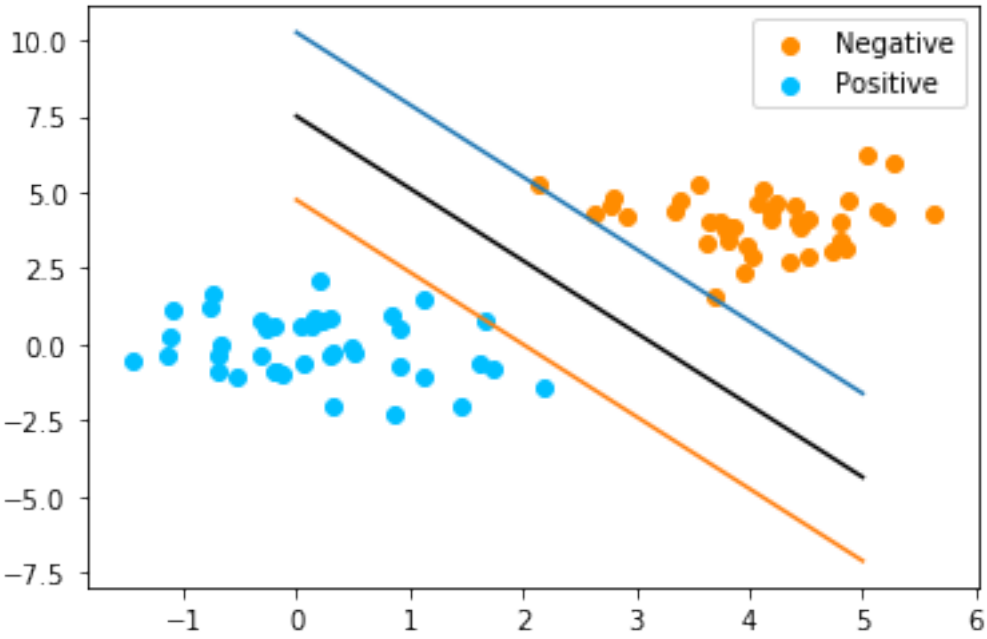
```
     pcost       dcost       gap    pres   dres
 0: -6.2205e+00 -1.0607e+01  2e+02  1e+01  2e+00
 1: -5.1204e+00 -1.6071e+00  3e+01  2e+00  2e-01
 2: -3.0840e-01 -5.4380e-01  5e-01  2e-02  2e-03
 3: -3.7365e-01 -4.2487e-01  9e-02  3e-03  3e-04
 4: -4.1115e-01 -4.1811e-01  9e-03  2e-04  2e-05
 5: -4.1661e-01 -4.1693e-01  3e-04  2e-06  2e-07
 6: -4.1685e-01 -4.1686e-01  6e-06  2e-08  2e-09
 7: -4.1685e-01 -4.1685e-01  6e-08  2e-10  2e-11
Optimal solution found.
```

[46]:
```
w = np.sum(alphas * y[:, None] * x, axis = 0)
cond = (alphas > 1e-4).reshape(-1)
b = y[cond] - np.dot(x[cond], w)
bias = b[0]

norm = np.linalg.norm(w)
w, bias = w / norm, bias / norm

fig, ax = plt.subplots()
plot_separator(ax, w, bias)
plot_data_with_labels(x, y, ax)

plt.legend()
plt.show()
```



3 Question 2 [25 Points] Linearly Non-seperable SVM using Penalized Loss

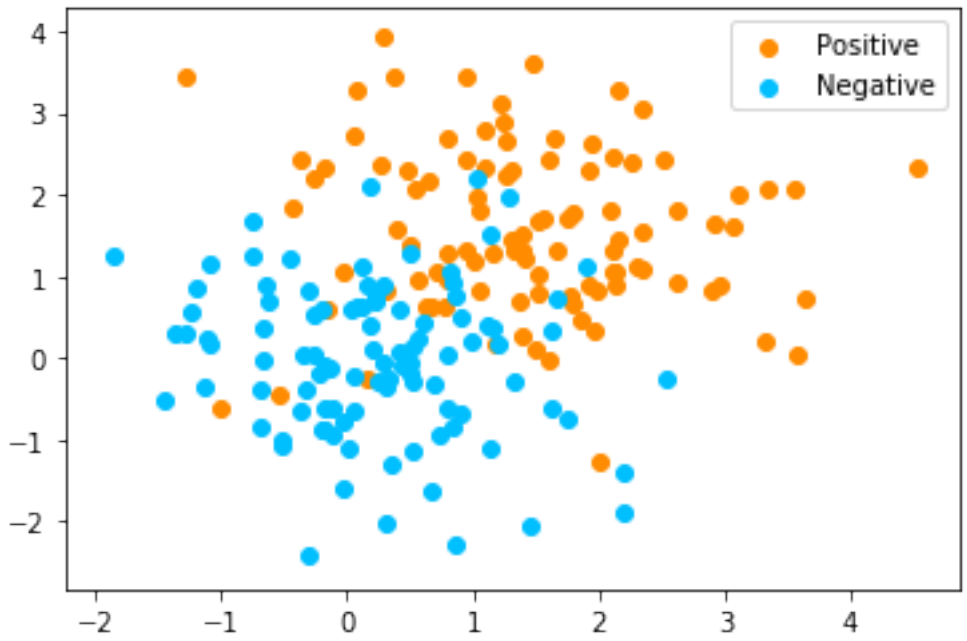
[51]:
```
np.random.seed(1)
```

[52]:
```
N = 100
P = 2
```

[53]:
```
xpos = np.random.normal(0, 1,(N,P))
xneg = np.random.normal(1.5, 1,(N,P))
```

[54]:
```
x = np.concatenate((xpos, xneg), axis=0)
y = np.concatenate((np.ones((N,)), -np.ones((N,))), axis=0)
```

[55]:
```
plot_data(x,y)
```



[56]:
```
def loss(b, x, y):
    return np.sum(
        np.log(
            1+np.exp(
                -y*
                (b[0]+np.dot(x,np.array([b[1],b[2]])))
            ))
    )+0.5*(np.power(b[0],2)+np.power(b[1],2)+np.power(b[2],2))
```

[57]:
```
results = minimize(loss, (1,1,1), args=(x,y))
```

[58]:
```
intercept = -results['x'][0]/results['x'][2]
```

[59]:
```
slope = -results['x'][1]/results['x'][2]
```

[60]:
```
intercept
```

[60]:
```
1.3611124182617051
```

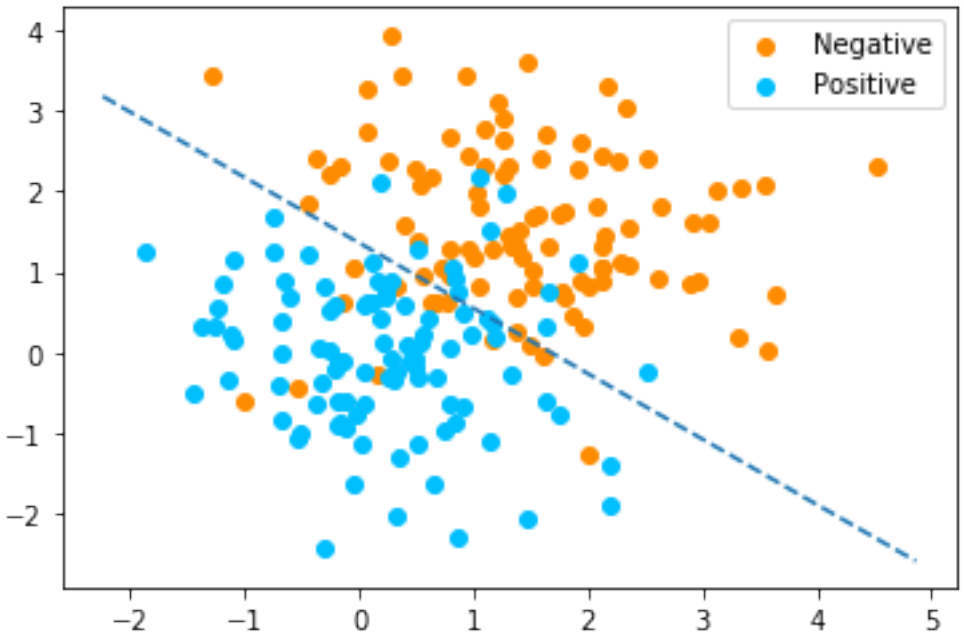
[61]:
```
slope
```

[61]:
```
-0.812670650533452
```

[62]:
```
fig, ax = plt.subplots()
plot_data_with_labels(x, y, ax)

axes = plt.gca()
x_vals = np.array(axes.get_xlim())
y_vals = intercept + slope * x_vals
ax.plot(x_vals, y_vals, '--')
```

```
plt.legend()
plt.show()
```



[ ]:

4 Question 3 [25 Points] Nonlinear and Non-seperable SVM using Penalized Loss

[11]:
```
from scipy.spatial.distance import pdist
from rpy2.robjects.packages import STAP
import rpy2.robjects.numpy2ri
from rpy2.robjects.packages import importr
rpy2.robjects.numpy2ri.activate()
```

[12]:
```
import os
os.environ['KMP_DUPLICATE_LIB_OK']='True'
```

[13]:
```
mfunc = 'get_k <- function(x){return(as.matrix(dist(x)^2)/0.25)}'
```

[14]:
```
myasmatrix = STAP(mfunc, "myasmatrix")
```

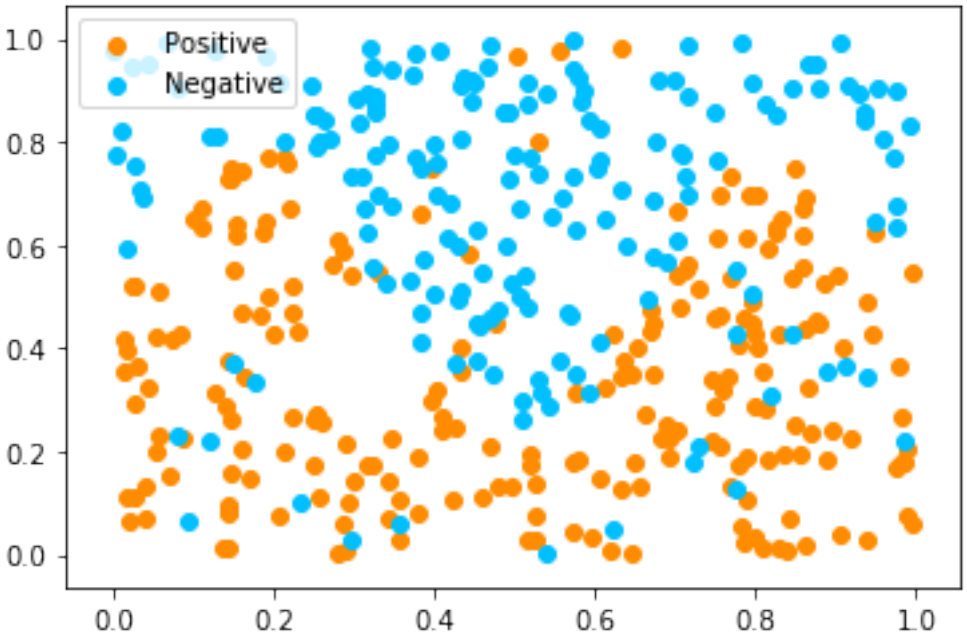
[15]:
```
N = 400
P = 2
```

[16]:
```
x = np.random.uniform(0,1,N*P).reshape(N,P)
```

[17]:
```
side = (x[:, 1] > 0.5 + 0.3*np.sin(3*math.pi*x[:, 0]))
```

[18]:
```
y = np.random.choice(np.array([1, -1]), N, True, np.array([0.9, 0.1]))*(side ==␣
↪1) + np.random.choice(np.array([1, -1]), N, True, np.array([0.1, 0.
↪9]))*(side == 0)
```

[19]:
```
plot_data(x,y)
```



[20]:
```
K = np.array(np.exp(-1 * np.array(myasmatrix.get_k(x))/.25))
```

[76]:
```
def my_k_loss(b, k, y):
    return np.sum(
        np.log(
            1+np.exp(
                -y*np.dot(k,b)
            )
        )
    )+np.sum(0.05*b%k%b)
```

[77]:
```
results = minimize(my_k_loss, np.ones(N), args=(K,y))
```

```
/anaconda3/lib/python3.7/site-packages/scipy/optimize/optimize.py:696:
RuntimeWarning: invalid value encountered in double_scalars
  grad[k] = (f(*((xk + d,) + args)) - f0) / d[k]
```

[79]:
```
results['x']
```

[79]:
```
array([1.15558281, 0.70875382, 0.63669815, 0.79667738, 0.85216392,
       0.83081225, 0.84137961, 0.72032158, 0.56282701, 0.73931824,
       0.75969523, 0.8130382 , 0.83313547, 0.59302644, 0.65992754,
       0.71233993, 0.83769635, 0.89331255, 0.85564678, 0.81355223,
       0.73191985, 0.70762637, 0.81347523, 0.71927339, 0.59213222,
       0.5980843 , 0.66520484, 0.63044827, 0.61124229, 0.76951721,
       0.65332069, 0.79852725, 0.85798987, 0.5617628 , 0.81162721,
       0.92121337, 0.78153629, 0.80277459, 0.65262116, 0.86834633,
       1.29268991, 0.73676209, 0.8385555 , 0.6523749 , 0.66611628,
       0.61270135, 0.58616137, 0.85472188, 0.64655303, 0.67138266,
       0.56904923, 0.83162056, 0.74578153, 1.04213048, 0.67870159,
       0.71208931, 0.75320888, 0.79521885, 0.67741315, 0.702786  ,
       0.82368113, 0.62970758, 0.86374028, 1.31426082, 0.73519383,
       0.67212268, 0.68532017, 0.74072777, 0.61720687, 0.57829385,
       0.57903322, 0.86683974, 0.62605405, 0.66027202, 0.75668138,
       0.73713743, 0.61690083, 0.75103914, 0.76783306, 0.70103815,
       0.72543048, 0.57548963, 0.73471236, 0.75361103, 0.61849657,
       0.57998583, 0.7098075 , 0.78479017, 0.69523624, 0.81972859,
       0.78737078, 0.7097401 , 0.82264276, 0.78431503, 0.84866025,
       0.62163203, 0.68221662, 0.72755932, 0.7174603 , 0.55423288,
       0.59159912, 0.6732506 , 0.71563074, 0.77201604, 0.63158439,
       0.80397743, 0.78864517, 0.68010466, 0.68678498, 0.95923493,
       0.73145427, 0.61473248, 0.69319686, 1.00367414, 0.86587116,
       0.84108271, 0.65884096, 0.70838442, 0.88463317, 0.74797158,
       0.62803947, 0.62306683, 0.80346811, 0.63895012, 0.57535216,
       0.69205292, 0.60684858, 0.62728362, 0.70737277, 0.76052302,
       0.83758071, 0.72854795, 0.81911373, 0.7659454 , 0.75281677,
       0.7052176 , 0.591633  , 0.8419964 , 0.61223551, 0.84516312,
       0.83798488, 0.64583305, 0.67229606, 0.61708582, 0.61134366,
       0.59889236, 0.55978286, 0.60244959, 0.69542959, 0.62263562,
       0.60485228, 0.91828448, 0.74762679, 0.91525649, 0.69817277,
       0.72874876, 0.64817876, 0.74117094, 0.8482199 , 0.67720632,
       0.61065295, 0.71914779, 0.79012797, 0.68106251, 0.7464706 ,
       0.56444432, 0.7002344 , 0.69974973, 0.58231344, 1.03435831,
       0.76826121, 0.61184052, 0.73180064, 0.71466502, 0.73012118,
       0.84571263, 0.69726276, 0.82928048, 0.68913852, 0.66352207,
       0.69691054, 0.56616597, 0.68606745, 0.55263171, 0.57032293,
       0.56436256, 0.77099601, 0.76684714, 0.56692635, 0.62518303,
       0.65164109, 0.79756595, 0.79188739, 0.68164673, 0.83951596,
       0.63531998, 1.38164649, 1.00833195, 0.59741445, 1.02566227,
       0.63905352, 0.66237537, 0.77970275, 0.71625123, 0.5831348 ,
       0.78241923, 0.68963192, 0.76361527, 0.66166683, 0.6678408 ,
       0.60221027, 0.62529056, 0.84989791, 0.70825499, 0.62121803,
       0.6778375 , 0.60274266, 0.63706882, 0.6302592 , 0.77579368,
       0.66816127, 0.58551187, 0.79118251, 0.68926453, 0.76640152,
       0.58063136, 0.74201191, 0.64600727, 0.79398604, 0.87102789,
       0.82865709, 1.02617864, 0.61016202, 0.77718103, 0.79362984,
```

```
0.6629127 , 0.77235963, 1.00732243, 0.62556691, 0.66533524,
0.82553891, 0.59773219, 0.64262153, 0.58066187, 0.67688981,
0.58311446, 0.75271182, 0.66352882, 0.72797191, 0.61391032,
0.58066318, 0.6044286 , 0.8403504 , 0.78928003, 0.69166278,
0.70497485, 0.84682147, 0.60117856, 0.59181586, 0.63099777,
0.73530564, 0.71861209, 0.75451909, 0.84762548, 0.85408246,
0.60232258, 0.78899257, 0.62156683, 0.8504993 , 0.82861802,
0.77000007, 0.70310839, 0.8581083 , 0.650706  , 0.80353209,
0.64016251, 0.60203972, 0.71622995, 0.56526504, 0.58294112,
0.61718202, 0.61254866, 1.25422837, 0.86404301, 0.58467305,
0.81337095, 0.78832955, 0.64312544, 0.75742773, 0.792693  ,
0.69641678, 0.82828381, 0.64350653, 0.83562934, 1.03645137,
0.71343492, 0.56418926, 0.71840348, 1.03622128, 0.59126521,
0.62334829, 0.72626896, 0.55887496, 0.56340463, 0.6494971 ,
0.67155376, 0.75425709, 0.72849339, 0.65169169, 0.83289332,
0.66268288, 0.70133248, 0.64136686, 0.82111618, 0.57429242,
0.61380465, 0.60908131, 1.0635679 , 0.61026012, 0.85747152,
0.7389343 , 0.7183351 , 0.65547081, 0.74066176, 0.76170961,
0.95977351, 0.88675008, 0.69624981, 0.63561203, 0.75521627,
0.67694151, 0.70952742, 0.61321703, 0.5527789 , 0.59386461,
0.68221838, 0.57507834, 0.65527599, 0.68799676, 0.68823273,
0.62407918, 0.61417007, 0.83427998, 0.87084267, 0.61909416,
0.82397428, 0.68642943, 0.59516321, 1.01827947, 0.61402378,
0.77527317, 0.70787347, 0.67670279, 0.81235126, 0.67574544,
0.64521861, 1.15834287, 0.68565191, 0.76900722, 0.85161726,
0.7928776 , 0.80571602, 0.75714527, 0.80628551, 0.77621197,
0.61405679, 0.71511316, 0.75504391, 0.69739854, 0.7420113 ,
0.58722887, 0.70670828, 0.84812818, 0.84768146, 0.75321783,
0.63394483, 0.66841825, 0.83683028, 0.83299792, 0.61936038,
0.7051146 , 0.72780196, 0.68607752, 0.6072381 , 0.72526687,
0.64862732, 0.89223517, 0.56797903, 0.61622736, 0.87597471,
0.74231113, 0.78671239, 0.82948971, 0.76166175, 0.60628193,
0.72489097, 0.77564554, 0.58994977, 0.61509671, 0.7449496 ])
```

[ ]: