

先端データ解析論 (杉山将先生・本多淳也先生)

第7回レポート

ashiato45

2017年5月29日

宿題1

1. $\alpha_i = 0$ とする。 $\alpha_i + \beta_i = C$ なので $\beta_i = C > 0$ となる。よって、 $\beta_i \xi_i = 0$ より、 $\xi_i = 0$ とならなければならない。 $y_i w^\top x_i - 1 + \xi_i \geq 0$ にこれを代入し、 $y_i w^\top x_i \geq 1$ を得る。
2. $0 < \alpha_i < C$ とする。 $\alpha_i + \beta_i = C$ と $\alpha_i < C$ より $\beta_i > 0$ である。よって、 $\xi_i = 0$ となる。 $0 < \alpha$ より $y_i w^\top x_i - 1 + \xi_i = 0$ となる。ここに $\xi_i = 0$ を代入し、 $y_i w^\top x_i = 1$ を得る。
3. $\alpha_i = C$ とする。 $\alpha_i > 0$ なので $y_i w^\top x_i - 1 + \xi_i = 0$ となる。 $y_i w^\top x_i = 1 - \xi_i \leq 1$ となる。
4. $y_i w^\top x_i > 1$ とする。 $y_i w^\top x_i - 1 + \xi_i > \xi_i \geq 0$ である。これと $\alpha_i (y_i w^\top x_i - 1 + \xi_i) = 0$ より、 $\alpha_i = 0$ を得る。
5. $y_i w^\top x_i < 1$ とする。 $\alpha_i \geq 0$ であることと2の対偶より、 $\alpha_i = 0$ であるか $\alpha_i \geq C$ のどちらかである。さらに1の対偶より $\alpha_i \neq 0$ がわかるので、 $\alpha_i \geq C$ である。 $\beta_i \geq 0$ と $\alpha_i + \beta_i = C$ より $\alpha_i \leq C$ がわかるので、 $\alpha_i = C$ である。

宿題2

以下の最小化を行なった：

$$\min_{w,b} \sum_{i=1}^{200} \max(0, 1 - y_i f_{w,b}(x_i)) + \frac{\lambda}{2} (w_1, w_2, b)(w_1, w_2, b)^\top \quad (1)$$

ここで、 $\lambda = 1$ とした。最適化には劣勾配法を用い、終了条件は「最適化するベクトルがあまり動かなくなったら」とした。すると、図1を得た。

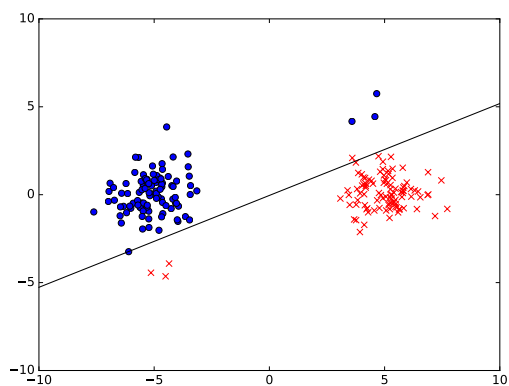


図1

Python での実装は付録にある。

付録

```
""" An assignment of ADA 6 """

import numpy as np
import numpy.matlib
import matplotlib.pyplot as plt

np.random.seed(42)

H = 1
L = 1
EPS = 10e-6

N = 200
X = np.hstack([np.random.randn(1, N//2) - 5, np.random.randn(1, N//2) + 5])
# add third line to simplify the calculation of f.
Y = np.vstack([X, np.random.randn(1, N), np.ones((1, N))])
X = X.transpose() # Make two groups
Y = np.vstack([np.ones((N//2, 1)), -np.ones((N//2, 1))])
X[0:3, 1] -= 5 # Make exceptional data
X[N//2:N//2+3, 1] += 5
Y[0:3] = -1
Y[N//2:N//2+3] = 1

def f(theta_, x_):
    return np.dot(theta_.transpose(), x_)

theta = np.ones((3, 1))
while True:
    d = -X*np.matlib.repmat(Y, 1, 3) # Nx3
    ind = 1 - Y*X.dot(theta)
    d[ind <= 0] = 0
    d = np.sum(d, axis=0) #1x3
    d = d.reshape(3, 1) #3x1
    d += L*theta
    theta2 = theta - EPS*d
    err = np.linalg.norm(theta2 - theta)
    print(err)
    theta = theta2
    if err < 10e-5:
        break

print(theta)

plt.axis([-10, 10, -10, 10])
plt.plot(X[(Y == 1).flatten(), 0], X[(Y == 1).flatten(), 1], 'bo')
plt.plot(X[(Y == -1).flatten(), 0], X[(Y == -1).flatten(), 1], 'rx')
plt.plot(np.array([-10, 10]), -(theta[2] + np.array([-10, 10])*theta[0])/theta[1], 'k-')
plt.show()
```