先端データ解析論　第 5 回レポート

電子情報学専攻　48-176403 石毛真修

2017 年 5 月 23 日

大問 1.

相補条件

$$\alpha_i(y_i\omega^{\mathrm{T}}\mathbf{x}_i - 1 + \xi_i) = 0 \tag{1}$$
$$\beta_i\xi_i = 0 \tag{2}$$

より，次が成り立つことを示す．

$$\alpha_i = 0 \rightarrow y_i\omega^{\mathrm{T}}\mathbf{x}_i \geq 1 \tag{3}$$
$$0 < \alpha_i < C \rightarrow y_i\omega^{\mathrm{T}}\mathbf{x}_i = 1 \tag{4}$$
$$\alpha_i = C \rightarrow y_i\omega^{\mathrm{T}}\mathbf{x}_i \leq 1 \tag{5}$$
$$y_i\omega^{\mathrm{T}}\mathbf{x}_i > 1 \rightarrow \alpha_i = 0 \tag{6}$$
$$y_i\omega^{\mathrm{T}}\mathbf{x}_i < 1 \rightarrow \alpha_i = C \tag{7}$$

証明

(3)

$\alpha_i + \beta_i = C$ より，$\beta_i = C$.
式 (2) より，$\beta_i \neq 0$ なので，$\xi_i = 0$.
よって，$y_i\omega^{\mathrm{T}}\mathbf{x}_i - 1 + \xi_i \geq 0$ なので，$y_i\omega^{\mathrm{T}}\mathbf{x}_i \geq 1$

(4)

$\alpha_i + \beta_i = C$ より，$\beta_i \neq 0$. これと式 (2) より，$\xi_i = 0$.
また，$\alpha_i \neq 0$ であるので，式 (1) より $y_i\omega^{\mathrm{T}}\mathbf{x}_i - 1 + \xi_i = 0$.
以上から，$y_i\omega^{\mathrm{T}}\mathbf{x}_i = 1$

(5)

式 (1) より，$\alpha_i \neq 0$ なので，$y_i\omega^{\mathrm{T}}\mathbf{x}_i - 1 + \xi_i = 0$.
$\xi_i \geq 0$ であるので，$y_i\omega^{\mathrm{T}}\mathbf{x}_i \leq 1$

(6)

$y_i \omega^{\mathrm{T}} \mathbf{x}_i - 1 > 0$ なので，$y_i \omega^{\mathrm{T}} \mathbf{x}_i - 1 + \xi_i > \xi_i \geq 0$.
よって，$y_i \omega^{\mathrm{T}} \mathbf{x}_i - 1 + \xi_i > 0$ であり，式 (1) より，$\alpha_i = 0$.

(7)

$y_i \omega^{\mathrm{T}} \mathbf{x}_i - 1 < 0$ であるので，$0 \leq y_i \omega^{\mathrm{T}} \mathbf{x}_i - 1 + \xi_i < \xi_i$．よって，$\xi_i > 0$.
これと式 (2) より，$\beta_i = 0$．$\alpha_i + \beta_i = C$ なので，$\alpha_i = C$.

大問 2.

線形モデル $f_{\omega,b}(\mathbf{x}) = \omega^{\mathrm{T}} + b$ に対する SVM の劣勾配アルゴリズムを実装する.

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

# Generate data to fit
rs = np.random.RandomState(42)
n = 200
X = np.array([np.concatenate((rs.randn(n//2) + 5, rs.randn(n//2) - 5)),
                             rs.randn(n)])
Y = np.concatenate((np.ones(n//2), - np.ones(n//2)))
Y[:3] = -1
Y[n//2:n//2 + 3] = 1
X[1, :3] += 5
X[1, n//2:n//2 + 3] -= 5

# Fitting
p_rs = np.random.RandomState(15)
theta = np.zeros(n)
b = 0

def f_th(x):
    global theta, b, X
    return x @ X @ theta + b

def sub_diff(i, k):
    """Sub-differentiate by k_th theta"""
    global theta, b, X, y
    if 1 - y[i] * f_th(X[:, i]) > 0:
        return - y[i] * np.matmul(X[:, k], X[:, i])
    else:
        return 0

def sub_diff_b(i):
    """Sub-differentiate by b"""
    global theta, b, X, y
    if 1 - y[i] * f_th(X[:, i]) > 0:
        return - y[i]
    else:
        return 0                                            C

def update_theta(k, lamb=0.01):
    """Return update for k_th theta"""
    global theta, n
    val = lamb * theta[k]
    for i in range(n):
        val += sub_diff(i, k)
    return val

def update_b():
    """Return update for b"""
    global n
    val = 0
    for i in range(n):
        val += sub_diff_b(i)
    return val

iter_num = 200
for l in range(iter_num):
    eps = 0.1
    new_theta = np.copy(theta)
    for i, t in enumerate(theta):
        new_theta[i] -= eps * update_theta(i)
    b -= eps * update_b()
    theta = new_theta

w = X @ theta
print(w)

# Show result
plt.scatter(X[0, np.where(Y==1)], X[1, np.where(Y==1)], c='r', marker='x')
plt.scatter(X[0, np.where(Y==-1)], X[1, np.where(Y==-1)], c='b', marker='o')
_x = np.linspace(-10, 10)
plt.plot(_x, b - _x * w[0] / w[1])
plt.ylim(-10, 10)
plt.xlim(-10, 10)
plt.show()
```

このコードを実行すると次のような結果を得た.

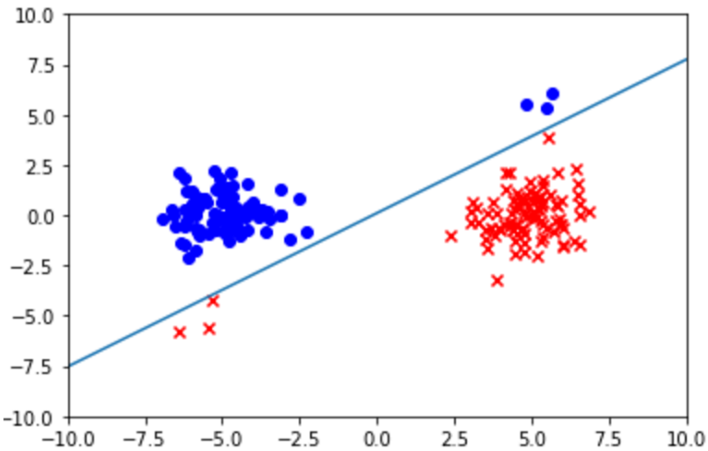

図 1: SVM の直線による分類