

# 先端データ解析論 (杉山将先生・本多淳也先生)

## 第8回レポート

ashiato45

2017年6月8日

### 宿題 1

まず、 $1 - \mu^\top \phi(x)y \geq 0$  のときを考える。

$$\frac{\partial J(\mu, \Sigma)}{\partial \mu} = 2(-\phi(x)y)(1 - \hat{\mu}^\top \phi(x)y) + 2\gamma \hat{\Sigma}^{-1}(\hat{\mu} - \tilde{\mu}) \quad (1)$$

これが0になるときを考えると、( $y^2 = 1$  に注意して)

$$\hat{\mu} = \left( \frac{\phi(x)\phi(x)^\top}{\gamma} + \hat{\Sigma}^{-1} \right)^{-1} \left( \frac{\phi(x)y}{\gamma} + \hat{\Sigma}^{-1}\tilde{\mu} \right) \quad (2)$$

となる。逆行列の公式より、

$$\hat{\mu} = \left( \hat{\Sigma} - \frac{\hat{\Sigma}\phi(x)\phi(x)^\top\hat{\Sigma}}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \right) \left( \frac{\phi(x)y}{\gamma} + \hat{\Sigma}^{-1}\tilde{\mu} \right) \quad (3)$$

$$= \left( \hat{\Sigma} - \frac{\hat{\Sigma}\phi(x)\phi(x)^\top\hat{\Sigma}}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \right) \frac{\phi(x)y}{\gamma} + \left( \hat{\Sigma} - \frac{\hat{\Sigma}\phi(x)\phi(x)^\top\hat{\Sigma}}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \right) \hat{\Sigma}^{-1}\tilde{\mu} \quad (4)$$

$$= \tilde{\mu} + \frac{\gamma\hat{\Sigma}}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \frac{\phi(x)y}{\gamma} - \frac{\hat{\Sigma}\phi(x)\phi(x)^\top}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \tilde{\mu} \quad (5)$$

$$= \tilde{\mu} + \frac{y - \tilde{\mu}^\top \phi(x)}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \hat{\Sigma}\phi(x) \quad (6)$$

となる。

次に、 $1 - \mu^\top \phi(x)y < 0$  のときを考える。

$$\frac{\partial J(\mu, \Sigma)}{\partial \mu} = 2\gamma \hat{\Sigma}^{-1}(\hat{\mu} - \tilde{\mu}) \quad (7)$$

これが0になるときを考えると、 $\hat{\mu} = \tilde{\mu}$  となる。

以上の2つの場合をまとめると、

$$\hat{\mu} = \tilde{\mu} + \frac{y \max(0, 1 - \tilde{\mu}^\top \phi(x)y)}{\phi(x)^\top\hat{\Sigma}\phi(x) + \gamma} \hat{\Sigma}\phi(x) \quad (8)$$

を得る。

### 宿題 2

Python 実装は付録にある。結果、図1を得た。

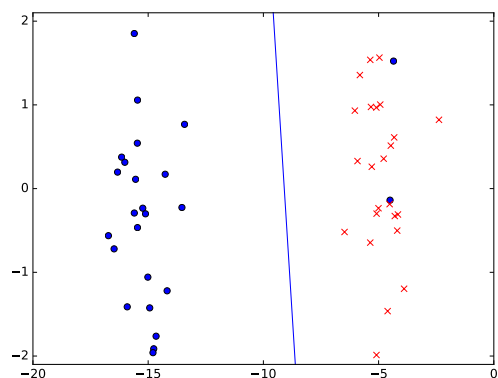


図 1

## 付録

```
import sys
import numpy as np
import numpy.matlib
import matplotlib.pyplot as plt
import scipy.io
import gc
import random

np.random.seed(42)

n = 50

# Prepare data
y = np.ones((n//2, 1))*np.array([1,-1])
y = y.reshape(y.size, 1).transpose()
y = np.ones((1, n))
y[0, n//2 + 1:] *= -1
# y = [1,...,1,-1,...,-1]
print(y.shape)
print(y)

x = np.random.randn(n, 2)
x[n//2 + 1:, 0] *= -1
x[0:n//2, 0] -= 15
x[n//2 + 1:, 0] -= 5
x[0:2, 0] += 10
x = np.hstack((x, np.ones((n, 1))))
x = x.transpose() # 3xn
print(x.shape)

mu = np.zeros((3, 1))
sigma = np.eye(3)
gamma = 0.5
indices = np.arange(n)
np.random.shuffle(indices)
for i in indices:
    beta = x[:, i].reshape(1, 3).dot(sigma).dot(x[:, i].reshape(3, 1)) + gamma
    mu += y[i]*np.maximum(0, 1-mu.transpose().dot(x[:, i])*y[i])/beta*sigma.dot(x[:, i].reshape(3, 1)).reshape(3, 1)
    print(sigma.dot(x[:, i]).reshape(3, 1).dot(x[:, i].reshape(1, 3)).shape)
    sigma -= sigma.dot(x[:, i]).reshape(3, 1).dot(x[:, i].reshape(1, 3)).dot(sigma)/beta

# print
plt.axis([-20, 0, -2.1, 2.1])
plt.plot(x[0, 0:n//2], x[1, 0:n//2], 'bo')
plt.plot(x[0, n//2 + 1:], x[1, n//2 + 1:], 'rx')
dots = np.linspace(-22, 2, 100)
plt.plot(dots, -(dots*mu[0] + mu[2])/mu[1], '-')
plt.show()
```