先端データ解析論 (杉山将先生・本多淳也先生)
第 4 回レポート

ashiato45

2017 年 5 月 9 日

宿題 1

$$e(\theta) = \frac{1}{2}\sum_{i=1}^{n}\widetilde{w}_i\left(\sum_{j=1}^{b}\theta_j\phi_j(x_i) - y_i\right)^2 \tag{1}$$

とおく。e は下に凸な二次関数なので、微分が 0 の点を求めれば最小点を求めることができる。

$J = 1, \ldots, b$ として、

$$\frac{\partial}{\partial\theta_J}e(\theta) = \sum_{i=1}^{n}\widetilde{w}_i\left(\sum_{j=1}^{b}\theta_j\phi_j(x_i) - y_i\right)\phi_J(x_i) \tag{2}$$

となるので、これが 0 になるときを考えると、次の式を得る:

$$\sum_{i=1}^{n}\widetilde{w}_i y_i\phi_J(x_i) = \sum_{i=1}^{n}\widetilde{w}_i\sum_{j=1}^{b}\theta_j\phi_j(x_i)\phi_J(x_i). \tag{3}$$

これが各 J について成立するので、

$$\Phi^\top\widetilde{W}y = \Phi^\top\widetilde{W}\Phi\theta. \tag{4}$$

よって、e を最小化する θ である $\widehat{\theta}$ は、

$$\widehat{\theta} = (\Phi^\top\widetilde{W}\Phi\theta)^{-1}\Phi^\top\widetilde{W}y. \tag{5}$$

宿題 2

対称性より、未知数 a, b を用いて $\widetilde{\rho}(r) = ar^2 + b$ とおいてよい。$\widetilde{(r)}$ は $(\widetilde{r}, \rho(\widetilde{r}))$ で $\rho(r)$ に接するので、$\widetilde{\rho}'(\widetilde{r}) = \rho'(\widetilde{r})$ と $\widetilde{\rho}(\widetilde{r}) = \rho(\widetilde{r})$ とを満たす。これは、$2a\widetilde{r} = \rho'(\widetilde{r})$ と $a\widetilde{r}^2 + b = \rho(\widetilde{r})$ を得るので、これを解いて

$$\widetilde{\rho}(r) = \frac{\rho'(\widetilde{r})}{2\widetilde{r}}r^2 + b \tag{6}$$

となる。

宿題 3

Python 実装は付録にある。$\eta = 1$ とした。結果、図 1 を得た。

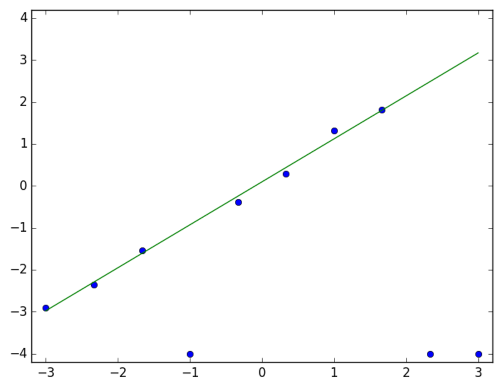

図 1

付録

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

np.random.seed(42)

datanum = 10
datasamples = np.linspace(-3, 3, datanum)
datasamples = datasamples.reshape(len(datasamples), 1)
y = datasamples + 0.2*np.random.randn(datanum, 1)
y[datanum - 1] = -4
y[datanum - 2] = -4
y[3] = -4

eta = 1
theta = np.zeros((2, 1))
phi = np.ones((datanum, 2))
phi[:, 1] = datasamples.reshape(len(datasamples))
for i in range(10000):
    r = np.abs(np.dot(phi, theta) - y)
    w = np.zeros((datanum, 1))
    w[r <= eta] = ((1-r**2/eta**2)**2)[r <= eta]
    W = np.diag(w.reshape(len(w)))
    A = np.dot(np.dot(phi.transpose(), W), phi)
    b = np.dot(np.dot(phi.transpose(), W), y)
    theta2 = np.linalg.solve(A, b)
    if np.linalg.norm(theta2 - theta) < 0.001:
        break
    theta = theta2

graphnum = 5000
graphsamples = np.linspace(-3, 3, graphnum)
graphsamples = graphsamples.reshape(graphnum, 1)
graph = theta[0, 0] + theta[1, 0]*graphsamples

plt.axis([-3.2, 3.2, -4.2, 4.2])
plt.plot(datasamples, y, 'o')
plt.plot(graphsamples, graph, '-')
plt.show()
```