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

Python 実装は付録にある。結果、図 1 を得た。

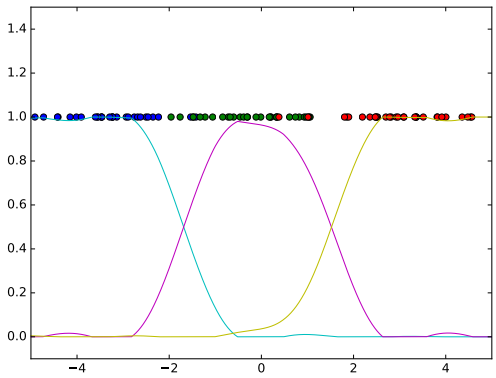

図 1

宿題 2

$$B_\tau(y^{(\tau)}) = \sum_{y^{(\tau+1)},\dots,y^{(m_i)}=1}^{c} \exp\left(\sum_{k=\tau+1}^{m_i} \zeta^\top \varphi(x_i^{(k)}, y^{(k)}, y^{(k-1)})\right) \tag{1}$$

$$= \sum_{y^{(\tau+1)},\dots,y^{(m_i)}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m_i} \zeta^\top \varphi(x_i^{(k)}, y^{(k)}, y^{(k-1)})\right) \exp(\zeta^\top \varphi(x_i^{(\tau+1)}, y^{(\tau+1)}, y^{(\tau)})) \tag{2}$$

$$= \sum_{y^{(\tau+1)}=1}^{c} \exp(\zeta^\top \varphi(x_i^{(\tau+1)}, y^{(\tau+1)}, y^{(\tau)})) \sum_{y^{(\tau+2)},\dots,y^{(m_i)}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m_i} \zeta^\top \varphi(x_i^{(k)}, y^{(k)}, y^{(k-1)})\right) \tag{3}$$

$$= \sum_{y^{(\tau+1)}=1}^{c} B_{\tau+1}(y^{(\tau+1)}) \exp(\zeta^\top \varphi(x_i^{(\tau+1)}, y^{(\tau+1)}, y^{(\tau)})). \tag{4}$$

宿題 3

$$P_\tau(y^{(\tau)}) = \max_{y^{(1)},\ldots,y^{(\tau-1)}=1,\ldots,c} \left(\sum_{k=1}^{\tau-1} \zeta^\top \varphi(x^{(k)}, y^{(k)}, y^{(k-1)}) + \zeta^\top \varphi(x^{(\tau)}, y^{(\tau)}, y^{(\tau-1)}) \right) \tag{5}$$

$$= \max_{y^{(\tau-1)}=1,\ldots,c} \left(\max_{y^{(1)},\ldots,y^{(\tau-2)}=1,\ldots,c} \left(\sum_{k=1}^{\tau-1} \zeta^\top \varphi(x^{(k)}, y^{(k)}, y^{(k-1)}) \right) + \zeta^\top \varphi(x^{(\tau)}, y^{(\tau)}, y^{(\tau-1)}) \right) \tag{6}$$

$$= \max_{y^{(\tau-1)}=1,\ldots,c} \left(P_{\tau-1}(y^{(\tau-1)}) + \zeta^\top \varphi(x^{(\tau)}, y^{(\tau)}, y^{(\tau-1)}) \right). \tag{7}$$

付録

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

np.random.seed(42)

n = 90
c = 3
h = 1
lam = 0.3

# Prepare data
y = np.ones((n//c, 1))*np.array([1,2,3])
y = y.transpose().reshape(y.size, 1)
# y = [1,..,1,2,..,2,3,..,3]
# print(y)

x = np.matlib.repmat(np.linspace(-3, 3, c), n//c, 1)
x = x.transpose().reshape(y.size, 1)
# x = [-3,..,-3,0,..,0,3,..,3]
x += np.random.randn(n, 1)
# np.linspace(-3, 3, c) = [-3, 0, 3]
print(x)

# Make pi
pi = np.zeros((n, c))
for i in range(c):
    pi[i*(n//c):(i+1)*(n//c), i] = np.ones(n//c)
# print(pi)

# Make phi
phi = np.fromfunction(lambda i, j: np.exp(-(x[i, 0] - x[j, 0])**2/(2*h*h)), (n, n), dtype=int)
# print(phi.shape)
# print(phi)

# Calc theta
A = np.dot(phi.transpose(), phi) + lam*np.eye(n)
B = np.dot(phi.transpose(), pi)
theta = np.linalg.solve(A, B)
print(theta.shape)

def gk(x_):
    res = np.zeros((n, 1), dtype=float)
    for i in range(n):
        res[i, 0] = np.exp(-(x_ - x[i, 0])**2/(2*h*h))
        # print(x_, x[i, 0], -(x_ - x[i, 0])**2/(2*h*h), res[i,0])
    return res
    # return np.fromfunction(lambda i: np.exp(-(x_ - x[i])**2/(2*h*h)), (n,), dtype=int)

def p(yy, xx):
    return np.maximum(0, np.dot(theta[:,yy].transpose(), gk(xx)))/np.sum(np.maximum(0, np.dot(theta.transpose(), gk(xx))))

# Make prediction
ptnum = 10000
pts = np.linspace(-5, 5, ptnum)
vals = np.zeros((ptnum, c), dtype=float)
test = np.zeros((ptnum,), dtype=float)
for i in range(ptnum):
    for j in range(c):
        vals[i, j] = p(j, pts[i])
        # print(theta[:,j].reshape((1, n)).shape, gk(pts[i]).shape)
        # print(pts[i], gk(pts[i])[:10])
        # vals[i,j ] = np.dot(theta[:,j].reshape((1, n)), gk(pts[i]))[0,0]
    test[i] = np.sum(np.maximum(0, np.dot(theta.transpose(), gk(pts[i]))))

a = np.fromfunction(lambda z: np.exp(((pts[z]-0)**2)/(2*h*h)*(-1)), (ptnum,), dtype=int)

print(x)
print(theta)
print(vals[0:10, :])
#print(a)
plt.axis([-5, 5, -0.1, 1.5])
plt.plot(x[0:n//c], np.ones((n//c, 1)), 'o')
plt.plot(x[n//c:2*n//c], np.ones((n//c, 1)), 'o')
plt.plot(x[2*n//c:], np.ones((n//c, 1)), 'o')
plt.plot(pts, vals[:, 0], '-')
plt.plot(pts, vals[:, 1], '-')
plt.plot(pts, vals[:, 2], '-')
# plt.plot(pts, vals[:, 0] + vals[:, 1] + vals[:, 2], '-')
# plt.plot(pts, test, '-')
```

```
# plt.plot(x, gk(2), '-') # It seems gk is working well.
# plt.plot(pts, a, '-')
plt.show()
```