先端データ解析論レポート 第 7 回

荻野 聖也（37-196323）

2019 年 5 月 30 日

宿題 1

ガウスカーネルモデル

$$q(y|\boldsymbol{x};\boldsymbol{\theta}^{(y)}) = \sum_{j:y_j=y} \theta_j^{(y)} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2h^2}\right)$$

に対して，最小二乗確率的分類を実装する。

訓練データについて

クラス 0,1,2 の計 3 クラスをそれぞれ 30 個ずつ生成する。ただし，講義の資料を参考に生成する。(図 1)

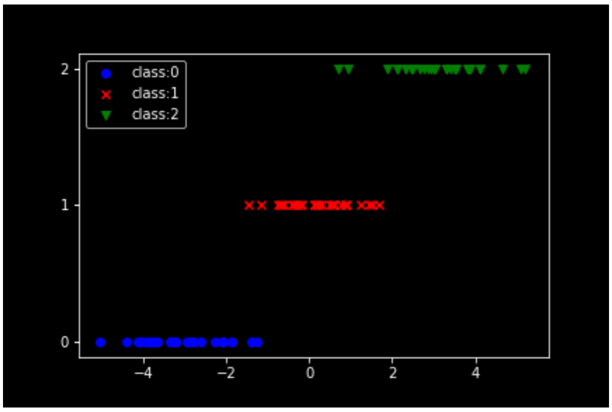

図 1　訓練データのサンプル。クラス 0,1,2 の計三種類を用意する。

最小二乗確率的分類について

あるクラス y をを線形モデルで表すとして，この係数行列 $\boldsymbol{\theta}_y$ とし，ガウスカーネルモデル由来の基底関数を $\boldsymbol{\phi}(\boldsymbol{x})$ とするとき，事後確率 $q(y|\boldsymbol{x};\boldsymbol{\theta}_y)$ は

$$q(y|\boldsymbol{x};\boldsymbol{\theta}_y) = \boldsymbol{\theta}_y^T \boldsymbol{\phi}(\boldsymbol{x})$$

となる。ただし，

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\phi_1(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x}))^T$$

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2h^2}\right)$$

である。このもとで，講義資料のように二乗誤差を標本近似を行うと，

$$J_y(\boldsymbol{\theta}_y) = \frac{1}{2n}(\boldsymbol{\theta}_y \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{\theta}_y - \frac{1}{n}\boldsymbol{\theta}_y \boldsymbol{\Phi}^T \boldsymbol{\pi}_y + \frac{\lambda}{2n}\|\boldsymbol{\theta}_y\|^2)$$

なので，最小点は

$$\hat{\boldsymbol{\theta}}_y = (\boldsymbol{\Phi}^T \boldsymbol{\Phi} + \lambda \boldsymbol{I})^{-1} \boldsymbol{\Phi}^T \boldsymbol{\pi}_y$$

で与えられる。なお，

$$\boldsymbol{\Phi} = (\boldsymbol{\phi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\phi}(\boldsymbol{x}_n))$$
$$\boldsymbol{\pi}_y = (\pi_{y,1}, \ldots, \pi_{y,n})^T$$

である。このもとで，クラス事後確率 $\hat{p}(y|\boldsymbol{x})$ は，

$$\hat{p}(y|\boldsymbol{x}) = \frac{\max(0, \hat{\boldsymbol{\theta}}_y^T \boldsymbol{\phi}(\boldsymbol{x}))}{\sum_{y'=1}^{c} \max(0, \hat{\boldsymbol{\theta}}_{y'}^T \boldsymbol{\phi}(\boldsymbol{x}))}$$

により，与えられる。

結果

結果をいかに示す。なお，数値計算については Python3.7 を使用し，コードについては補遺と添付ファイルに示す。

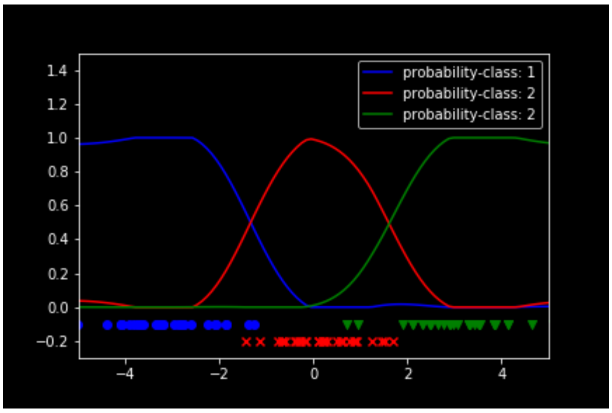

図 2　確率的最小二乗分類の結果。おおよそ，うまく分類できていることがわかる。

宿題 2

$$B_\tau(y) = \sum_{y^{(\tau+1)},\ldots,y^{(m)}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m} \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^k(y^{(k)}, y^{(k-1)}) + \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^{(\tau+1)}(y^{(\tau+1)}, y) \right)$$

について，

$$\begin{aligned} B_\tau(y) &= \sum_{y^{(\tau+1)}=1}^{c} \sum_{y^{(\tau+2)},\ldots,y^{(m)}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m} \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^k(y^{(k)}, y^{(k-1)}) \right) \exp\left(\boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^{(\tau+1)}(y^{(\tau+1)}, y) \right) \\ &= \sum_{y^{(\tau+1)}=1}^{c} \exp\left(\boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^{(\tau+1)}(y^{(\tau+1)}, y) \right) \sum_{y^{(\tau+2)},\ldots,y^{(m)}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m} \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^k(y^{(k)}, y^{(k-1)}) \right) \end{aligned}$$

ここで，

$$\begin{aligned} \sum_{y^{(\tau+2)},\ldots,y^{(m)}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m} \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^k(y^{(k)}, y^{(k-1)}) \right) &= \sum_{y^{(\tau+2)},\ldots,y^{(m)}=1}^{c} \exp\left(\sum_{k=\tau+3}^{m} \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^k(y^{(k)}, y^{(k-1)}) + \boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^{\tau+2}(y^{(\tau+2)}, y^{(\tau+1)}) \right) \\ &= B_{\tau+1}(y^{(\tau+1)}) \end{aligned}$$

なので，

$$B_\tau(y) = \sum_{y^{(\tau+1)}=1}^{c} B_{\tau+1}(y^{(\tau+1)}) \exp\left(\boldsymbol{\zeta}^T \boldsymbol{\varphi}_i^{(\tau+1)}(y^{(\tau+1)}, y) \right)$$

よって，再帰表現ができる。

補遺

Listing 1　宿題 1 のソースコード

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

np.random.seed(1)


def generate_data(sample_size=90, n_class=3):
    x = (np.random.normal(size=(sample_size // n_class, n_class))
         + np.linspace(-3., 3., n_class)).flatten()
    y = np.broadcast_to(np.arange(n_class),
                        (sample_size // n_class, n_class)).flatten()
    return x, y


def solve_LSPC(x, y, width, n_class=3, regurater=10.):最小二乗確率的分類を解く
     """
     Arg:
          x(ndarray(float)) サンプルの座標 x shape=90
          y(ndarray(int)) クラス (0,1,2)のどれか shape=90
          width(float) ガウス幅
          n_class(int) クラス数←既知として進める。
          ragurater(float) 正則化係数

     Return:
          theta(ndarray) 分類結果の係数行列
     """
     sample_size = len(x)
     Phi = np.exp(-(x[:] - x[:,None])**2 / (2 * width ** 2))
     y0 = np.where(y==0, 1, 0)
     y1 = np.where(y==1, 1, 0)
     y2 = np.where(y==2, 1, 0)
     Pi = np.array([y0,y1,y2]).T
     Theta = np.linalg.solve((Phi.T.dot(Phi)+regurater*np.eye(sample_size)), Phi.T.dot(Pi))
     return Theta

x,y = generate_data()
plt.clf()
plt.scatter(x[y==0], y[y==0], c='blue', marker="o", label="class:0")
plt.scatter(x[y==1], y[y==1], c='red', marker="x", label="class:1")
plt.scatter(x[y==2], y[y==2], c='green', marker="v", label="class:2")
ylabels=[int(0),int(1),int(2)]
plt.yticks(y, ylabels)
plt.legend()
plt.savefig("train_sample.png")
plt.show()

```

```
h = 1.
theta = solve_LSPC(x, y, h)

X = np.linspace(-5., 5., num=100)
K = np.exp(-(x - X[:, None]) ** 2 / (2 * h ** 2))
plt.clf()
plt.xlim(-5, 5)
plt.ylim(-.3, 1.5)
logit = K.dot(theta)
print(logit.shape)
zeros_mat = np.zeros_like(logit)
unnormalized_prob = np.maximum(zeros_mat, logit)
prob = unnormalized_prob / unnormalized_prob.sum(1, keepdims=True)

plt.plot(X, prob[:, 0], c='blue', label="probability-class: 1")
plt.plot(X, prob[:, 1], c='red' ,label="probability-class: 2")
plt.plot(X, prob[:, 2], c='green', label="probability-class: 3")
plt.scatter(x[y == 0], -.1 * np.ones(len(x) // 3), c='blue', marker='o')
plt.scatter(x[y == 1], -.2 * np.ones(len(x) // 3), c='red', marker='x')
plt.scatter(x[y == 2], -.1 * np.ones(len(x) // 3), c='green', marker='v')
plt.legend()
plt.savefig("LSPC.png")
plt.show()
```