先端データ解析論
第七回　レポート

37-196360　森田涼介

2019 年 6 月 2 日

宿題 1

ガウスカーネルモデルに対して，最小二乗確率的分類を実装する。

$$q\left(y \mid \boldsymbol{x};\, \boldsymbol{\theta}^{(y)}\right) = \sum_{j:y_j=y} \theta_j^{(y)} K(\boldsymbol{x},\, \boldsymbol{x}_j) \tag{1}$$

$$K(\boldsymbol{x},\, \boldsymbol{c}) = \exp\left(-\frac{||\boldsymbol{x}-\boldsymbol{c}||^2}{2h^2}\right) \tag{2}$$

二乗誤差は次のようになる。

$$J_y(\boldsymbol{\theta}_y) = \frac{1}{2}\int (q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y) - p(y \mid \boldsymbol{x}))^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{3}$$

$$= \frac{1}{2}\int (q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y))^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} - \int q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y)p(y|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} + \frac{1}{2}\int p(y|\boldsymbol{x})^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{4}$$

これを標本近似することを考える。第一項は，

$$\frac{1}{2}\int (q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y))^2 p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \;\to\; \frac{1}{2}\cdot\frac{1}{n}\sum_{i=1}^{n} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y)^2 \tag{5}$$

第二項は，

$$\int q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y)p(y|\boldsymbol{x})p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \int q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y)p(y)p(\boldsymbol{x}|y)\mathrm{d}\boldsymbol{x} \tag{6}$$

$$= p(y)\int q(y|\boldsymbol{x};\, \boldsymbol{\theta}_y)p(\boldsymbol{x}|y)\mathrm{d}\boldsymbol{x} \tag{7}$$

$$\to \frac{n_y}{n}\cdot\frac{1}{n_y}\sum_{i:y_i=y} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y) \tag{8}$$

$$= \frac{1}{n}\sum_{i:y_i=y} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y) \tag{9}$$

第三項は定数なので無視できる。以上に L2 正則化項を加えると，二乗誤差は，

$$\hat{J}_y(\boldsymbol{\theta}_y) = \frac{1}{2n}\sum_{i=1}^{n} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y)^2 - \frac{1}{n}\sum_{i:y_i=y} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y) + \frac{\lambda}{2n}||\boldsymbol{\theta}_y||^2 \tag{10}$$

$$= \frac{1}{n}\left(\frac{1}{2}\sum_{i=1}^{n} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y)^2 - \sum_{i:y_i=y} q(y|\boldsymbol{x}_i;\, \boldsymbol{\theta}_y) + \frac{\lambda}{2}||\boldsymbol{\theta}_y||^2\right) \tag{11}$$

と表せる。

いま，ガウスカーネルモデルを考えると，カーネル行列

$$\boldsymbol{K} = \begin{bmatrix} K(\boldsymbol{x}_1,\, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_1,\, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n,\, \boldsymbol{x}_1) & \cdots & K(\boldsymbol{x}_n,\, \boldsymbol{x}_n) \end{bmatrix} \tag{12}$$

を用いれば，二乗誤差は次のように表せる。

$$\hat{J}_y(\boldsymbol{\theta}_y) = \frac{1}{n}\left\{\frac{1}{2}\boldsymbol{\theta}_y^{\mathrm{T}}\boldsymbol{K}^2\boldsymbol{\theta}_y - \boldsymbol{\theta}_y^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\pi}_y + \frac{\lambda}{2}||\boldsymbol{\theta}_y||^2\right\} \tag{13}$$

ここで，$\boldsymbol{K}$ は対称行列であることを利用した。また，

$$\pi_{y,i} = \begin{cases} 1 & (y_i = y) \\ 0 & (y_i \neq y) \end{cases} \tag{14}$$

である。$\boldsymbol{\theta}_y$ による偏微分は，

$$\frac{\partial \hat{J}_y}{\partial \boldsymbol{\theta}_y} = \frac{1}{n}\left\{\boldsymbol{K}^2\boldsymbol{\theta}_y - \boldsymbol{K}\boldsymbol{\pi}_y + \lambda\boldsymbol{\theta}_y\right\} \tag{15}$$

$$= \frac{1}{n}\left\{\left(\boldsymbol{K}^2 + \lambda\boldsymbol{I}\right)\boldsymbol{\theta}_y - \boldsymbol{K}\boldsymbol{\pi}_y\right\} \tag{16}$$

これを $\mathbf{0}$ にするような $\boldsymbol{\theta}_y$ は，

$$\hat{\boldsymbol{\theta}}_y = \left(\boldsymbol{K}^2 + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{K}\boldsymbol{\pi}_y \tag{17}$$

となる。プログラムでは，この式を用いてパラメータを求める。

バンド幅 $h = 2.0$，L2 正則化パラメータ $\lambda = 0.0001$ に対する結果を，以下の図 1 に示す。なお，プログラムは 4 ページの Listing 1 に示した。

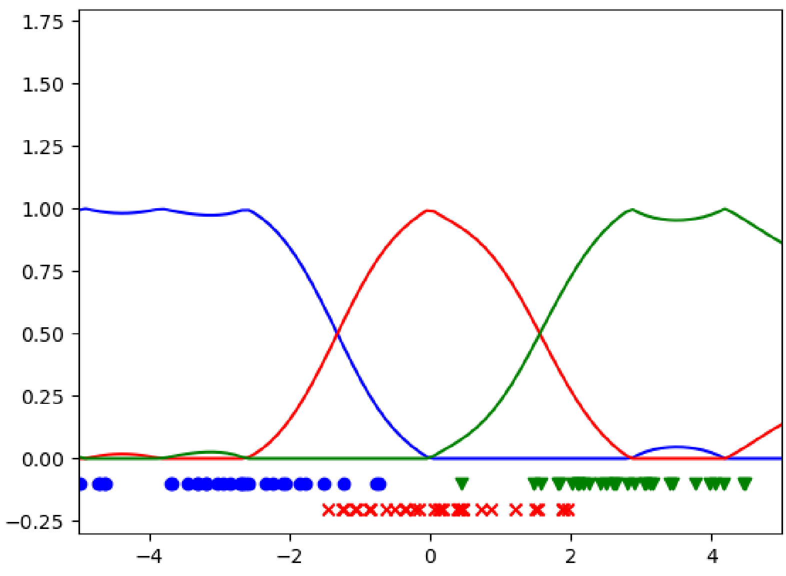

図 1: 結果

宿題 2

$B_\tau(y)$ を再帰的に表現する。

$$B_\tau(y) = \sum_{y^{(\tau+1)},\dots,y^{(\mathrm{m})}=1}^{c} \exp\left(\sum_{k=\tau+2}^{m} \boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(k)}\left(y^{(k)},\, y^{(k-1)}\right) + \boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(\tau+1)}\left(y^{(\tau+1)},\, y\right)\right) \tag{18}$$

$$= \sum_{y^{(}\tau+1)=1}^{c} \cdot \sum_{y^{(\tau+2)},\dots,y^{(\mathrm{m})}=1}^{c} \exp\left\{\sum_{k=\tau+2}^{m} \boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(k)}\left(y^{(k)},\, y^{(k-1)}\right)\right\} \exp\left(\boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(\tau+1)}\left(y^{(\tau+1)},\, y\right)\right) \tag{19}$$

ここで,

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

$$= \sum_{y^{(\tau+2)},\dots,y^{(\mathrm{m})}=1}^{c} \exp\left\{\sum_{k=\tau+3}^{m} \boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(k)}\left(y^{(k)},\, y^{(k-1)}\right) + \boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(\tau+2)}\left(y^{(\tau+2)},\, y^{(\tau+1)}\right)\right\} \tag{20}$$

$$= B_{\tau+1}\left(y^{(\tau+1)}\right) \tag{21}$$

となるから，結局 $B_\tau(y)$ は,

$$B_\tau(y) = \sum_{y^{(}\tau+1)=1}^{c} B_{\tau+1}\left(y^{(\tau+1)}\right) \exp\left(\boldsymbol{\zeta}^{\mathrm{T}} \boldsymbol{\varphi}_i^{(\tau+1)}\left(y^{(\tau+1)},\, y\right)\right) \tag{22}$$

という形で再帰表現される。

プログラム

実行環境と用いた言語・ライブラリを以下の表 1 に示す。

表 1: プログラムの実行環境

OS	:	Microsoft Windows 10 Pro (64bit)
CPU	:	Intel(R) Core(TM) i5-4300U
RAM	:	4.00 GB
使用言語	:	Python3.6
可視化	:	matplotlib ライブラリ

Listings 1: assignment1.py

```
# -*- coding: utf-8 -*-


import numpy as np
import matplotlib.pyplot as plt


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 train(x, y, h, lamb, n_class):
    n_sample = x.shape[0]
    theta = np.zeros((n_sample, n_class))
    K = np.exp(-(x - x[:, None])**2 / (2*h**2))
    for label in range(n_class):
        pi_y = (y == label).astype(int)
        theta[:, label] = np.linalg.inv(K.T.dot(K) +
    lamb*np.eye(n_sample)).dot(K).dot(pi_y)
    return theta


def visualize(x, y, theta, h, num=100, path=None):
    X = np.linspace(-5, 5, num=num)
```

```
    K = np.exp(-(x - X[:, None])**2 / (2*h**2))
    logit = K.dot(theta)
    unnormalized_prob = logit - logit.min(axis=1, keepdims=True)
    prob = unnormalized_prob / unnormalized_prob.sum(axis=1, keepdims=True)

    plt.clf()
    plt.xlim(-5, 5)
    plt.ylim(-.3, 1.8)

    plt.plot(X, prob[:, 0], c='blue')
    plt.plot(X, prob[:, 1], c='red')
    plt.plot(X, prob[:, 2], c='green')

    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')

    if path:
        plt.savefig(path)
    plt.show()


def main():
    # settings
    n_sample = 90
    n_class = 3
    h = 2
    lamb =  1e-4
    fig_path = '../figures/assignment1_result.png'
    np.random.seed(0)

    # load data
    x, y = generate_data(n_sample, n_class)
    #print(x)
    #print(y)

    # train
    theta = train(x, y, h, lamb, n_class)

    # result
    print(f'#Sample: {n_sample}    #Class: {n_class}')
    print(f'h = {h}    lambda = {lamb}')
    print(f'theta: \n{theta}')
    visualize(x, y, theta, h, path=fig_path)


if __name__ == '__main__':
```

```
    main()
```