先端データ解析論
第四回　レポート

37-196360　森田涼介

2019 年 5 月 7 日

宿題 1

線形モデル

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{j=1}^{b} \theta_j \phi_j(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{\theta} \tag{1}$$

に対して，重み付き最小二乗法を考える。ここで，

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \cdots & \theta_b \end{bmatrix}^{\mathrm{T}} \tag{2}$$

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) & \cdots & \phi_b(\boldsymbol{x}) \end{bmatrix}^{\mathrm{T}} \tag{3}$$

である。このとき，損失関数は以下のように表される。

$$\begin{aligned}
J(\boldsymbol{\theta}) &= \frac{1}{2} \sum_{i=1}^{n} \tilde{w}_i (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i)^2 & (4)\\
&= \frac{1}{2} \sum_{i=1}^{n} \tilde{w}_i \left(\boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\theta} - y_i\right)^2 & (5)\\
&= \frac{1}{2} \sum_{i=1}^{n} \left(\boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\theta} - y_i\right) \tilde{w}_i \left(\boldsymbol{\phi}(\boldsymbol{x}_i)^{\mathrm{T}} \boldsymbol{\theta} - y_i\right) & (6)\\
&= \frac{1}{2} (\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y})^{\mathrm{T}} \tilde{\boldsymbol{W}} (\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}) & (7)\\
&= \frac{1}{2} (\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} - \boldsymbol{y}^{\mathrm{T}}) \tilde{\boldsymbol{W}} (\boldsymbol{\Phi}\boldsymbol{\theta} - \boldsymbol{y}) & (8)\\
&= \frac{1}{2} \left\{ \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi} \boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} - \boldsymbol{y}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi} \boldsymbol{\theta} + \boldsymbol{y}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} \right\} & (9)
\end{aligned}$$

ここで，

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \cdots & \phi_b(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_n) & \cdots & \phi_b(\boldsymbol{x}_n) \end{bmatrix} \tag{10}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^{\mathrm{T}} \tag{11}$$

$$\tilde{\boldsymbol{W}} = \mathrm{diag}\{\tilde{w}_1, \cdots, \tilde{w}_n\} \tag{12}$$

である。また，$\boldsymbol{y}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi} \boldsymbol{\theta}$ はスカラーであるから転置しても値は変わらず，$\tilde{\boldsymbol{W}}$ は対称行列であるので転置しても $\tilde{\boldsymbol{W}}$ のままであることに注意すると，結局，損失関数は次のように表される。

$$J(\boldsymbol{\theta}) = \frac{1}{2} \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi} \boldsymbol{\theta} - \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} + \frac{1}{2} \boldsymbol{y}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} \tag{13}$$

これを $\boldsymbol{\theta}$ で偏微分すると，

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi} \boldsymbol{\theta} - \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} \tag{14}$$

となる。これを $\boldsymbol{0}$ とするような $\boldsymbol{\theta}$ が J に最小を与えるので，$\boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi}$ の逆行列が存在することを仮定すると，

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} \tag{15}$$

となる。

宿題 2

微分可能で対称な損失 $\rho(r)$ に対して $\tilde{r}$ で接する二次上界は，（存在するなら）次式で与えられることを示す。

$$\tilde{\rho}(r) = \frac{\tilde{w}}{2}r^2 + \text{Const.} \qquad \left(\tilde{w} = \frac{\rho'(\tilde{r})}{\tilde{r}}\right) \tag{16}$$

微分可能で対称な損失 $\rho(r)$ に対して $\tilde{r}$ で接する二次上界を，

$$\tilde{\rho}(r) = ar^2 + br + c \tag{17}$$

とおく。$\rho(r)$ は対称より，$b = 0$ がわかる。このとき，$\tilde{\rho}(r)$ は $\rho(r)$ に $\tilde{r}$ で接するので，自身の値及び 1 階微分の値が $\tilde{r}$ で一致する。したがって，

$$\tilde{\rho}(\tilde{r}) = a\tilde{r}^2 + c = \rho(\tilde{r}) \tag{18}$$

$$\tilde{\rho}'(\tilde{r}) = 2a\tilde{r} = \rho'(\tilde{r}) \tag{19}$$

これを解くと，

$$a = \frac{\rho'(\tilde{r})}{2\tilde{r}} \tag{20}$$

$$c = \rho(\tilde{r}) - \frac{\rho'(\tilde{r})}{2}\tilde{r} \tag{21}$$

となるので，結局，

$$\tilde{\rho}(r) = \frac{\rho'(\tilde{r})}{2\tilde{r}}r + \left(\rho(\tilde{r}) - \frac{\rho'(\tilde{r})}{2}\tilde{r}\right) \tag{22}$$

$$= \frac{\tilde{w}}{2}r^2 + \text{Const.} \qquad \left(\tilde{w} = \frac{\rho'(\tilde{r})}{\tilde{r}}\right) \tag{23}$$

と表せることがわかる。

宿題 3

直線モデル $f_{\boldsymbol{\theta}}(x) = \theta_1 + \theta_2 x$ に対してテューキー回帰を行う。

理論

微分可能で対称な損失 $\rho(r)$ に対して $\tilde{r}$ で接する二次上界は，（存在するなら）次式で与えられる。

$$\tilde{\rho}(r) = \frac{\tilde{w}}{2} r + \text{Const.} \qquad \left(\tilde{w} = \frac{\rho'(\tilde{r})}{\tilde{r}}\right) \tag{24}$$

これより，損失は次のように表される。

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \rho(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i) \tag{25}$$

また，J の上界は次式で表される。

$$\begin{aligned} \tilde{J}(\boldsymbol{\theta}) &= \frac{1}{2} \sum_{i=1}^{n} \tilde{w}_i (f_{\boldsymbol{\theta}}(\boldsymbol{x}_i) - y_i) & (26) \\ &= \frac{1}{2} (\boldsymbol{\Phi\theta} - \boldsymbol{y})^{\mathrm{T}} \tilde{\boldsymbol{W}} (\boldsymbol{\Phi\theta} - \boldsymbol{y}) & (27) \end{aligned}$$

ここで，

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \cdots & \phi_b(\boldsymbol{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\boldsymbol{x}_n) & \cdots & \phi_b(\boldsymbol{x}_n) \end{bmatrix} \tag{28}$$

$$\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}^{\mathrm{T}} \tag{29}$$

$$\tilde{\boldsymbol{W}} = \mathrm{diag}\{\tilde{w}_1, \cdots, \tilde{w}_n\} \tag{30}$$

である。これを最小にするパラメータは，$\boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi}$ の逆行列が存在することを仮定すると次式で与えられる。

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{\Phi}\right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}} \tilde{\boldsymbol{W}} \boldsymbol{y} \tag{31}$$

テューキー損失は次式で表される。

$$\rho(r) = \begin{cases} \frac{1}{6}\left\{1 - (1 - (r/\eta)^2)^3\right\} & (|r| \leq \eta) \\ \frac{1}{6} & (|r| > \eta) \end{cases} \tag{32}$$

これより，重みは次のように表される。

$$\tilde{w} = \begin{cases} \left(1 - \left(\frac{r}{\eta}\right)^2\right)^2 & (|r| \leq \eta) \\ 0 & (|r| > \eta) \end{cases} \tag{33}$$

以上より，以下の手順でパラメータを求めればよい。

1. $\boldsymbol{\theta}$ を適当に初期化する
2. 式 (33) を用いて $\tilde{\boldsymbol{W}}$ を計算する
3. 式 (31) を用いて $\boldsymbol{\theta}$ を更新する
4. 2，3 を収束するまで繰り返す

表 1: 結果

η	θ_1 (initial)	θ_2 (initial)	θ_1	θ_2	loss
0.5	0.4427	0.9095	0.5170	1.024	3.759
1.0	0.8966	0.2608	0.1900	1.011	0.9676
1.5	0.9991	0.4859	0.2455	0.9955	0.5501

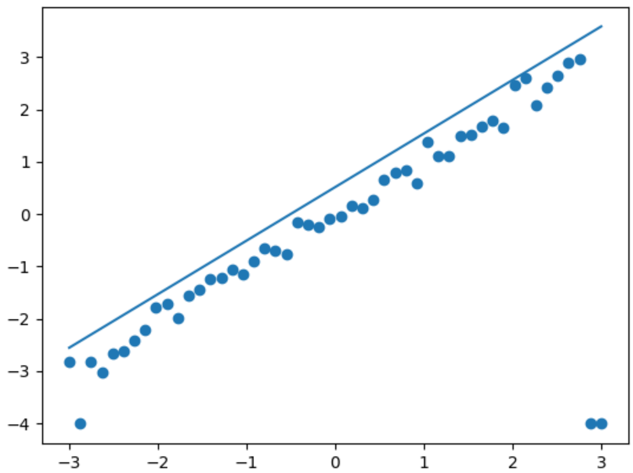

図 1: $\eta = 0.5$ のときの実データと直線モデルのプロット

結果

$\eta = 0.5,\ 1.0,\ 1.5$ についての結果を示す。表 1 に，$\boldsymbol{\theta}$ の初期値と最終的な値，また収束時の損失を示した。また，図 1–3 にプロットを示す。なお，$\boldsymbol{\theta}$ は $[0,\ 1)$ の一様分布で初期化した。また，プログラムは 6 ページの Listing 1 に示した。

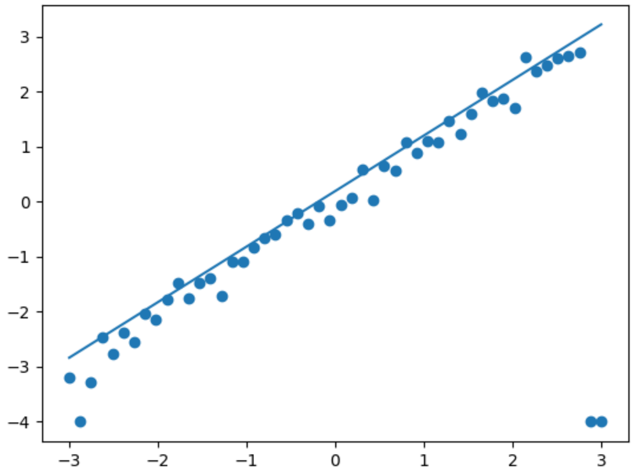

図 2: $\eta = 1.0$ のときの実データと直線モデルのプロット

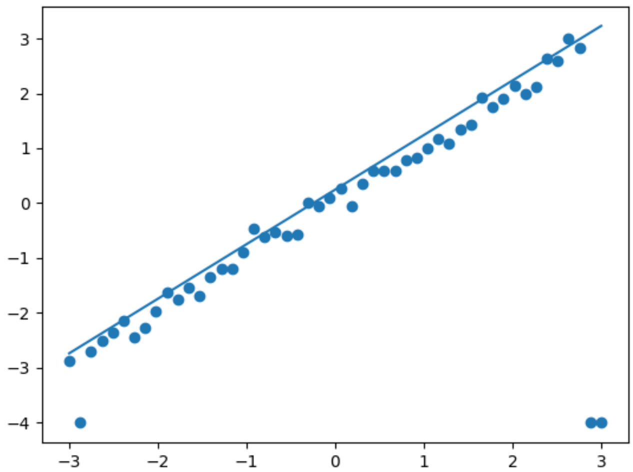

図 3: $\eta = 1.5$ のときの実データと直線モデルのプロット

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 2 に示す。

表 2: 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: assignment3.py

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


import numpy as np
import matplotlib.pyplot as plt


def generate_sample(x_min=-3., x_max=3., sample_size=10):
    x = np.linspace(x_min, x_max, num=sample_size)
    y = x + np.random.normal(loc=0., scale=.2, size=sample_size)
    y[-1] = y[-2] = y[1] = -4  # outliers
    return x, y


def model(x, theta):
    f = theta[0] + theta[1] * x
    return f


def turkey_loss(r, eta):
    rho = (1 - (1 - (r/eta)**2)**3)/6
    rho[np.abs(r) > eta] = 1/6
    loss = (1/2) * np.sum(rho)
    return loss


def compute_loss(x, y, theta, eta):
    y_pred = model(x, theta)
    r = y_pred - y
    loss = turkey_loss(r, eta)
    return loss

```

```

def calc_Phi(x, theta):
    n = x.shape[0]
    b = theta.shape[0]
    phi = np.zeros((n, b))
    phi[:, 0] = theta[0]
    phi[:, 1] = theta[1] * x
    return phi


def calc_W(r, eta):
    w = (1 - (r/eta)**2)**2
    w[np.abs(r) > eta] = 0
    w = np.diag(w)
    return w


def update(theta, phi, y, w):
    theta = np.linalg.inv(phi.T.dot(w).dot(phi)).dot(phi.T).dot(w).dot(y)
    return theta


def solve(x, y, theta_initial, eta, eps=1e-4, n=5, max_iter=100):
    diffs = []
    theta = theta_initial
    for i in range(max_iter):
        theta_old = theta.copy()
        phi = calc_Phi(x, theta)
        y_pred = model(x, theta)
        r = y_pred - y
        w = calc_W(r, eta)
        theta = update(theta, phi, y, w)
        diff = np.linalg.norm(theta - theta_old)
        if len(diffs) < n:
            diffs.append(diff)
        else:
            if (max(diffs) - min(diffs)) < eps:
                break
            diffs = diffs[1:] + [diff]
    n_iter = i + 1
    return theta, n_iter


def main():
    #np.random.seed(0)  # set the random seed for reproducibility

    # create sample
```

```
    x_min, x_max = -3, 3
    sample_size = 50
    x, y = generate_sample(x_min, x_max, sample_size)
    # print(x.shape, y.shape)

    # hyper parameter
    eta = 1.5

    # parameter
    theta_init = np.random.rand(2)

    # solve
    theta, n_iter = solve(x, y, theta_init, eta, eps=1e-4, n=5, max_iter=100)

    # calc loss
    loss = compute_loss(x, y, theta, eta)

    # result
    print('eta: {}'.format(eta))
    print('theta_init: {}'.format(theta_init))
    print('theta: {}'.format(theta))
    print('loss: {:.4f}'.format(loss))
    print('n_iter: {}'.format(n_iter))

    # plot
    x_axis = np.linspace(x_min, x_max, 100)
    plt.scatter(x, y)
    plt.plot(x_axis, model(x_axis, theta))
    plt.savefig('../figures/assignment3_result_eta{}.png'.format(int(10*eta)))
    plt.show()


if __name__ == '__main__':
    main()
```