先端データ解析論レポート 第 2 回

荻野 聖也（37-196323）

2019 年 4 月 29 日

課題 1

関数，

$$T(z) = \lambda|z| + u(\theta - z) + \frac{1}{2}(\theta - z)^2$$

について，

$$\theta - z \equiv Z$$

とするとき，$T(z) \equiv U(Z)$ にたいして，

$$U(Z) = \frac{1}{2}Z^2 + uZ + \lambda|\theta - Z|$$

この関数は微分不可能な点が存在しうるので，場合分けにより最小点を求めることにする。
(i)$\theta - Z \geq 0$ すなわち，$Z \leq \theta$ のとき，

$$\begin{aligned} U(Z) &= \frac{1}{2}Z^2 + uZ + \lambda(\theta - Z) \\ &= \frac{1}{2}Z^2 + (u - \lambda)Z + \lambda\theta \\ &= \frac{1}{2}(Z + (u - \lambda))^2 - \frac{1}{2}(u - \lambda)^2 + \lambda\theta \end{aligned}$$

一方，(ii)$\theta - Z \leq 0$ すなわち，$\theta \leq Z$ について，

$$\begin{aligned} U(Z) &= \frac{1}{2}Z^2 + uZ - \lambda(\theta - Z) \\ &= \frac{1}{2}(Z + (u + \lambda))^2 - \frac{1}{2}(u + \lambda)^2 - \lambda\theta \end{aligned}$$

となる。当然それぞれの関数は放物線であるが，軸の値に対して，極小点 (すなわち最小点) が変わってくる。

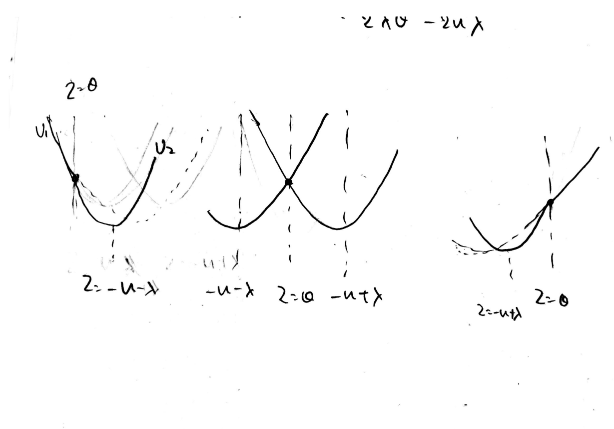

図 1　軸による場合分け。境界 $Z = \theta$ について U は連続なので高々 3 通りの場合を考えれば十分。(図が汚くてすいません。)

(i)$\theta \leq -u-\lambda$ の時,
最小値は，$Z=-u-\lambda$ のとき

(ii)$-u-\lambda \leq \theta \leq -u+\lambda$ の時,
最小値は,$Z=\theta$ のときとる。

(iii)$-u+\lambda \leq \theta$ のとき，
最小値は $Z=-u+\lambda$ の時，以上をまとめると，

$$\arg\min_{Z} U(Z) = \begin{cases} -u-\lambda & (\theta \leq -u-\lambda) \\ \theta & (-u-\lambda \leq \theta \leq -u+\lambda) \\ -u+\lambda & (-u+\lambda \leq \theta) \end{cases}$$

ここで，$z=\theta-Z$ より，

$$\arg\min_{z} T(z) = \begin{cases} \theta+u+\lambda & (\theta \leq -u-\lambda) \\ 0 & (-u-\lambda \leq \theta \leq -u+\lambda) \\ \theta+u-\lambda & (-u+\lambda \leq \theta) \end{cases}$$

ここで，$\lambda \geq 0$ より，

$$u+\theta-\lambda \leq u+\theta+\lambda$$

は常に成立する。このことから

$$\arg\min_{z} T(z) = \min(0, \theta+u++\lambda) + \max(0, \theta+u-\lambda)$$

宿題 2

l_1-正則化最小二乗法の目的関数，$f(\boldsymbol{\theta})$，$g(\boldsymbol{z})$ はそれぞれ，

$$\begin{cases} f(\boldsymbol{\theta}) = \|\boldsymbol{K}\boldsymbol{\theta}-\boldsymbol{y}\|^2 \\ g(\boldsymbol{z}) = \lambda\|\boldsymbol{z}\|_1 \end{cases}$$

とあらわされる。ただし，

$$\boldsymbol{K} \equiv \begin{pmatrix} K(\boldsymbol{x}_1, \boldsymbol{x}_1) & \dots & K(\boldsymbol{x}_1, \boldsymbol{x}_n) \\ \vdots & \ddots & \vdots \\ K(\boldsymbol{x}_n, \boldsymbol{x}_1) & \dots & K(\boldsymbol{x}_n, \boldsymbol{x}_n) \end{pmatrix}$$

このとき，拡張ラグランジュ関数 $\boldsymbol{L}(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u})$ は，

$$\boldsymbol{L}(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{K}\boldsymbol{\theta}-\boldsymbol{y}\|^2 + \lambda\|\boldsymbol{z}\|_1 + \boldsymbol{u}^T(\boldsymbol{A}\boldsymbol{\theta}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}) + \frac{1}{2}\|\boldsymbol{A}\boldsymbol{\theta}+\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}\|^2$$

解の更新式は，

$$\begin{cases} \boldsymbol{\theta}^{(k+1)} = \arg\min_{\boldsymbol{\theta}} \boldsymbol{L}(\boldsymbol{\theta}, \boldsymbol{z}^{(k)}, \boldsymbol{u}^{(k)}) \\ \boldsymbol{z}^{(k+1)} = \arg\min_{\boldsymbol{z}} \boldsymbol{L}(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{z}, \boldsymbol{u}^{(k)}) \\ \boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{A}\boldsymbol{\theta}^{(k+1)} + \boldsymbol{B}\boldsymbol{z}^{(k+1)} - \boldsymbol{c} \end{cases}$$

であらわされる。$\boldsymbol{\theta}$ について，

$$\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{\theta}} = \boldsymbol{K}^T\boldsymbol{K}\boldsymbol{\theta} - \boldsymbol{K}^T\boldsymbol{y} + \boldsymbol{A}^T\boldsymbol{u} + \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{\theta} + \boldsymbol{A}^T(\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}) = \boldsymbol{0}$$

を解くと，$\boldsymbol{K}^T = \boldsymbol{K}$ より，

$$\boldsymbol{\theta} = (\boldsymbol{K}^2 + \boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{K}^T\boldsymbol{y} - \boldsymbol{A}^T\boldsymbol{u} - \boldsymbol{A}^T(\boldsymbol{B}\boldsymbol{z}-\boldsymbol{c}))$$

以下，講義同様に

$$\begin{cases} \boldsymbol{A} = \boldsymbol{I} \\ \boldsymbol{B} = -\boldsymbol{I} \\ \boldsymbol{c} = \boldsymbol{0} \end{cases}$$

とすると，

$$\boldsymbol{\theta} = (\boldsymbol{K}^2 + \boldsymbol{I})^{-1}(\boldsymbol{K}\boldsymbol{y} + \boldsymbol{z} - \boldsymbol{u})$$

なので，

$$\begin{aligned} \boldsymbol{\theta}^{(k+1)} &= \arg\min_{\boldsymbol{\theta}} \boldsymbol{L}(\boldsymbol{\theta}, \boldsymbol{z}^{(k)}, \boldsymbol{u}^{(k)}) \\ &= (\boldsymbol{K}^2 + \boldsymbol{I})^{-1}(\boldsymbol{K}\boldsymbol{y} + \boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}) \end{aligned}$$

また，$\boldsymbol{z}$ に対して

$$\boldsymbol{L} = \frac{1}{2}\|\boldsymbol{K}\boldsymbol{\theta} - \boldsymbol{y}\|^2 + \lambda\|\boldsymbol{z}\|_1 + \boldsymbol{u}^T(\boldsymbol{\theta} - \boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2$$

の最小点を求める。

$$\arg\min_{\boldsymbol{z}} \boldsymbol{L} = \arg\min_{\boldsymbol{z}} \left\{ \lambda\|\boldsymbol{z}\|_1 + \boldsymbol{u}^T(\boldsymbol{\theta} - \boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{z}\|^2 \right\}$$

各基底（添え字 i とする）成分について，

$$\arg\min_{z_i} \left\{ \lambda|z_i| + u_i(\theta_i - z_i) + \frac{1}{2}(\theta_i - z_i)^2 \right\} = \max(0, \theta_i + u_i - \lambda) + \min(0, \theta_i + u_i - \lambda)$$

なので（ただし，宿題 1 の結果を用いた。），

$$\arg\min_{\boldsymbol{z}} \boldsymbol{L} = \max(0, \boldsymbol{\theta} + \boldsymbol{u} - \lambda\boldsymbol{1}) + \min(0, \boldsymbol{\theta} + \boldsymbol{u} - \lambda\boldsymbol{1})$$

これより，$\boldsymbol{z}$ に関する更新式は

$$\begin{aligned} \boldsymbol{z}^{(k+1)} &= \arg\min_{\boldsymbol{z}} \boldsymbol{L}(\boldsymbol{\theta}^{(k+1)}, \boldsymbol{z}, \boldsymbol{u}^{(k)}) \\ &= \max(0, \boldsymbol{\theta}^{(k+1)} + \boldsymbol{u}^{(k)} - \lambda\boldsymbol{1}) + \min(0, \boldsymbol{\theta}^{(k+1)} + \boldsymbol{u}^{(k)} - \lambda\boldsymbol{1}) \end{aligned}$$

最後に，$\boldsymbol{u}$ についてであるが，これは更新された $\boldsymbol{\theta}^{(k+1)}$ および $\boldsymbol{z}^{(k+1)}$ を用いて，

$$\boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{\theta}^{(k+1)} - \boldsymbol{z}^{(k+1)}$$

以上をまとめると，

$$\begin{cases} \boldsymbol{\theta}^{(k+1)} = (\boldsymbol{K}^2 + \boldsymbol{I})^{-1}(\boldsymbol{K}\boldsymbol{y} + \boldsymbol{z}^{(k)} - \boldsymbol{u}^{(k)}) \\ \boldsymbol{z}^{(k+1)} = \max(0, \boldsymbol{\theta}^{(k+1)} + \boldsymbol{u}^{(k)} - \lambda\boldsymbol{1}) + \min(0, \boldsymbol{\theta}^{(k+1)} + \boldsymbol{u}^{(k)} - \lambda\boldsymbol{1}) \\ \boldsymbol{u}^{(k+1)} = \boldsymbol{u}^{(k)} + \boldsymbol{\theta}^{(k+1)} - \boldsymbol{z}^{(k+1)} \end{cases}$$

データについて

データは前回のレポート同様に，

$$f(x) = \frac{\sin(\pi x)}{\pi x} + 0.1x$$

を真の関数とし，サンプル $\{(x_i, y_i\}_{i=1}^n$ は，

$$y_i = f(x_i) + \varepsilon_i$$

ただし，ノイズ ε_i は正規分布

$$\varepsilon_i \sim N(0, \sigma^2)$$

に従うものとする。サンプル数 n は，$n = 50$ とする。

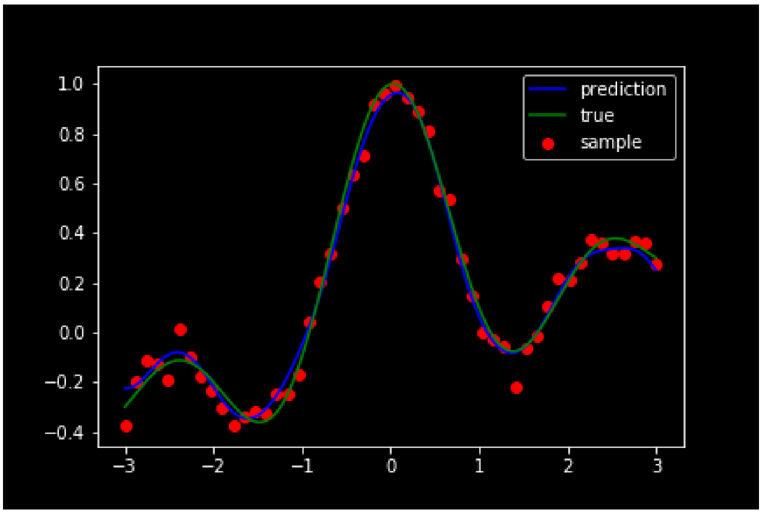

図 2　スパース回帰による実行結果

結果

実行結果を図 2 に示す。コードについては補填か添付したファイルを参考にしてください。

補填

コードについて示す。但し，使用した言語は Python3.7.1 である。

Listing 1　宿題 2 のソースコード

```
##Requirement
import numpy as np
import matplotlib.pyplot as plt
import random

def func(x, with_noise=True):
    """基本方程式
    Arg:
        x(float) 座標 x
        with_noise(boolian) ノイズ入れるかどうか
    Return:
        fx(float) 座標に対応する xy(with_noise?)
    """
    noise = 0.05 * np.random.normal(loc=0., scale=1.) if with_noise else 0
    return np.sin(np.pi*x)/np.pi/x+0.1*x + noise

def get_sample(x_min=-3., x_max=3., num_sample=50):
    """サンプル生成
    Arg:
        x_min(float) 座標の最小値 x default=-3.
        x_max(float) 座標の最大値 x default=3.
        num_sample(int) サンプルの生成数 (=dim(theta)) default=50

    Return:
        x_lst(list(float)) 座標の集合 x
        fx_lst(list(float)) 座標の集合 y
    """
    x_lst = np.linspace(x_min, x_max, num_sample)
    fx_lst = np.array([func(x, with_noise=True) for x in x_lst])
    return x_lst, fx_lst
```

```
def get_gauss_kernel(x, c, h):
    return np.exp(-(x-c)**2/2/h**2)

def calc_admm(zk, uk, K_mat, y_vec, lam):
    """反復法により交互方向乗数法の更新式を解く
    Arg:
        zk(ndarray(float)) 前状態の z
        uk(ndarray(float)) 前状態の u
        K_mat(ndarray(ndrray(float))) ガウスカーネル行列←サンプルに依存 x
        y_vec(ndarray(float)) サンプル fx
        lam(float) 正則化パラメータ
    Return:
        theta_next(ndarray(float)) 次状態の theta
        u_next(ndarray(float)) 次状態の u
        z_next(ndarray(float)) 次状態の z
    """
    n = len(y_vec)
    I = np.eye(n)
    theta_next = np.dot(np.linalg.inv((np.dot(K_mat,K_mat)+I)), np.dot(K_mat,y_vec)+zk-uk)
    max_factor = np.vstack((np.zeros(n), theta_next+uk-lam*np.ones(n)))
    min_factor = np.vstack((np.zeros(n), theta_next+uk+lam*np.ones(n)))
    z_next = np.max(max_factor,axis=0) + np.min(min_factor,axis=0)
    u_next = uk+theta_next-z_next
    return theta_next, z_next, u_next

def calc_lasso(X, Y, h, lam):
    """スパース回帰を解く
    Arg:
        X(ndarray(float)) サンプルの座標の集合 x
        Y(ndarray(float)) サンプルの座標の集合 y
        h(float) ガウス幅
        lam(float) 正則化パラメータ
        z0(float) の初期状態 z default=0
        uo(float) の初期状態 u default=0

    Return:
        x_pred(ndarray(float)) 予測系の座標←予測しているわけではない x
        y_pred(ndarray(float)) 予測された fx(座標に対応 x)
    """
    n = len(Y)初期化
    #
    zk = np.zeros(n)
    uk = np.zeros(n)
    k_matrix = np.empty((n,n))ガウスカーネルマトリックス
    #
    for i in range(len(k_matrix)):
        for j in range(len(k_matrix[0])):
            k_matrix[i][j] = get_gauss_kernel(X[i], X[j], h)
    #による更新 ADMM
    for _ in range(10):
        theta_next, z_next, u_next = calc_admm(zk, uk, k_matrix, Y, lam)
        zk = z_next
        uk = u_next生成フェーズ
    #
    def get_pred_fx(x):
        fx_pred = 0
        for i in range(n):
            k=get_gauss_kernel(x, X[i], h)
            fx_pred += theta_next[i]*k
        return fx_pred
    x_pred = np.linspace(-3, 3, 1000)
    fx_pred = np.empty_like(x_pred)
```

```python
    for i in range(len(x_pred)):
        fx_pred[i] = get_pred_fx(x_pred[i])

    return x_pred, fx_pred


def plot_graph(x_sample, y_sample, x_pred, y_pred, y_acc):
    plt.clf()
    plt.scatter(x_sample, y_sample, label="sample" ,c='red', marker='o')
    plt.plot(x_pred, y_pred, label="prediction", c='blue')
    plt.plot(x_pred, y_acc, label="true", c="green")
    plt.legend()
    plt.savefig(r"C:\Users\msy-o\Documents\lecture\summer\data_analytics\report\03\output\lasso.png")
    plt.show()
    return None
if __name__ == "__main__":
    x_sample, y_sample = get_sample()
    x_pred, y_pred = calc_lasso(x_sample, y_sample, 0.3, 0.1)
    y_acc = np.empty_like(x_pred)
    for i in range(len(y_acc)):
        y_acc[i] = func(x_pred[i], with_noise=False)
    plot_graph(x_sample, y_sample, x_pred, y_pred, y_acc)
```