

先端データ解析論レポート第三回

48196635 桑原亮介

2019 年 4 月 26 日

1 宿題 1

目的：下記式の証明

$$\operatorname{argmin}_z T(z) = \max(0, \theta + u - \lambda) + \min(0, \theta + u + \lambda) \quad (1)$$

ただし、

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

$z \geq 0$ および $z < 0$ で場合分けを行う。

$z \geq 0$ のとき、

$$(T(z))' = \lambda - u - \theta + z \quad (3)$$

$$z = \theta + u - \lambda \quad (4)$$

ここで、 $\theta + u - \lambda < 0$ の場合にも、条件より $z \geq 0$ であるため、

$$z = \max(0, \theta + u - \lambda) \quad (5)$$

$z < 0$ のとき、

$$(T(z))' = -\lambda - u - \theta + z \quad (6)$$

$$z = \theta + u + \lambda \quad (7)$$

ここで、 $\theta + u + \lambda \geq 0$ の場合にも、条件より $z < 0$ であるため、

$$z = \min(0, \theta + u + \lambda) \quad (8)$$

式 (5) および式 (8) より、

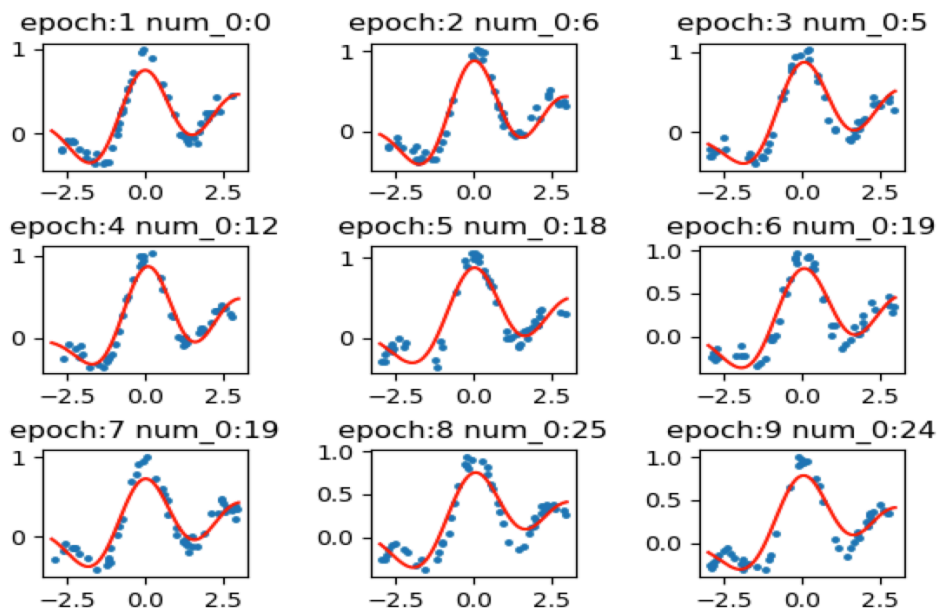
$$\operatorname{argmin}_z T(z) = \max(0, \theta + u - \lambda) + \min(0, \theta + u + \lambda)$$

2 宿題 2

目的: ガウスカーネルモデルに対して反復式を求め、適当なデータとモデルに対してスパース回帰を実装する。

方法: データ数 1000×9 エポックでパラメータ (50 個) の調整を行い、値が 0 になるパラメタ個数の推移をエポック毎に見ていく。

結果: 下記図の通りとなった。曲線の当てはまり具合はさほど変わらないが、値が 0 となるパラメータの数が増えている。



実装コード:

Listing 1 lasso regression

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split

np.random.seed(0)
ans=10**9
ans_h=10**9
ans_l=10**9
xmin=-3
xmax=3
n=500
```

```

lam=0.1
zeros=np.zeros(n//10).reshape(n//10,1)
Lam=np.full(n//10,lam).reshape(n//10,1)
u=np.random.rand(n//10).reshape(n//10,1)
z=np.random.rand(n//10).reshape(n//10,1)
h=1.0
flag=False

#データを生成する
def generate_sample(sample_size):
    x=np.linspace(start=xmin,stop=xmax,num=sample_size)
    pix=np.pi*x
    target=np.sin(pix)/pix+0.1*x
    noise=0.05*np.random.normal(loc= 0.,scale=1.,size=sample_size)
    return x,target+noise

#ガウスカネル
def calc_design_matrix(x_train,x_test,y_train,h,lam):
    global theta
    global z
    global u
    global flag
    #を算出theta
    k=np.exp(-(x_train[None]-x_train[:,None])**2/(2*h**2))
    theta=np.linalg.solve(k.T.dot(k)+np.identity(len(k)),k.T.dot(y_train[:,None])+(z-u))
    z=theta if not flag else np.maximum(zeros,theta+u-Lam)-np.maximum(zeros,-theta-u-Lam)
    flag=True
    u=u+theta-z
    # print(z)
    #データからカーネル関数設計
    K = np.exp(-(x_train[None]-x_test[:,None])**2/(2*h**2))
    return K.dot(theta),z #予測値を返す

# 二乗誤差
def sqared_error(y,t):
    y=y.flatten()
    t=t.flatten()

```

```

    return np.sum(((y-t)**2))

fig=plt.figure()
fig.subplots_adjust(hspace=0.6, wspace=0.6)
i=0
for epoch in range(1,10):
    x,y=generate_sample(sample_size=n)
    x,y= shuffle(x,y)
    x_train , x_test , y_train , y_test=train_test_split(x,y, test_size=0.9)
    idx=np.argsort(x_test)
    x_test=x_test[idx]
    y_test=y_test[idx]
    prediction , z=calc_design_matrix(x_train , x_test , y_train , h, lam)
    if epoch%1==0:
        i+=1
        error=sqared_error(y_test , prediction)
        ax=fig.add_subplot(3,3,i)
        ax.set_title("epoch:{ } \num_0:{ }".format(str(i),str(np.sum(z==0.0))))
        ax.scatter(x_train , y_train , s=6)
        ax.plot(x_test , prediction , c='red ')

plt.show()

```