先端データ解析論レポート 第 8 回

荻野 聖也（37-196323）

2019 年 6 月 17 日

宿題 1

二乗ヒンジ損失に対する適応正則化回帰

$$
\begin{aligned}
J(\boldsymbol{\mu},\boldsymbol{\Sigma}) = &\left(\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(\boldsymbol{x})y\right)\right)^2 + \boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\Sigma}\boldsymbol{\phi}(\boldsymbol{x}) \\
&+\gamma\left\{\log\frac{\det(\widetilde{\boldsymbol{\Sigma}})}{\det(\boldsymbol{\Sigma})} + \mathrm{Tr}(\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma}) + (\boldsymbol{\mu}-\widetilde{\boldsymbol{\mu}})^T\widetilde{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{\mu}-\widetilde{\boldsymbol{\mu}}) - d\right\}
\end{aligned}
\tag{1}
$$

の最小点を与える $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}$ を求める。
$\boldsymbol{\mu}$ に対する停留点を求める。

$$
\frac{\partial}{\partial\boldsymbol{\mu}}\left(\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y\right)\right)^2 = 2\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y\right)(-\boldsymbol{\phi}(x)y)
$$

であることに注意すると，

$$
\begin{aligned}
\frac{\partial J}{\partial\boldsymbol{\mu}} &= 2\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y\right)(-\boldsymbol{\phi}(x)y) + 2\gamma\left\{\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\mu} + \left(\boldsymbol{\mu}^T\widetilde{\boldsymbol{\Sigma}}^{-1}\right) - \widetilde{\boldsymbol{\Sigma}}^{-1}\widetilde{\boldsymbol{\mu}} - \left(\widetilde{\boldsymbol{\mu}}^{-1}\widetilde{\boldsymbol{\Sigma}}^{-1}\right)^T\right\} \\
&= 2\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y\right)(-\boldsymbol{\phi}(x)y) + 2\gamma\left(\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\mu} - \widetilde{\boldsymbol{\Sigma}}^{-1}\widetilde{\boldsymbol{\mu}}\right)
\end{aligned}
$$

となる。ここで，0 と $1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y$ の大小について考える。
(i) $\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y\right) = 0$ の時，$\frac{\partial J}{\partial\boldsymbol{\mu}} = 0$ に対して，

$$
\boldsymbol{\mu} = \widetilde{\boldsymbol{\mu}}
$$

を得る。
(ii) $\max\left(0, 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y\right) = 1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y$ の時，$\frac{\partial J}{\partial\boldsymbol{\mu}} = 0$ に対して，

$$
\begin{aligned}
&(1-\boldsymbol{\mu}^T\boldsymbol{\phi}(x)y)(-\boldsymbol{\phi}(x)y) + \gamma\widetilde{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{\mu}-\widetilde{\boldsymbol{\mu}}) = 0 \\
\Longleftrightarrow &\left(\gamma\widetilde{\boldsymbol{\Sigma}}^{-1} + y^2\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}^T(\boldsymbol{x})\right)\boldsymbol{\mu} = \gamma\widetilde{\boldsymbol{\Sigma}}^{-1}\widetilde{\boldsymbol{\mu}} + y\boldsymbol{\phi}(\boldsymbol{x}) \\
\Longleftrightarrow &\left(\widetilde{\boldsymbol{\Sigma}}^{-1} + \frac{y^2}{\gamma}\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}^T(\boldsymbol{x})\right)\boldsymbol{\mu} = \widetilde{\boldsymbol{\Sigma}}^{-1}\widetilde{\boldsymbol{\mu}} + \frac{y}{\gamma}\boldsymbol{\phi}(\boldsymbol{x})
\end{aligned}
$$

ここで，公式より

$$
\left(\widetilde{\boldsymbol{\Sigma}}^{-1} + \frac{y^2}{\gamma}\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}^T(\boldsymbol{x})\right)^{-1} = \widetilde{\boldsymbol{\Sigma}} - \frac{\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}}{\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x}) + \frac{\gamma}{y^2}}
$$

なので，

$$
\begin{aligned}
\boldsymbol{\mu} &= \left(\widetilde{\boldsymbol{\Sigma}} - \frac{\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x})\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}}{\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x}) + \frac{\gamma}{y^2}}\right)\left(\widetilde{\boldsymbol{\Sigma}}^{-1}\widetilde{\boldsymbol{\mu}} + \frac{y}{\gamma}\boldsymbol{\phi}(\boldsymbol{x})\right) \\
&= \widetilde{\boldsymbol{\mu}} + \frac{y(1-\widetilde{\boldsymbol{\mu}}\boldsymbol{\phi}(\boldsymbol{x})y)}{y^2\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x}) + \gamma}\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x})
\end{aligned}
$$

ここで，$y \in \{1, -1\}$ なので，$y^2 = 1$ であることから，

$$\boldsymbol{\mu} = \widetilde{\boldsymbol{\mu}} + \frac{y(1 - \widetilde{\boldsymbol{\mu}}\boldsymbol{\phi}(\boldsymbol{x})y)}{\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x}) + \gamma}\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x})$$

となる。(i) と (ii) をまとめると，

$$\boldsymbol{\mu} = \widetilde{\boldsymbol{\mu}} + \frac{y \max(0, 1 - \widetilde{\boldsymbol{\mu}}\boldsymbol{\phi}(\boldsymbol{x})y)}{\boldsymbol{\phi}^T(\boldsymbol{x})\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x}) + \gamma}\widetilde{\boldsymbol{\Sigma}}\boldsymbol{\phi}(\boldsymbol{x})$$

となり，題意を示せた。

宿題 2

二乗ヒンジ損失に基づく適応正則化分類を用いて，線形モデル

$$\begin{aligned} f_{\boldsymbol{\theta}}(\boldsymbol{x}) &= f_{\boldsymbol{\theta}}(x^{(1)}, x^{(2)}) \\ &= (x^{(1)}, x^{(2)}, 1)\boldsymbol{\theta} \end{aligned}$$

に対して分類を行う。

サンプルについて

サンプルは講義資料を参考にして，2 クラスのサンプルを用意した。この様子を図 1 に示す。外れ値があることに留意する。

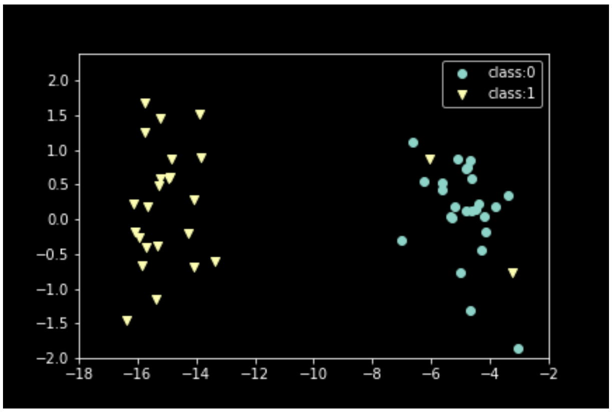

図 1　Caption

適応正則化分類について

パラメータの学習基準は訓練の適合性・パラメータの変化量の抑制・適応正則化から構成され，これを $J(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ とすると

$$J(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathrm{loss}(f_{\boldsymbol{\mu}}(\boldsymbol{x}), y) + 2\gamma \mathrm{KL}\left[N(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \| N(\widetilde{\boldsymbol{\mu}}, \widetilde{\boldsymbol{\Sigma}})\right] + \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{\Sigma} \boldsymbol{\phi}(\boldsymbol{x})$$

で与えられる。これらの $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ に関する停留点を求めればよい。導出は割愛するが，ヒンジ損失に対する解 $\hat{\boldsymbol{\mu}}$，$\hat{\boldsymbol{\Sigma}}$ は

$$\begin{cases} \hat{\boldsymbol{\mu}} = \widetilde{\boldsymbol{\mu}} + \frac{y \max(0, 1 - \widetilde{\boldsymbol{\mu}}^T \boldsymbol{\phi}(\boldsymbol{x})y)}{\boldsymbol{\phi}(\boldsymbol{x})^T \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\phi}(\boldsymbol{x}) + \gamma} \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\phi}(\boldsymbol{x}) \\ \hat{\boldsymbol{\Sigma}} = \widetilde{\boldsymbol{\Sigma}} - \frac{\widetilde{\boldsymbol{\Sigma}} \boldsymbol{\phi}(\boldsymbol{x}) \boldsymbol{\phi}(\boldsymbol{x})^T \widetilde{\boldsymbol{\Sigma}}}{\boldsymbol{\phi}(\boldsymbol{x})^T \widetilde{\boldsymbol{\Sigma}} \boldsymbol{\phi}(\boldsymbol{x}) + \gamma} \end{cases}$$

で与えられる。今回，バッチ処理をせずにオンライン学習を行う。

結果について

結果を以下に示す。ただし，計算に用いたパラメータ等については表 1 に示す。外れ値に対しての影響がほとんど出ていないことがうかがえる。コードについては補遺および添付ファイルに示す。

表 1　パラメータについて

γ	10
y	$\{-1, 1\}$
iteration	1000 回

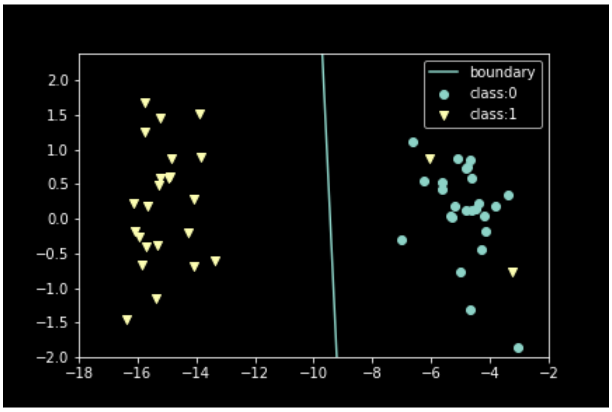

図 2　適応正則化分類による分類結果

補遺

Listing 1　宿題 2 のソースコード

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

np.random.seed(1) # set the random seed for reproducibility

# 出力のディレクトリの作成
if not os.path.exists("output"):
    os.mkdir("output")

def data_generate(n=50):
    """ 訓練データ生成
    Arg:
        n(int) 生成データ数default=50

    Return:
        x(ndarray(float)) 生成データの入力
            shape=(n,3)
```

```
            x_i = [x1, x2, で構成される 1]

        y(ndarray(int)) 生成データのラベル
            shape=(n,)
            y_i = {-1, 1}
    """
    x = np.random.randn(n, 3)
    x[:n // 2, 0] -= 15
    x[n // 2:, 0] -= 5
    x[1:3, 0] += 10
    x[:, 2] = 1
    y = np.concatenate((np.ones(n // 2), -np.ones(n // 2)))
    index = np.random.permutation(np.arange(n))
    return x[index], y[index]

def AdaptedRegulateClassifer(x, y):
    """ 適応正則化分類
    Arg:
        x(ndarray(float)) 生成データの入力
            shape=(n,3)
            x_i = [x1, x2, で構成される 1]

        y(ndarray(int)) 生成データのラベル
            shape=(n,)
            y_i = {-1, 1}

    Return:
        mu(ndarray(float)) 適応正則化による mu

    Memo:
        phi = (x1, x2, 1).Tとを求める
        muSigma反復法

    """
    iteration=1000
    gamma = 10
    indexes = range(50)
    mu = np.array([1,1,1]).reshape(3,1)
    print("mu shape : {}".format(mu.shape))
    print(mu)
    Sigma = np.eye(3)
    print("Sigma shape : {}".format(Sigma.shape))
    print(Sigma)
    for index in range(iteration):
        index = np.random.choice(indexes)
        index = index%50
        y_i = y[index]
        phi = x[index].reshape(3,1)
        beta = np.dot(phi.T, Sigma.dot(phi))+gamma
        phiphiT = np.dot(phi, phi.T)
```

```
        Sigma_next = Sigma - np.dot(Sigma.dot(phiphiT),Sigma)/float(beta)
        mu_next = mu + y_i*max(0, 1-y_i*mu.T.dot(phi))/beta*Sigma.dot(phi)
        mu = mu_next
        Sigma = Sigma_next
    return mu

x, y = data_generate()
class0_index = np.where(y==-1)
class1_index = np.where(y==1)
class0_x = x[class0_index]
class1_x = x[class1_index]
class0_x = np.delete(class0_x, 2, axis=1)
class1_x = np.delete(class1_x, 2, axis=1)
# サンプルプロット
plt.clf()
plt.scatter(class0_x[:,0], class0_x[:,1], marker="o", label="class:0")
plt.scatter(class1_x[:,0], class1_x[:,1], marker="v", label="class:1")
plt.ylim(-2,2.4)
plt.xlim(-18,-2)
plt.legend()
plt.savefig("./output/sample_plot.png")
plt.show()

plt.clf()
plt.scatter(class0_x[:,0], class0_x[:,1], marker="o", label="class:0")
plt.scatter(class1_x[:,0], class1_x[:,1], marker="v", label="class:1")
mu = AdaptedRegulateClassifer(x, y)
x0 = np.linspace(-12,-8, num=50)
x1 = -mu[0]/mu[1]*x0-mu[2]/mu[1]
plt.plot(x0,x1, label="boundary")
plt.ylim(-2,2.4)
plt.xlim(-18,-2)
plt.legend()
plt.savefig("./output/result.png")
plt.show()
```