先端データ解析論
第十一回　レポート

37-196360　森田涼介

2019 年 7 月 8 日

宿題 1

フィッシャー判別分析を実装する。

クラス内散布行列は，

$$\boldsymbol{S}^{(\mathrm{w})} = \sum_{y=1}^{c} \sum_{i:y_i=y} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \tag{1}$$

クラス間散布行列は，

$$\boldsymbol{S}^{(\mathrm{b})} = \sum_{y=1}^{c} n_y \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathrm{T}} \tag{2}$$

である。クラス内散布を小さく，クラス間散布を大きくするような写像を求める。つまり，次を求めればよい。

$$\boldsymbol{T}_{\mathrm{FDA}} = \arg\max_{\boldsymbol{T}} \left((\boldsymbol{T}\boldsymbol{S}^{(\mathrm{w})}\boldsymbol{T}^{\mathrm{T}})^{-1} \boldsymbol{T}\boldsymbol{S}^{(\mathrm{b})}\boldsymbol{T}^{\mathrm{T}} \right) \tag{3}$$

これは，次の一般化固有値問題を解くことにより，以下のように求まる。

$$\boldsymbol{S}^{(\mathrm{b})}\boldsymbol{\xi} = \lambda \boldsymbol{S}^{(\mathrm{w})}\boldsymbol{\xi} \tag{4}$$

$$\boldsymbol{\xi}_j^{\mathrm{T}} \boldsymbol{S}^{(\mathrm{w})} \boldsymbol{\xi}_j = 1 \tag{5}$$

$$\boldsymbol{T}_{\mathrm{FDA}} = (\boldsymbol{\xi}_1, \cdots, \boldsymbol{\xi}_m)^{\mathrm{T}} \tag{6}$$

7 ページの Listing 1 にプログラムを示した。結果は図 1，2 に示した通りである。

これらより，フィッシャー判別分析によって異なるクラスのデータをうまく分離できるような写像が求まるが，クラス内にクラスタ構造があるとうまくいかないことがわかる。

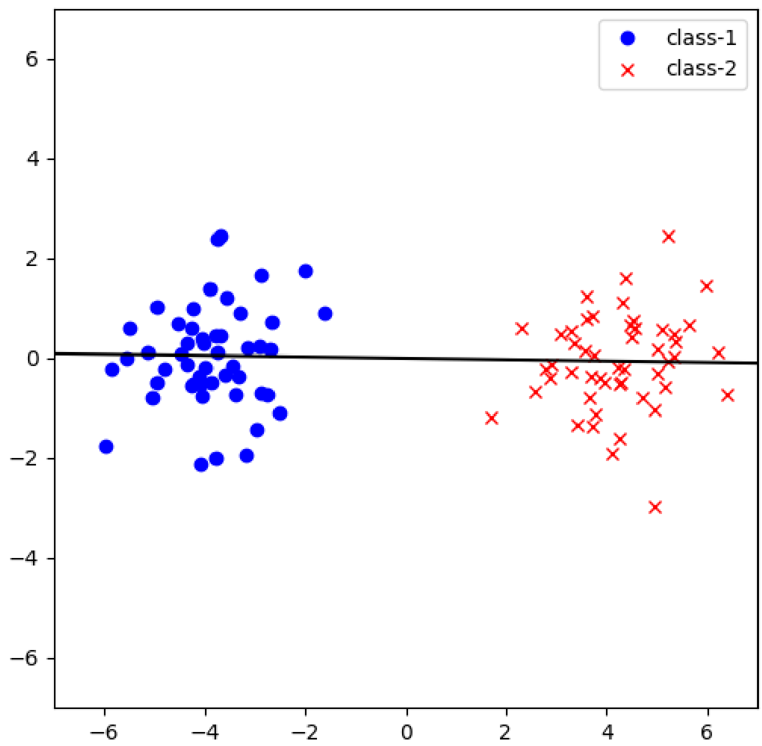

図 1: 2 クラスタのデータに対する結果

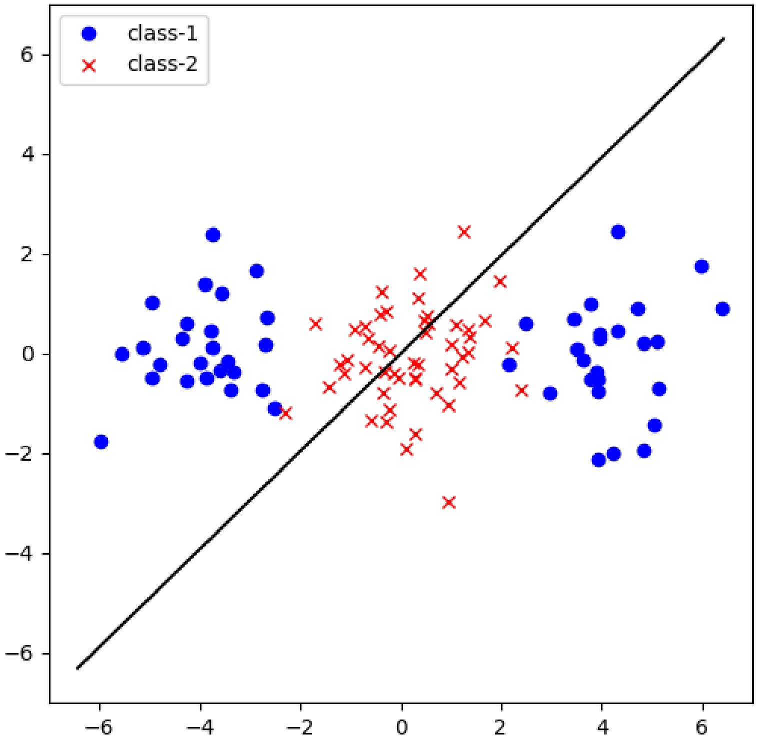

図 2: 3 クラスタのデータに対する結果

宿題 2

標本 $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ を考える。$\boldsymbol{x}$ は中心化されているとする。つまり，

$$\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i = \boldsymbol{0} \tag{7}$$

とする。このとき，散布行列は，

$$\boldsymbol{C} = \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} \tag{8}$$

クラス内散布行列は，

$$\boldsymbol{S}^{(\mathrm{w})} = \sum_{y=1}^c \sum_{i:y_i=y} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \tag{9}$$

クラス間散布行列は，

$$\boldsymbol{S}^{(\mathrm{b})} = \sum_{y=1}^c n_y \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathrm{T}} \tag{10}$$

である。ここで，n_y はクラス y の標本数であり，

$$\boldsymbol{\mu}_y = \frac{1}{n_y}\sum_{i:y_i=y} \boldsymbol{x}_i \tag{11}$$

である。

いま，散布行列について，

$$\boldsymbol{C} = \sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} = \frac{1}{2}\left(\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} + \sum_{i'=1}^n \boldsymbol{x}_{i'} \boldsymbol{x}_{i'}^{\mathrm{T}}\right)$$

と変形できる。

$$\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} = \sum_{i'=1}^n \frac{1}{n}\left(\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}}\right) = \frac{1}{n}\sum_{i,i'=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}}$$

であることから，

$$\boldsymbol{C} = \frac{1}{2}\cdot\frac{1}{n}\sum_{i,i'}^n \left(\boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} + \boldsymbol{x}_{i'} \boldsymbol{x}_{i'}^{\mathrm{T}}\right)$$

となる。ここで，式 (7) から，

$$\begin{aligned}
\sum_{i,i'=1}^n \boldsymbol{x}_i \boldsymbol{x}_{i'}^{\mathrm{T}} &= \sum_{i=1}^n \boldsymbol{x}_i \left(\sum_{i'=1}^n \boldsymbol{x}_{i'}\right)^{\mathrm{T}} + \sum_{i'=1}^n \left(\sum_{i=1}^n \boldsymbol{x}_i\right) \boldsymbol{x}_{i'}^{\mathrm{T}} \\
&= 2\sum_{i=1}^n \boldsymbol{x}_i \sum_{i'=1}^n \boldsymbol{x}_{i'}^{\mathrm{T}} \\
&= 2n\boldsymbol{\mu}\cdot n\boldsymbol{\mu}^{\mathrm{T}} \\
&= \boldsymbol{0}
\end{aligned}$$

が成立することを用いると，結局，

$$
\begin{aligned}
\boldsymbol{C} &= \frac{1}{2} \cdot \frac{1}{n} \sum_{i,i'=1}^{n} \left(\boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} - 2\boldsymbol{x}_i \boldsymbol{x}_{i'}^{\mathrm{T}} + \boldsymbol{x}_{i'} \boldsymbol{x}_{i'}^{\mathrm{T}} \right) \\
&= \frac{1}{2} \sum_{i,i'=1}^{n} \frac{1}{n} (\boldsymbol{x}_i - \boldsymbol{x}_{i'})(\boldsymbol{x}_i - \boldsymbol{x}_{i'})^{\mathrm{T}} \qquad (12)
\end{aligned}
$$

得る。

同様に，クラス内散布行列は，

$$
\begin{aligned}
\boldsymbol{S}^{(\mathrm{w})} &= \sum_{y=1}^{c} \sum_{i:y_i=y} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \\
&= \frac{1}{2} \sum_{y=1}^{c} \frac{1}{n_y} \sum_{i,i':y_{i,i'}=y} \left\{ \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)^{\mathrm{T}} + \left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \right\}
\end{aligned}
$$

と変形でき，ここで，式 (11) から，

$$
\begin{aligned}
\sum_{i,i':y_{i,i'}=y} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)^{\mathrm{T}} &= 2 \sum_{i:y_i=y} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right) \cdot \sum_{i':y'_i=y} \left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \\
&= 2\left(n_y \boldsymbol{\mu}_y - n_y \boldsymbol{\mu}_y\right)\left(n_y \boldsymbol{\mu}_y - n_y \boldsymbol{\mu}_y\right)^{\mathrm{T}} \\
&= \boldsymbol{0}
\end{aligned}
$$

となることから，結局，

$$
\begin{aligned}
\boldsymbol{S}^{(\mathrm{w})} &= \frac{1}{2} \sum_{y=1}^{c} \frac{1}{n_y} \sum_{i,i':y_{i,i'}=y} \left\{ \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)^{\mathrm{T}} - 2\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)^{\mathrm{T}} + \left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \right\} \\
&= \frac{1}{2} \sum_{y=1}^{c} \frac{1}{n_y} \sum_{i,i':y_{i,i'}=y} \left\{ \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right) - \left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right) \right\} \left\{ \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right) - \left(\boldsymbol{x}_{i'} - \boldsymbol{\mu}_y\right) \right\}^{\mathrm{T}} \\
&= \frac{1}{2} \sum_{y=1}^{c} \frac{1}{n_y} \sum_{i,i':y_{i,i'}=y} (\boldsymbol{x}_i - \boldsymbol{x}_{i'})(\boldsymbol{x}_i - \boldsymbol{x}_{i'})^{\mathrm{T}}
\end{aligned}
$$

となる。これは，

$$
Q_{i,i'}^{(\mathrm{w})} = \begin{cases} 1/n_y & (y_i = y_{i'} = y) \\ 0 & (y_i \neq y_{i'}) \end{cases} \qquad (13)
$$

なる $Q_{i,i'}^{(\mathrm{w})}$ を用いて，

$$
\boldsymbol{S}^{(\mathrm{w})} = \frac{1}{2} \sum_{i,i'=1}^{n} Q_{i,i'}^{(\mathrm{w})} (\boldsymbol{x}_i - \boldsymbol{x}_{i'})(\boldsymbol{x}_i - \boldsymbol{x}_{i'})^{\mathrm{T}} \qquad (14)
$$

と表せる。

いま，$\boldsymbol{S}^{(\mathrm{w})}$ について変形すると，

$$
\begin{aligned}
\boldsymbol{S}^{(\mathrm{w})} &= \sum_{y=1}^{c} \sum_{i:y_i=y} \left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)\left(\boldsymbol{x}_i - \boldsymbol{\mu}_y\right)^{\mathrm{T}} \\
&= \sum_{y=1}^{c} \sum_{i:y_i=y} \left(\boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} - 2\boldsymbol{\mu}_y \boldsymbol{x}_i^{\mathrm{T}} + \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathrm{T}}\right) \\
&= \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} - \sum_{y=1}^{c} 2\boldsymbol{\mu}_y \sum_{i:y_i=y} \boldsymbol{x}_i^{\mathrm{T}} + \sum_{y=1}^{c} n_y \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathrm{T}} \\
&= \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} - \sum_{y=1}^{c} 2\boldsymbol{\mu}_y n_y \boldsymbol{\mu}_y^{\mathrm{T}} + \sum_{y=1}^{c} n_y \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathrm{T}} \\
&= \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} - \sum_{y=1}^{c} n_y \boldsymbol{\mu}_y \boldsymbol{\mu}_y^{\mathrm{T}}
\end{aligned}
$$

となり，式 (8)，(10) とから，

$$
\boldsymbol{C} = \boldsymbol{S}^{(\mathrm{w})} + \boldsymbol{S}^{(\mathrm{b})} \tag{15}
$$

が成立する。

式 (12)，(14)，(15) から，

$$
\begin{aligned}
\boldsymbol{S}^{(\mathrm{b})} &= \boldsymbol{C} - \boldsymbol{S}^{(\mathrm{w})} \\
&= \frac{1}{2} \sum_{i,i'=1}^{n} \left(\frac{1}{n} - Q_{i,i'}^{(\mathrm{w})}\right) (\boldsymbol{x}_i - \boldsymbol{x}_{i'})(\boldsymbol{x}_i - \boldsymbol{x}_{i'})^{\mathrm{T}} \\
&= \frac{1}{2} \sum_{i,i'=1}^{n} Q_{i,i'}^{(\mathrm{b})} (\boldsymbol{x}_i - \boldsymbol{x}_{i'})(\boldsymbol{x}_i - \boldsymbol{x}_{i'})^{\mathrm{T}} \qquad (16) \\
Q_{i,i'}^{(\mathrm{b})} &= \frac{1}{n} - Q_{i,i'}^{(\mathrm{w})} \\
&= \begin{cases} 1/n - 1/n_y & (y_i = y_{i'} = y) \\ 1/n & (y_i \neq y_{i'}) \end{cases} \qquad (17)
\end{aligned}
$$

となる。

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
import scipy.linalg


def generate_data(sample_size=100, pattern='two_cluster'):
    if pattern not in ['two_cluster', 'three_cluster']:
        raise ValueError('Dataset pattern must be one of '
                         '[two_cluster, three_cluster].')
    x = np.random.normal(size=(sample_size, 2))
    if pattern == 'two_cluster':
        x[:sample_size // 2, 0] -= 4
        x[sample_size // 2:, 0] += 4
    else:
        x[:sample_size // 4, 0] -= 4
        x[sample_size // 4:sample_size // 2, 0] += 4
    y = np.ones(sample_size, dtype=np.int64)
    y[sample_size // 2:] = 2
    return x, y


def scatter_matrices(x, y):
    n = x.shape[0]
    d = x.shape[1]

    labels = np.unique(y)
    C, Sw, Sb = np.zeros((d, d)), np.zeros((d, d)), np.zeros((d, d))
    for label in labels:
        x_y = x[(y == label), :]
```

```
        n_y = x_y.shape[0]
        mu_y = x_y.mean(axis=0)[:, np.newaxis]
        Sb += n_y * mu_y.dot(mu_y.T)

        for i in range(n_y):
            x_i = x_y[i][:, np.newaxis]
            diff = (x_i - mu_y)
            Sw += diff.dot(diff.T)

            C += x_i.dot(x_i.T)

    return C, Sw, Sb


def train(x, y, n_components):
    """Fisher Discriminant Analysis.
    Implement this function

    Returns
    -------
    T : (1, 2) ndarray
        The embedding matrix.
    """

    C, Sw, Sb = scatter_matrices(x, y)
    eigen_values, eigen_vectors = scipy.linalg.eig(Sb, Sw)

    # normalize
    for i in range(len(eigen_vectors)):
        eigen_vectors[i] = eigen_vectors[i]/np.linalg.norm(eigen_vectors[i])

    T = eigen_vectors[:n_components]
    return T


def visualize(x, y, T, path=None):
    plt.figure(1, (6, 6))
    plt.xlim(-7., 7.)
    plt.ylim(-7., 7.)
    plt.plot(x[y == 1, 0], x[y == 1, 1], 'bo', label='class-1')
    plt.plot(x[y == 2, 0], x[y == 2, 1], 'rx', label='class-2')
    plt.plot(
        np.array([-T[:, 0], T[:, 0]]) * 9,
        np.array([-T[:, 1], T[:, 1]]) * 9,
        'k-',
        )
    plt.legend()
```

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


def main():
    # settings
    n = 100
    n_components = 1
    #mode = 'two_cluster'
    mode = 'three_cluster'
    fig_path = f'../figures/assignment1_result_{mode}.png'
    np.random.seed(10)


    # generate data
    x, y = generate_data(sample_size=n, pattern=mode)
    #print(x.shape, y.shape)


    # train
    T = train(x, y, n_components)


    # result
    print(f'data: {mode}  (#sample = {n})')
    print(f'T = {T}')

    visualize(x, y, T, path=fig_path)



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