

パターン認識

2019-04-16 授業分 レポート

37-196360 森田涼介

2019 年 4 月 26 日

宿題

点群 (2D 2000 点) と正規分布の十分統計量を含むデータファイル `data.mat` 及び `data2.mat` のデータに対して、それぞれ同時対角化を適用する。

結果

まず、`data.mat` のデータについて示す。同時対角化前の、`data.mat` に含まれる正規分布の十分統計量は、

$$\mu_1 = \begin{bmatrix} 0.53823082 \\ -3.04269717 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 3.34176503 & 1.33496408 \\ 1.33496408 & 1.73462199 \end{bmatrix} \quad (1)$$

$$\mu_2 = \begin{bmatrix} -0.99565419 \\ 0.48512018 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 3.2089801 & -0.87806376 \\ -0.87806376 & 3.07417451 \end{bmatrix} \quad (2)$$

であった。これらを同時対角化すると、その変換後の正規分布の統計量は次のようになる。なお、ここでは簡単のため、上と同じ文字を用いた。

$$\mu_1 = \begin{bmatrix} 0.72903906 \\ 2.89642411 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \quad (3)$$

$$\mu_2 = \begin{bmatrix} 0.24511607 \\ -0.92796166 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.57246206 & 0 \\ 0 & 3.95701516 \end{bmatrix} \quad (4)$$

また、同時対角化前後の散布図を図 1 に示す。

次に、`data2.mat` のデータについて示す。同時対角化前の、`data2.mat` に含まれる正規分布の十分統計量は、

$$\mu_1 = \begin{bmatrix} 0.46886439 \\ -2.96015249 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 3.3154755 & 1.23521877 \\ 1.23521877 & 1.69501881 \end{bmatrix} \quad (5)$$

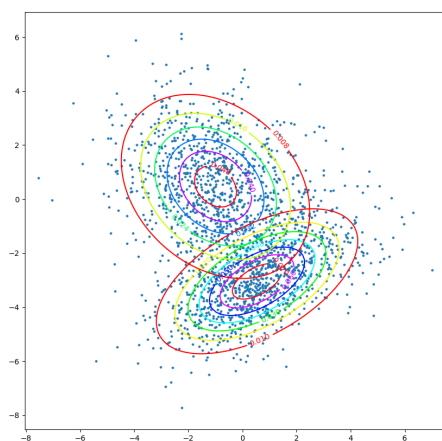
$$\mu_2 = \begin{bmatrix} -0.99565419 \\ 0.48512018 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.44210079 & -0.18739151 \\ -0.18739151 & 0.4396869 \end{bmatrix} \quad (6)$$

であった。これらを同時対角化すると、その変換後の正規分布の統計量は次のようになる。

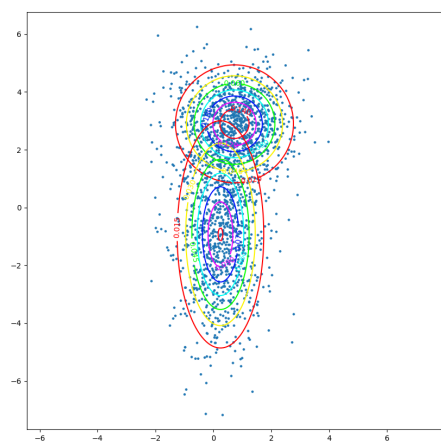
$$\mu_1 = \begin{bmatrix} 0.77239293 \\ 2.72545062 \end{bmatrix} \quad \Sigma_1 = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix} \quad (7)$$

$$\mu_2 = \begin{bmatrix} 0.23380354 \\ -0.9153229 \end{bmatrix} \quad \Sigma_2 = \begin{bmatrix} 0.06641293 & 0 \\ 0 & 0.58577592 \end{bmatrix} \quad (8)$$

また、同時対角化前後の散布図を図 1 に示す。

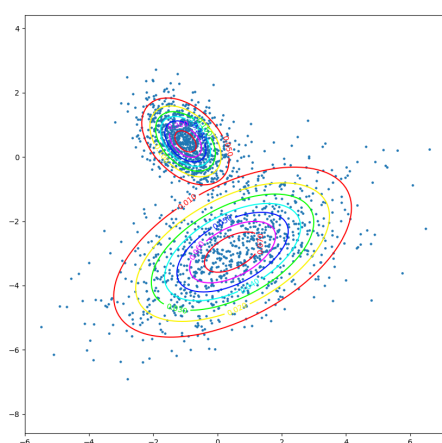


(a) 同時対角化前の散布図

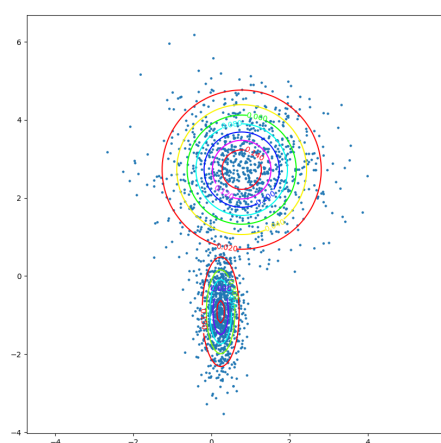


(b) 同時対角化後の散布図

図 1: data.mat に含まれる点群についての同時対角化前後の散布図



(a) 同時対角化前の散布図



(b) 同時対角化後の散布図

図 2: data.mat に含まれる点群についての同時対角化前後の散布図

プログラム

プログラムは 3 ページの listing 1 に示した。プログラムに含まれる関数とその簡単な説明を以下に示す。

- `load`
`scipy.io.loadmat` を用いて `.mat` ファイルから点群データとその正規分布の十分統計量を `load` する関数
- `print_data`
点群データの型, 及びその正規分布の十分統計量を `console` に出力する関数
- `sampling_norm_dist`
`plot` 時に等高線を出すために, 正規分布から `sampling` をするための関数
- `plot_data`
点群の散布図と, その正規分布の統計量から描ける等高線を描く関数
- `simultaneous_diagonalize`
点群データとその正規分布の十分統計量に対して同時対角化を行う関数
- `main` 実行用の関数。`data_path` という変数で `data.mat` を読むか `data2.mat` を読むかを切り替える。

なお, 実行環境と用いた言語・ライブラリは以下の表 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: simultaneous_diagonalization.py

```
1  # -*- coding: utf-8 -*-
2
3  import numpy as np
4  import matplotlib.pyplot as plt
5  from scipy.io import loadmat
6
7
8  def load(path):
9      data = loadmat(path)
10     x = data['x']
11     cov1 = data['cov1']
12     cov2 = data['cov2']
13     m1 = data['m1']
14     m2 = data['m2']
15     return x, m1, m2, cov1, cov2
16
17
```

```

18 def print_data(x, m1, m2, cov1, cov2):
19     print('data')
20     print('x shape: ', x.shape)
21     print()
22     print('data 1')
23     print('m1 {}: \n'.format(m1.shape), m1)
24     print('cov1 {}: \n'.format(cov1.shape), cov1)
25     print()
26     print('data 2')
27     print('m2 {}: \n'.format(m2.shape), m2)
28     print('cov2 {}: \n'.format(cov2.shape), cov2)
29     print()
30
31
32 def sampling_norm_dist(x, y, mean, cov):
33     icov = np.linalg.inv(cov)
34     xt = x - mean[0, 0]
35     yt = y - mean[0, 1]
36     p = (
37         1./(2. * np.pi * np.sqrt(np.linalg.det(cov)))
38         * np.exp(-(1./2.) * (icov[0,0]*xt*xt + (icov[0,1] + icov[1,0])*xt*yt
39         + icov[1,1]*yt*yt))
40     )
41     return p
42
43 def plot_data(
44     x, m1, m2, cov1, cov2,
45     x_linespace=None, y_linespace=None,
46     show=True, save=False, path=None,
47 ):
48     if x_linespace is None:
49         x_linespace = np.linspace(-10, 10, 100)
50     if y_linespace is None:
51         y_linespace = np.linspace(-10, 10, 100)
52
53     x1, x2 = np.meshgrid(x_linespace, y_linespace)
54     p1 = sampling_norm_dist(x1, x2, m1, cov1)
55     p2 = sampling_norm_dist(x1, x2, m2, cov2)
56
57     plt.figure(figsize=(10, 10))
58     plt.axis('equal')
59     plt.scatter(x[:, 0], x[:, 1], s=5)
60     cs1 = plt.contour(x1, x2, p1, cmap='hsv')
61     plt.clabel(cs1)
62     cs2 = plt.contour(x1, x2, p2, cmap='hsv')
63     plt.clabel(cs2)

```

```

64
65     if save:
66         plt.savefig(path)
67     if show:
68         plt.show()
69     return
70
71
72 def simultaneous_diagonalize(x, m1, m2, cov1, cov2):
73     theta, phi = np.linalg.eig(cov1)
74     # theta = np.diag(theta)
75     theta_inv_half = np.diag(theta**(-1/2))
76     k = (theta_inv_half).dot(phi.T).dot(cov2).dot(phi).dot(theta_inv_half)
77     lamb, psi = np.linalg.eig(k)
78     lamb = np.diag(lamb)
79
80     # convert
81     def convert(x):
82         y = theta_inv_half.dot(phi.T).dot(x.T).T
83         z = psi.T.dot(y.T).T
84         return z
85
86     z = convert(x)
87     m1_z = convert(m1)
88     m2_z = convert(m2)
89     cov1_z = np.eye(x.shape[1])
90     cov2_z = lamb
91     return z, m1_z, m2_z, cov1_z, cov2_z
92
93
94 def main():
95     # settings
96     data_path = '../data/data2.mat'
97     fig_path_before = '../figures/scatter_before_2.png'
98     fig_path_after = '../figures/scatter_after_2.png'
99     offset = 0.5
100
101     print('data file: {}'.format(data_path))
102
103     # load and look data
104     x, m1, m2, cov1, cov2 = load(data_path)
105     print_data(x, m1, m2, cov1, cov2)
106     x_linespace = np.linspace(
107         x[:, 0].min()-offset, x[:, 0].max()+offset, 100
108     )
109     y_linespace = np.linspace(
110         x[:, 1].min()-offset, x[:, 1].max()+offset, 100

```

```

111     )
112     plot_data(
113         x, m1, m2, cov1, cov2,
114         x_linespace, y_linespace,
115         save=True, path=fig_path_before,
116     )
117
118     # simultaneous diagonalization and convert
119     z, m1_z, m2_z, cov1_z, cov2_z = simultaneous_diagonalize(x, m1, m2, cov1,
120     cov2)
121     print_data(z, m1_z, m2_z, cov1_z, cov2_z)
122     x_linespace = np.linspace(
123         z[:, 0].min()-offset, z[:, 0].max()+offset, 100
124     )
125     y_linespace = np.linspace(
126         z[:, 1].min()-offset, z[:, 1].max()+offset, 100
127     )
128     plot_data(
129         z, m1_z, m2_z, cov1_z, cov2_z,
130         x_linespace, y_linespace,
131         save=True, path=fig_path_after,
132     )
133
134 if __name__ == '__main__':
135     main()

```