

平成 30 年度 情報理工学実験 II :  
「信号処理」に関する実験

Klak Bartosz  
クラク バルトツシュ  
学生番号 02166102

2019 年 1 月 22 日

## 1 問題 1

Listing 1 課題 1

```
1 >> A = randn(5,5);
2 >> [L,U] = LU_simple(A);
3 >> A - L*U
4
5 ans =
6
7 1.0e-13 *
8
9      0      0      0      0      0
10     0 -0.0006      0 -0.0003  0.0001
11     0  0.0006      0      0 -0.0022
12     0      0      0 -0.0044  0.0067
13     0      0 -0.0022 -0.0144  0.2864
```

回答としては 5x5 行列である。その行列はほぼ零行列である。0 以外の要素は 0 に近いである。それで  $A \approx LU$  ということが確認できる。

## 2 問題 2

Listing 2 課題 2

```
1 >> A = randn(5,5);
2 >> [L,U,P,sg] = LU(A);
3 >> P*A - L*U
4
5 ans =
6
7 1.0e-15 *
8
9      0      0      0      0      0
10     0      0      0      0      0
11     0 -0.1110      0  0.0555      0
12     0      0      0  0.1110  0.1110
13     0      0      0  0.0278  0.0555
```

問題 1 と同じく回答としては 5x5 行列である。その行列もほぼ零行列である。0 以外の要素は 0 に近いである。それで  $PA = LU$  ということが確認できる。

## 3 問題 3

Listing 3 課題 3

```
1 >> A = randn(5,5);
2 >> a = DET(A)
3
4 a =
5
6 2.3562
7
8 >> b = det(A)
9
10 b =
11
12 2.3562
```

問題 2 で作成した関数を使い、行列式を計算する関数を作った。その関数を使って、MATLAB の `det()` と比較すれば、同じ結果が出る。

## 4 問題 4

Listing 4 課題 4

```
1 >> A = [1 2 3 4; 0 5 6 7; 0 0 8 9; 0 0 0 10];
2 >> b = randn(4,1);
3 >> x1 = LIN_SOL_U(A,b)
4
5 x1 =
6
7     -0.8606
8     -0.3568
9      0.0810
10    -0.0177
11
12 >> x2 = inv(A)*b
13
14 x2 =
15
16     -0.8606
17     -0.3568
18      0.0810
19    -0.0177
```

上三角行列の線形方程式の求解を求めるための関数を作成した。その結果を定義から計算されたベクトルと比較によって、正しい結果が得られたということが分かった。

## 5 問題 5

Listing 5 課題 5

```
1 >> A = randn(5,5);
2 >> b = randn(5,1);
3 >> x1 = LIN_SOL(A,b)
4
5 x1 =
6
7      0.1753
8      0.0283
9      0.4301
10     -0.2279
11     -0.3956
12
13 >> x2 = inv(A)*b
14
15 x2 =
16
17      0.1753
18      0.0283
19      0.4301
20     -0.2279
21     -0.3956
```

下三角行列の線形方程式の求解を計算する関数と問題 4 で作った上三角行列の線形方程式の求解を求めるための関数を使用することによって、どんなでも正方行列 A の線形方程式の求解を求める関数を作成した。次に、その結果を定義から計算されたベクトルと比較によって、正しい結果が得られたということが分かった。

## 6 問題 6

Listing 6 課題 6

```
1 >> A = randn(5,5);
2 >> A = A + A';
3 >> [P,L] = EIG(A)
4
5 P =
6
7     -0.2212    -0.5715    -0.2828     0.6781    -0.2910
8         0.5711     0.2008    -0.4508     0.3929     0.5252
9         0.7569    -0.0817     0.2661     0.0357    -0.5902
10        0.0855    -0.3777     0.7339     0.2332     0.5070
11       -0.2113     0.6955     0.3277     0.5746    -0.1848
12
13 L =
14
15
16        6.3320         0         0         0         0
17         0         4.1203         0         0         0
18         0         0         0.9819         0         0
19         0         0         0        -3.5206         0
20         0         0         0         0        -5.9431
21
22 >> [P,L] = eig(A)
23
24 P =
25
26     -0.2910     0.6781    -0.2828     0.5715    -0.2212
27     0.5252     0.3929    -0.4508    -0.2008     0.5711
28    -0.5902     0.0357     0.2661     0.0817     0.7569
29     0.5070     0.2332     0.7339     0.3777     0.0855
30    -0.1848     0.5746     0.3277    -0.6955    -0.2113
31
32 L =
33
34
35     -5.9431         0         0         0         0
36         0        -3.5206         0         0         0
37         0         0         0.9819         0         0
38         0         0         0         4.1203         0
39         0         0         0         0         6.3320
```

本問題では、固有ベクトルと固有値の行列を計算する関数を作成した。MATLAB の eig() 関数の結果と比べたら、P 行列では要素の符号が違ったりする場合もあり、列の順は逆順になった。さらに、L 行列では、固有値が大きい準にした (eig() と逆)。

## 7 問題 7

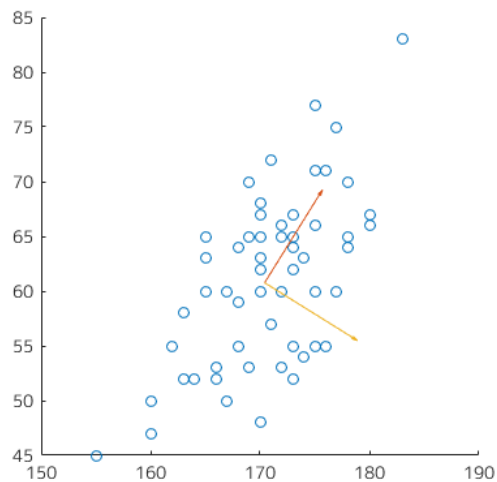
Listing 7 課題 7

```
1 >> data = load('taikaku.txt');
2 >> [P,c,a] = PCA(data)
3
4 P =
5
6     0.5285     0.8489
7     0.8489    -0.5285
8
9
10 c =
11
12     0.8419
13     0.1581
14
```

```

15
16 a =
17
18     0.8419
19     1.0000

```



a の第  $n$  番目の要素は予想通り 1 である。グラフでは、データの点を表し、 $u_1$  と  $u_2$  のベクトルも求めた。

## 8 問題 8

Listing 8 課題 8

```

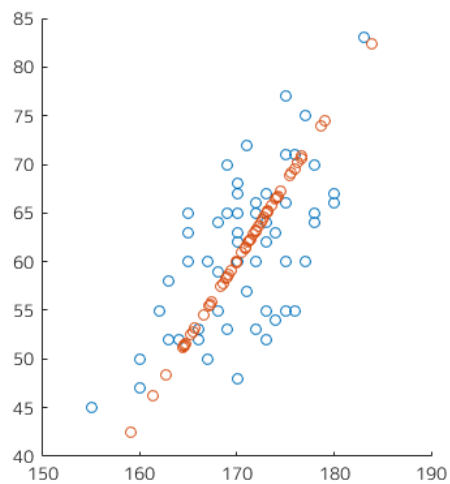
1 >> data = load('taikaku.txt');
2 >> [result, num] = PCA_represent(data, 0.9)
3
4 result =
5
6     170.5192    60.9220
7     178.9850    74.5189
8     165.7022    53.1854
9     183.9120    82.4322
10    168.8938    58.3114
11    171.9160    63.1654
12    171.9246    63.1791
13    171.3066    62.1865
14    176.5722    70.6437
15    173.9901    66.4966
16    171.6960    62.8119
17    161.3340    46.1694
18    162.6800    48.3314
19    162.6800    48.3314
20    168.5552    57.7675
21    170.8579    61.4659
22    168.3858    57.4955
23    169.9605    60.0246
24    171.1965    62.0098
25    168.9531    58.4066
26    165.4822    52.8319
27    171.3573    62.2680
28    175.4462    68.8353
29    174.4388    67.2173
30    164.6948    51.5674
31    174.0494    66.5919
32    169.1139    58.6648
33    173.0420    64.9738
34    174.4388    67.2173
35    175.6241    69.1209
36    173.0420    64.9738

```

```

37 173.9394 66.4151
38 172.2039 63.6278
39 166.5403 54.5314
40 172.7626 64.5252
41 176.2928 70.1950
42 169.9097 59.9431
43 169.1224 58.6785
44 169.3932 59.1135
45 168.5637 57.7812
46 167.3784 55.8775
47 174.1680 66.7823
48 173.2113 65.2458
49 169.9605 60.0246
50 162.6800 48.3314
51 164.5763 51.3769
52 170.8071 61.3844
53 167.1076 55.4425
54 159.0398 42.4847
55 170.8071 61.3844
56 176.6823 70.8205
57 175.8949 69.5559
58 167.1583 55.5240
59 173.5500 65.7897
60 165.7022 53.1854
61 172.4240 63.9813
62 172.5933 64.2532
63 164.4155 51.1187
64 165.2536 52.4648
65 164.6355 51.4722
66 173.1013 65.0691
67 170.5192 60.9220
68 178.6463 73.9750
69 167.2091 55.6056
70
71
72 num =
73
74 1

```



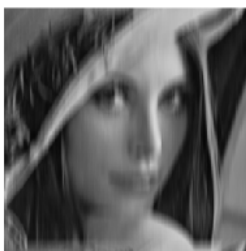
グラフの結果としては予想通り、データの点が同じ線に整理出来た。その線は  $u_1$  ベクトルに並行である。

## 9 問題 9



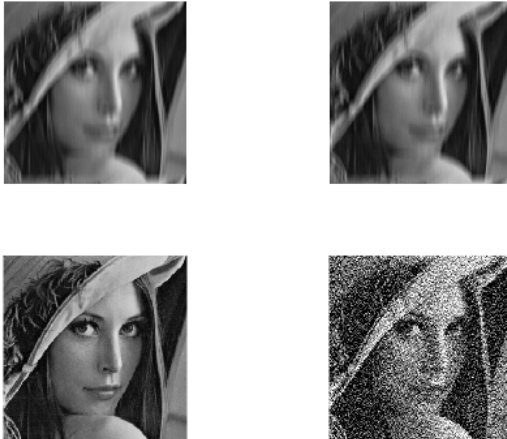
左の画像は元の画像である（上：'LENA256.pgm', 下：'ROOM256.pgm'）。右では、その 2 枚の画像の累積寄与率が 0.98 超えない再構成した形である。急に気がつかないかもしれないがよく見ると右の画像は少々質が悪くなった形である。

## 10 問題 10



左の画像は元の画像（'LENA256.pgm'）の  $A_{15}$  で劣化した形である。右の画像はその劣化画像を復元した形である。復元画像は元の'LENA256.pgm'と同じぐらいの質である。

## 11 問題 11



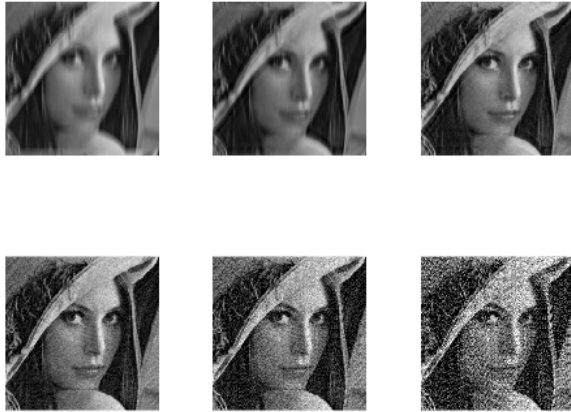
上は劣化された画像でしたそれぞれに当てはまる復元した形である。左の画像は問題 10 のノイズなしの場合である。それに比べれば、右は分散が 0.01 の正規乱数を追加したものでその画像を前の課題 10 と同じく扱ったら、復元確かにできるが、ノイズが明らかに見える（右下の画像）。

## 12 問題 12

A の固有値と固有ベクトルには複素数の要素が入る。その値は非常に小さいであるため、 $\text{inv}(A)$  にすると、その固有値の値が非常に大きくなる。その大きい値をノイズ  $N$  にかけると  $(F + \text{inv}(A) * N)$  ノイズが大変影響を与えて、見えるようになる。



### 13 問題 13



左上の画像は'LENA256DEG.pgm'で劣化された画像である。その左に（上真ん中の画像）MATLABの端末から `prob13('LENA256DEG.pgm', 0.1)` によって、つまり  $\gamma = 0.1$  で復元された形である。その次（右上）は  $\gamma = 0.01$  で復元された形である。下には左から  $\gamma = 0.001, 0.0001, 0.00001$  それぞれ復元された形である。各ステップと共に劣化が消えていくが、ノイズの量が増えている。もともと'LENA256DEG.pgm'は散が0.01の正規乱数を追加したものであったから、復元の時  $\gamma = 0.01$  の場合（右上）は、画像が最も綺麗形になる。

### 14 問題 14

結果としては、最初の2つの音は2曲が混ざった音声である。その後の2つの音はそれぞれ音源分散アルゴリズムできれいにした別れた形の2つの曲である。

## 15 本実験の感想

MATLAB を使うことが好きです。それは使いやすく、間違えてもすぐ自分の間違いが見つかります。実験そのものは割と難しいと思いますが、説明通りやたら自分ですべて納得できます。本当に勉強になった感じがします。内容も面白かったです。