

情報工学科 レポート

実験演習記録			判定・指示		
	年月日時	共同作業者			
1	2019/05/27				
2					
3					
4					
レポート提出記録					
	提出年月日	期限年月日			
初	2019/06/10	2019/06/10			
再					
科目名		テーマ担当教員	学年	学期	単位
情報工学実験2		堀田先生	3	前期	2
テーマ番号 テーマ名		学籍番号 氏 名			
主成分分析（PCA）		16268062 SALIC ERTUGRUL			

1. はじめに

本レポートでは、情報工学実験 2 の第 5 回目のレポートとして、主成分分析についての課題に取り組み、作成するプログラムの設計や実装について述べる。

2. 目的

データの分析において、あるデータの次元数（1 つのサンプルを表す数値の組の数）が大きい場合、実際にどのようにサンプル集合が分布しているのかを直接観察することはできないことがある。そこで、本実験では主成分分析 (principal component analysis, 主成分分析), または KL 展開 (Karhunen-Loeve expansion) と呼ばれる手法を高次元サンプルに適用して低い次元 (2 次元または 3 次元) のアフィン部分空間に射影することで、高次元サンプルの分布を観察したり、次元数を削減したりすることを試みる。主成分分析に関する知識を増やせることを、そしてその際に使用する Octave 言語に慣れることを目的とする。

3. 実験環境

本実験では、全ての課題を EDEN において、「Octave 言語」で行う。

4. 課題 1

課題 1 では配布資料の `sample2d.txt` に対して主成分分析を適用し、最大固有値に対応する固有ベクトルを求め、図示するプログラムを作成する。図に示す直線はサンプル分布を楕円で表した時の長軸であり、傾きは固有ベクトルで決まり、かつサンプル平均を通る直線である。また、この固有ベクトルにサンプルを正射影して求める z_i のサンプル分散を求める。

4.1 課題 1 の設計

課題 1 では、次のステップに従ってプログラムを作成し、図を出力させる。

- (1) `sample2d.txt` を読み込む
- (2) サンプルの平均値を取り、共分散行列を計算する
- (3) 共分散行列から固有ベクトルと固有値を取得する
- (4) 固有値を降順でソートし、最大固有値に対する固有ベクトルを求める
- (5) サンプルを 2 次元平面にプロットし、直線を引く
- (6) z_i のサンプル分散を求める

なお、共分散行列 Σ を式 1 と式 2 で求める。

$$R = \frac{1}{n} \sum_{i=1}^n x_i x_i^T \quad (1)$$

$$\Sigma = R - m m^T \quad (2)$$

課題 1 で作成したプログラムを次に示す。

```

1. clear all; close all; % すべてのグローバル変数/ウィンドを消去
2. % STEP(1)
3. X=load("sample2d.txt"); % sample2d.txt の読み込み
4. % STEP(2)
5. [d,n]=size(X); % X の次元 d とサンプル数 n を取得
6. R=zeros(d,1);
7. m=mean(X,2); % サンプルの平均値
8. for ii = 1 : n
9.     R=R+(X(:,ii)*X(:,ii)');
10. end
11. R=R./n;
12. cv=R-(m*m');
13. % STEP(3)
14. [v, lambda] = eig(cv); % 固有ベクトルと固有値を取得
15. % STEP(4)
16. [sorteigen, order] = sort(diag(lambda),'descend');
17. % 固有値をソートする
18. U = v(:,order); % 同じやり方で固有ベクトル内をソーティング
19. u=U(:,1); % 最大固有値に対する固有ベクトル
20. % STEP(5)
21. slope=u(2)/u(1); % 直線の傾き
22. xx=[min(X(1,:)):.05:max(X(1,:))];
23. figure(1), clf, hold on, plot(X(1,:),X(2,:), "bo");
24. % サンプルを 2 次元平面に青点でプロット
25. figure(1), plot(xx,slope.*xx+m(2)-slope.*m(1), "g-");
26. % 直線 0.1x+0.1 を緑の直線で描画
27. axis square;
28. % STEP(6)
29. Z=zeros(n,1);
30. for i = 1 : n
31.     Z(i)=X'(i,:) * u;
32. end
33. % STEP(6) -version2-
34. Z2=zeros(1,n);
35. for i = 1 : n
36.     Z2(i)=u' * (X(:,i)-m);
37. end
38. Z_bunsan=u' * cv * u;

```

本プログラムでは、初めに sample2d.txt を読み込み (STEP1)、主成分分析に必要な数値を計算する (STEP2)。次に、eig 関数で、固有ベクトルと固有値を求める (STEP3)。固有ベクトルは、

$$v = \begin{pmatrix} -0.74185 & -0.67057 \\ -0.67057 & 0.74185 \end{pmatrix}$$

固有値は,

$$lambda = \begin{pmatrix} 0.96743 & 0 \\ 0 & 6.0377 \end{pmatrix}$$

となる．求めた固有値を降順にソートする（STEP4）．ソートしたものから固有値が最大のものを取り，図を作成する（STEP5），また z_i とそのサンプル分散の計算（STEP6）を行う．ただし， z_i の求め方に関して，式3と式4に示す式両方を使用し， Z と $Z2$ という2つの行列を計算する．式3で求めた結果（ Z ）と式4で求めた結果（ $Z2$ ）を節4.2の考察において比較する．

$$z_i = x_i^T u = \sum_{k=1}^d x_{ik} u_{ik} \quad (i = 1, \dots, n) \quad (3)$$

$$z_i = U^T(x_i - m) \quad (4)$$

4.2 課題1の実装結果および考察

課題1の実行結果を図1に示す．

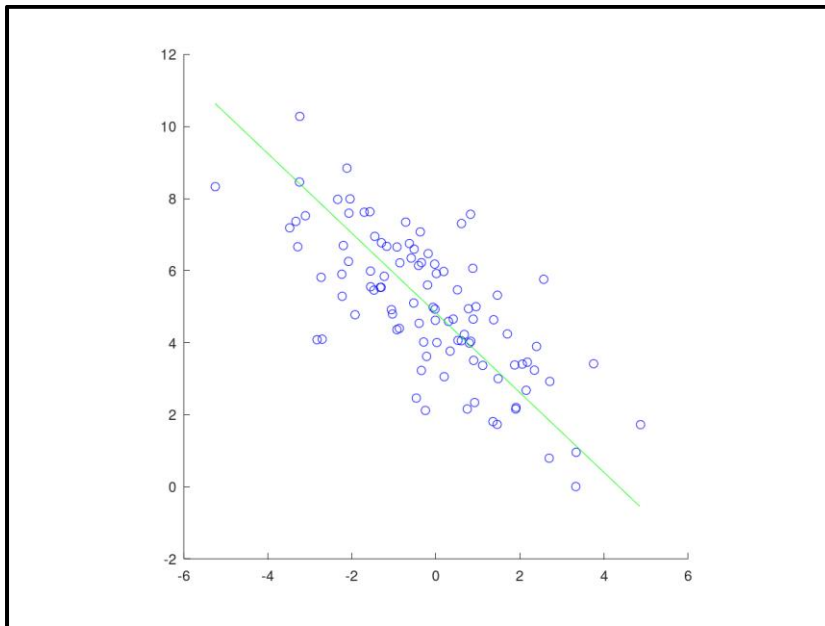


図1 課題1の結果（青点：サンプル，直線：平均値を通り，固有ベクトルを傾きにしている直線）

課題1より，レポート課題1の結果が，テキストの図5と同様であり，正しい結果が得られたことが分かる．課題1では，2次元のサンプル分布を1次元で示すことができた．それによって，サンプル分布の傾きや傾向について分析できると思われる．

また，求めた z_i のサンプル（サイズが大きいため最初の10個のみ）を表1に示す．なお，表1に式3と式4で求めた z_i 両方も示している．

表 1 式 3 と式 4 で求めた z_i のサンプル

z_i の結果	
Z	Z2
2.1358	-1.705
0.548	-3.2927
1.1074	-2.7333
6.7981	2.9573
2.4407	-1.4001
4.2525	0.41171
4.6046	0.76383
2.6634	-1.1774
4.3839	0.54316
7.297	3.4563

表 1 より、式 3 と式 4 で求めた z_i が異なっていることが分かる。しかしそこで、 $Z-Z2$ が定数であり、 Z の各値が $Z2$ より 3.8407 ほど大きいことに気づく。式 3 で求めた結果が式 4 で求めた結果と同じにはならない。その理由が分からなかった。しかし、 $Z-Z2$ の各値が 3.8407 であるので、両方の分散値が同じであることが推測できる。

z_i の分散をプログラムの `Z_bunsan` 変数で求めた。その結果を次に示す。

$$z_i \text{ の分散値} = 6.0377$$

その結果が最大固有値に等しいことが分かる。

よって、課題 1 において気づいた・分かったことを次に求める。

- (1) 2 次元のサンプルを 1 次元で示すことによって、サンプル分布の傾向等について分析することができるようになった。
- (2) z_i の求め方に関して、式 3 (テキストの式 1) と式 4 (テキストの式 8) それぞれで求めた z_i が異なってしまった。しかし同じ分散をしていた。
- (3) z_i サンプルの分散と共分散行列の最大固有値が同じであった。

5. 課題 2

Step2 では直接サンプル分布を観察できない高次元サンプルに対して主成分分析を適用し、サンプル分布の概略を可視化してみる。具体的には、機械学習の分野で古くから用いられている `iris flowerdataset` に対して主成分分析を適用する。このデータセットは、3 種類の花のアヤメ (*Iris setosa*, *versicolor*, *virginica*) を各種 50 個体選び、それぞれの花のがく片の長さ、花弁の長さ、花弁の幅を調べた結果からなるデータである。すなわち、サンプル総数は 150、次元数は 4 となるので直接サンプルの分布を観察することができない。

そこで Step2 では、主成分分析により 4 次元空間上のサンプルを 2 次元空間上に射影することで、元のサンプル分布が、おおよそではあるが、どのようなになっているのかを観察できるようにしてみる。

課題 2 では、配布資料の 1 つである iris4d.txt のデータを使用する。

このファイルを Step1 と同様に読み込んで、そのサイズを調べれば 4 行 150 列となっている。すなわち、各列がそれぞれの花を表し、各行が 4 つの長さの幅の値となっている。また、3 種類の花の内訳は

- 1 列から 50 列目 : setosa
- 51 列目から 100 列目 : versicolor
- 101 列目から 150 列目 : virginica

となっており、各行は

- 1 行目 : がく片(sepal) の長さ
- 2 行目 : がく片の幅
- 3 行目 : 花弁(petal) の長さ
- 4 行目 : 花弁の幅

となっている。

また分布の近似について調べるために**累積寄与率**も計算する。

5.1 課題 2 の設計

課題 2 では、次のステップでプログラムを作成する。

- ➔ STEP(1): 課題 1 の同様に主成分分析で z_i を求める。
- ➔ STEP(2): 求めた分布を 2 次元で出力。
- ➔ STEP(3): 累積寄与率を計算して、グラフを作成する。

ただし、2 次元の分布を得るために、**STEP(1)** で z_i を計算する際は選び出す固有ベクトルを 1 本ではなく 2 本にする。それは固有ベクトルの本数と次元数が同じだからである。

また、**STEP(2)** では分布を出力する時には「setosa を赤い丸で、versicolor を緑の丸で、virginica を青い丸で」出力する。

STEP(3) では、累積寄与率を計算する。累積寄与率について、以下の説明をテキストより引用する。

「累積寄与率とは主成分分析の際に求めた固有値を降順に並べたものを $\lambda_1, \dots, \lambda_d$ とすると

$$\sum_{i=1}^r \lambda_i / \sum_{j=1}^d \lambda_j \quad (5)$$

によって計算される。ただし d は共分散行列の階数(行列の列または行の中で独立であるものの最大個数, 0 でない固有値の個数)である。この値は上位 r 個の固有値(分散)の総和を、すべての固有値の総和(各軸の分散の総和)で割ったものであり、元のサンプル分布をどの程度近似しているかを表す値と解釈できる。つまり、この値が 1 に近ければ近いほど r 次元空間のサンプルは元のサンプル分布を良く近似しているといえる。」(堀田 政二, 東京農工大学, 情報工学科, 情報工学実験 2, 2019 年度第 5 回講義資料より)

上記のステップに従ってプログラムを作成した。そのプログラムを次に示す。ただし、STEP(1)は課題1とほぼ同様であるため、STEP2からのコードを次に示す。なお、プログラム全体をレポートを提出する際に添付する。

```
1. % STEP(2):分布の出力
2. % Zi を 3 種類の花に分割
3. x1 = Z(1,1:50); % setosa
4. y1 = Z(2,1:50);
5. x2 = Z(1,51:100); % versicolor
6. y2 = Z(2,51:100);
7. x3 = Z(1,101:150); % virginica
8. y3 = Z(2,101:150);
9. % 2次元での分布を出力する
10. xx=[min(X(1,:)):.05:max(X(1,:))];
11. figure(1),clf,hold on,plot(x1,y1,"ro"); % setosa を赤い丸で
12. figure(1),plot(x2,y2,"go"); % versicolor を緑の丸で
13. figure(1),plot(x3,y3,"bo"); % virginica を青い丸で
14. % STEP(3):累積寄与率を計算して、グラフを作成する
15. % すべての固有値の総和
16. lambda_d=0;
17. for i = 1 : d
18.     lambda_d=lambda_d+sorteigen(i);
19. end
20. % 上位 r 個の固有値（分散）の総和
21. lambda_R=zeros(d,1);
22. lambda_R(1)=sorteigen(1);
23. for i = 2 : d
24.     lambda_R(i)=lambda_R(i-1)+sorteigen(i);
25. end
26. % 累積寄与率を保存するベクトルの定義
27. rate=zeros(d,1);
28. % r=1,2,3,4 の場合の累積寄与率を rate に格納
29. for i = 1 : d
30.     rate(i)=lambda_R(i)/lambda_d;
31. end
32. % 寄与率をグラフ化する
33. r=[1,2,3,4];
34. figure(2),hold on,plot(r,rate,"ro-");
35. axis square;
```

5.2 課題 2 の実装結果および考察

課題 2 のプログラムでは 2 つのグラフを出力する．その 1 つ目の figure(1) は 3 種類の花の分布図である．figure(1) は図 2 に示す．

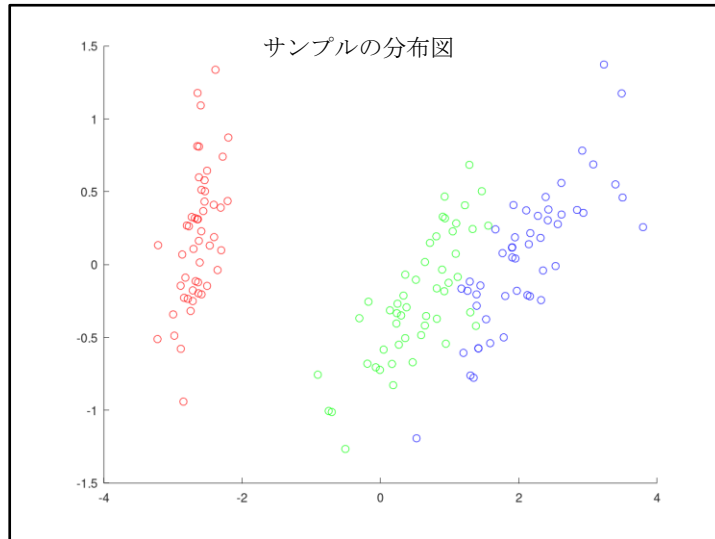


図 2 3 種類の花のサンプル分布図

プログラムで出力する 2 つ目のグラフ figure(2) は累積寄与率のグラフである．figure(2) は図 3 に示す．

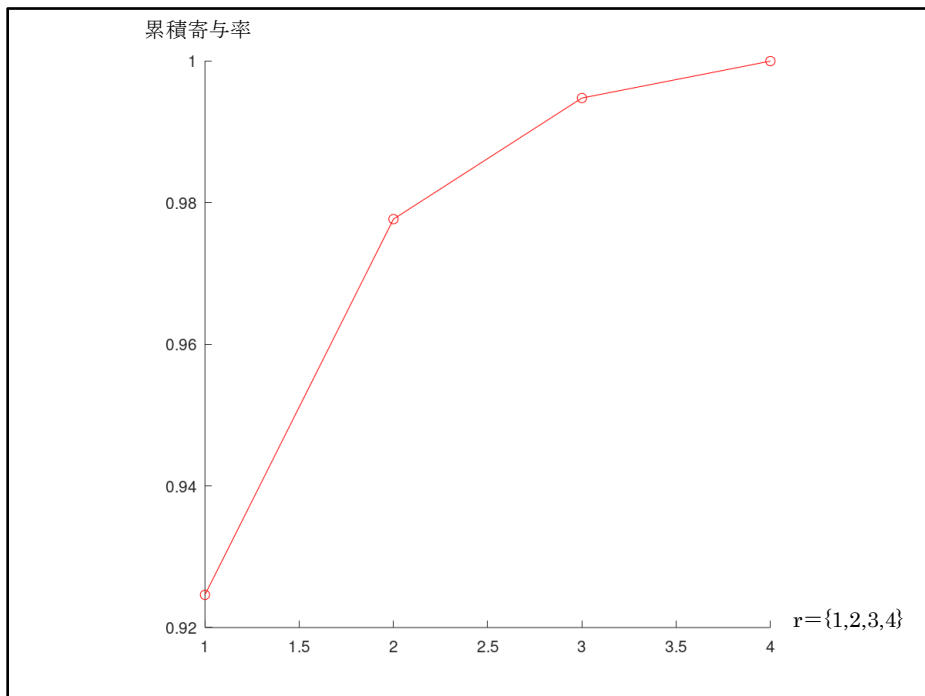


図 3 累積寄与率のグラフ

図 2 では，固有値と固有ベクトルを 3 種類の花の合同のデータから計算するため，主成分分析によって同じグラフ上に 3 種類の花を分類できている．また，図 3 より，累積寄与率は 1 次元時点で一番小さく，次元数が大きくなるにつれて累積寄与率も大きくなっており， $r=d$ の時に 1 になるといえる．

図 2 より，4 次元の観察できないデータから，2 次元の観察できるデータが得られたことがわか

る．4次元で観察できないデータを2次元で観察でき，3種類の花の分類をはっきり観察することができた．分類の正確さから，主成分分析を，**パターン認識**等の技術に利用できると考えた．

また，3種類の花の各行の平均値をとってみた．それを表2に示す．

表2 3種類の花の各行の平均値

	がく片(SEPAL) の長さ	がく片の幅	花弁(PETAL) の長さ	花弁の幅
SETOSA (赤点)	5.006	3.428	1.462	0.246
VERSICOLOR (緑点)	5.936	2.77	4.26	1.326
VIRGINICA (青点)	6.588	2.974	5.552	2.026

図2より，SETOSAという花がVERSICOLORとVIRGINICAに比べると異なる分布をしていることが分かる．それを表3でも確認することができる．SETOSAの花弁(petal)の長さの平均値が他の2類の花に比べるとずっと小さい値になっている．(約1/3～1/4倍になっている)また，花弁の幅の平均値も小さい値となっており，異なる分布になることが，平均値よりも確認できる．

6. 課題3

$$\tilde{x}_i = m + Uz_i = m + UU^T(x_i - m) \quad (6)$$

課題3では配布データのiris4d.txtに対して主成分分析を適用し，固有値の大きい上位r本の固有ベクトルを用いて，式(6)により \tilde{x}_i を求める．その後，元のサンプル x_i との2乗誤差の平均

$$error_r = \frac{1}{n} \sum_{i=1}^n \|x_i - \tilde{x}_i\|^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \tilde{x}_i)^T (x_i - \tilde{x}_i) \quad (7)$$

を計算する．その結果で横軸をr，縦軸を $error_r$ としたグラフ出力させる．

6.1 課題3の設計

課題3でもプログラムを3つのSTEPで設計する．それを次に示す．

- ➔ STEP(1): 課題1の同様に主成分分析で固有値を求め，降順に並べた後に，対応する固有ベクトルも並べ替える．並べ替えられた固有ベクトルの行列はUである．
- ➔ STEP(2): 誤差errorを求める
- ➔ STEP(3): 誤差のグラフを出力する

STEP(2)では，誤差errorを計算する時に，最大固有値に対する固有ベクトルを選び出す際は，次元数毎の場合を考え，次元数と同じ本数の固有ベクトルを選び出す．

STEP(1)は課題1・2とほぼ同様であるため，コードは省略する．STEP(2)からのコードを次に示す．なお，課題3のプログラム全体を，レポートを提出する際に添付する．(ファイル名；step3.m)

```

1. % STEP(2): 誤差 error を求める
2. error=zeros(d,1);
3. for i = 1:d
4.     u=U(:,1:i); % 最大固有値に対する固有ベクトル i 本
5.     Z=zeros(i,n); % Zi を求める
6.     for j = 1 : n
7.         Z(:,j)=u'*(X(:,j)-m);
8.     end
9.
10.    X_new=zeros(d,n); % 式 (6) で  $\tilde{x}_i$  を求める (X_new)
11.    for j = 1 : n
12.        X_new(:,j)=m+(u*Z(:,j));
13.    end
14.
15.    for j = 1 : n % 誤差平均値の計算 (式 (7))
16.        error(i)=error(i)+((X(:,j)-X_new(:,j))'*(X(:,j)-X_new(:,j)));
17.    end
18.    error(i)=error(i)./n;
19. end
20. % STEP(3): 誤差のグラフを出力
21. % 次元数毎の誤差の平均値を kekka に格納
22. kekka=[1,2,3,4;error'];
23. figure(1),clf,hold on,plot(kekka(1,:),kekka(2,:),"bo");

```

6.2 課題 3 の実装結果および考察

課題 3 のプログラムを実装した結果は図 4 に示す。図 4 は横軸を r 、縦軸を $error_r$ としたグラフである。

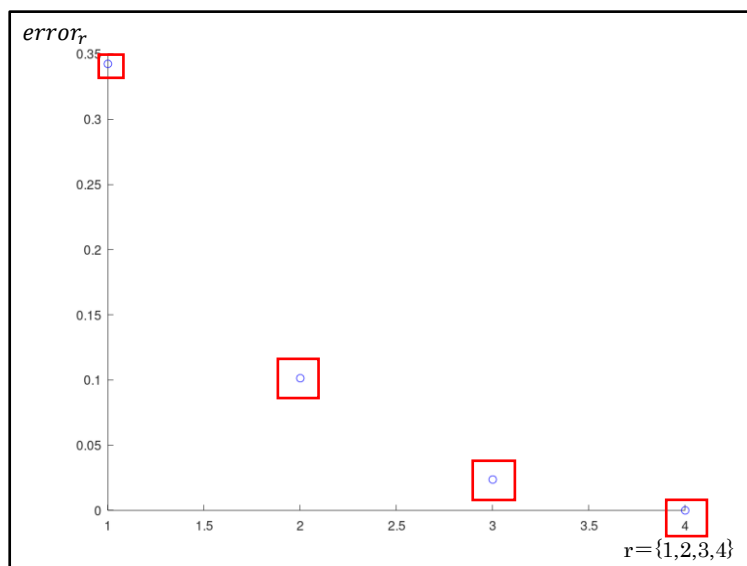


図 4 課題 3 の結果 (横軸が r 、縦軸が $error_r$ であるグラフ)

図 4 より、2 乗平均誤差は次元数が大きくなるにつれて小さくなっていることがわかる。次元数が元の時は誤差が 0 となる。

主成分分析を用いることによって、元のサンプルを近似し、次元数を減らすことができる。また近似したデータからもとのサンプルを復元できる。復元する際の誤差が、近似した次元数によって異なる。次元数が多ければ多いほど（元の次元数より小さいではあるが）誤差も小さくなっていく。元のデータより少ない次元数に近似することによって、データを少ない記憶容量で保存することができるとも考えられる。

課題 3 では、主成分分析によって、少ない誤差でデータをより少ない記憶容量で保存することができることに気づいた。

7. 挑戦課題 1

挑戦課題 1 では、自由にデータを収集し、そのサンプル集合に対して主成分分析を適用し、2 次元散布図を使ってサンプル分布の特性について考察する。このとき、以下の項目について報告する。

- ・どのようなデータであるか
- ・データの内訳の詳細
- ・2 次元空間でのサンプル分布がどの程度、元の分布をよく近似しているか
- ・散布図から何が読み取れるか
- ・作成したプログラム

7.1 挑戦課題 1 の設計

挑戦課題 1 で使用するデータは“glass.txt”というデータであり、そのデータを提出する際にレポートに添付する。データは（<<https://www.kaggle.com/uciml/glass/downloads/glass-classification.zip/1>>, 閲覧日：2019 年 6 月 10 日）よりダウンロードしたデータである。

本課題で使用するデータは、6 つのガラスの種類のサンプルを 214 個分析した結果である。列は 9 つの元素となっている。データは各サンプルが含む元素の量を表している。

ガラスの種類の内訳は、

- | | |
|--------------------------|--------------------------|
| ・ 1 行から 70 行目 ➡ Type1 | ・ 164 行から 176 行目 ➡ Type4 |
| ・ 71 行から 146 行目 ➡ Type2 | ・ 177 行から 185 行目 ➡ Type5 |
| ・ 147 行から 163 行目 ➡ Type3 | ・ 186 行から 241 行目 ➡ Type6 |

となっており、各列は

- | | |
|-------------------------------|-----------------------|
| ・ 1 列目 ➡ RI: refractive index | ・ 6 列目 ➡ K: Potassium |
| ・ 2 列目 ➡ Na: Sodium | ・ 7 列目 ➡ Ca: Calcium |
| ・ 3 列目 ➡ Mg: Magnesium | ・ 8 列目 ➡ Ba: Barium |
| ・ 4 列目 ➡ Al: Aluminum | ・ 9 列目 ➡ Fe: Iron |
| ・ 5 列目 ➡ Si: Silicon | |

となっている。しかし、データは上を転置したものを使う。（つまり PCA する際は行と列を転置して分析を行う。）

本課題でも 3 つの STEP でプログラムを作成する。その STEP を次に示す。

- ➔ STEP(1): 課題 1 の同様に主成分分析で固有値を求め、降順に並べた後に、対応する固有ベクトルも並べ替える。その結果から式 (4) で Z_i を計算する。
- ➔ STEP(2): 散布図の出力

➡ STEP(3): 累積寄与率を計算する

本課題で作成するプログラムを次に示す. ただし, STEP(1)は課題 1・2 とほぼ同様であるため省略する. プログラム全体をレポートと一緒に提出する.

```
1. % STEP(2): 散布の出力
2. x1 = Z(1,1:70); % Type1
3. y1 = Z(2,1:70);
4. x2 = Z(1,71:146); % Type2
5. y2 = Z(2,71:146);
6. x3 = Z(1,147:163); % Type3
7. y3 = Z(2,147:163);
8. x4 = Z(1,164:176); % Type4
9. y4 = Z(2,164:176);
10. x5 = Z(1,177:185); % Type5
11. y5 = Z(2,177:185);
12. x6 = Z(1,186:214); % Type6
13. y6 = Z(2,186:214);
14.
15. xx=[min(X(1,:)).05:max(X(1,:))];
16. figure(1),clf,hold on,plot(x1,y1,"ro"); % Type1 を赤い丸で
17. figure(1),plot(x2,y2,"go"); % Type2 を緑の丸で
18. figure(1),plot(x3,y3,"bo"); % Type3 を青の丸で
19. figure(1),plot(x4,y4,"ko"); % Type4 を黒の丸で
20. figure(1),plot(x5,y5,"yo"); % Type5 を黄色の丸で
21. figure(1),plot(x6,y6,"co"); % Type6 をシアンの丸で
22.
23. % STEP(3): 累積寄与率を計算する
24. % すべての固有値の総和
25. lambda_d=0;
26. for i = 1 : d
27.     lambda_d=lambda_d+sorteigen(i);
28. end
29. % 上位 r 個の固有値 (分散) の総和
30. lambda_r=0;
31. for i = 1 : r
32.     lambda_r=lambda_r+sorteigen(i);
33. end
34. % 累積寄与率の計算
35. rate=lambda_r/lambda_d;
36.
37. axis square;
```

7.2 挑戦課題 1 の実装結果および考察

挑戦課題 1 では、データを読み込み、データのサンプル集合に対して主成分分析を適用し、2 次元散布図を出力した。その結果を図 5 に示す。

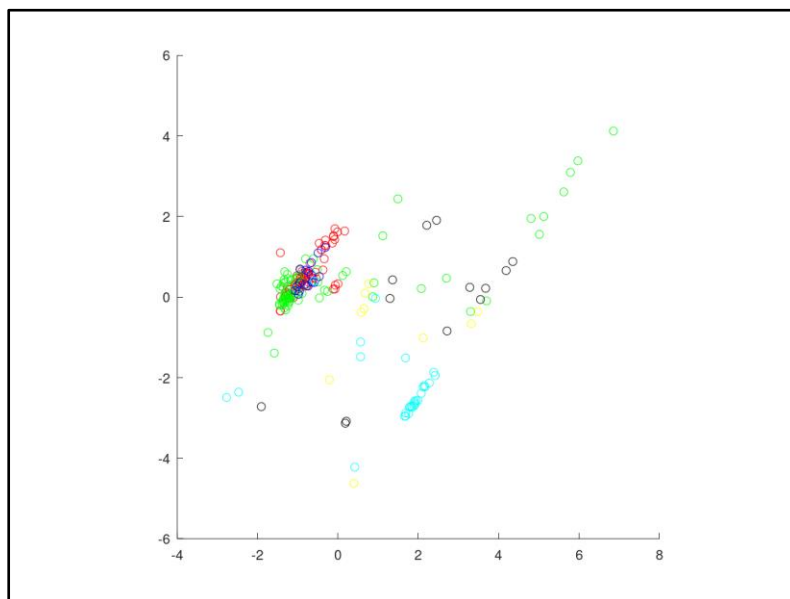


図 5 課題 4 の結果 データを 2 次元に近似した散布図

図 5 より、9 次元のデータでも 2 次元に近似することができて、観察できるということが分かる。図 5 では 6 種類のガラスの分布が観察できる。Type1, 2, 3 の 2 種類のガラスが近い分布をしているが、Type4, 5, 6, 7 は独特な分布をしており、Type4, 5, 6, 7 に関して 2 次元でも分類が簡単にできると考えられる。

また、累積寄与率 $rate$ が、

$$rate = 0.73940$$

となった。累積寄与率が 1 に近ければ近いほど誤差が少なくなるが、今回は 9 次元のデータを使用しており、次元数の差が大きいため、累積寄与率 $rate < 0.75$ となったと考えられる。よって、近似に関しては優れた近似ができたとは言えない状態である。

8. 挑戦課題 2

本課題では、ウェブなどで公開されている画像データに対して主成分分析を適用し固有顔に対応する画像を求める。今回は <https://cs.nyu.edu/~roweis/data.html> よりダウンロードした顔データセットの一部を使用する。使用する元の顔を図 6 に示す。



図 6 挑戦課題 2 データとして使用するオリジナル (元) の画像

8.1 挑戦課題2の設計

固有顔とは、主成分分析により求めた固有ベクトルのことである。固有ベクトルを計算する時に、以下のステップで計算を行うことにした。

- STEP(1): 64x64 の画像を読み込み、4096x1 のベクトルにする。
(今回はサイズが 64x64 の 16 個の画像を使用するので、サイズが 4096x16 の行列 X を得る)
- STEP(2): 行列 X の平均ベクトルを求める
- STEP(3): 行列 X の共分散行列を求め、固有値と固有ベクトルを求める。
- STEP(4): 固有値を降順に並べ替え、最大 16 の固有値に対応する 16 本の固有ベクトルを求める。
- STEP(5): 求めた固有ベクトルの 1 本が 4096x1 のサイズであり、それを 64x64 サイズに reshape して画像化する。

そのプログラムを次に示す。

```
1. clear all; , close all;
2. d=64;
3. n=16;
4. load Olivettifaces; % https://cs.nyu.edu/~roweis/data.html からダウンロードし
   た顔のデータセット
5. for i=1:n
6.     X(:,i)=faces(:,n.*(i-1)+1); % 16 人の顔の画像データ (画像の画素値を縦に並べたベ
   クトルからなる行列)
7.     a=reshape(X(:,i),[d d]); % 元の画像を png として出力するためにサイズ変換を行う
8.     a=uint8(a); % double 型から int 型へ変換
9.     namae=[num2str(i),".png"]; % 元の画像を保存するファイル名を指定
10.    imwrite(a,namae); % 画像の保存
11. end
12. m=mean(X,2); % サンプルの平均値
13. cv=cov(X'); % 共分散行列の計算
14. [v, lambda] = eig(cv); % 固有ベクトルと固有値を取得
15. [sorteigen, order] = sort(diag(lambda),'descend'); % 固有値をソートする
16. U = v(:,order); % 同じやり方で固有ベクトル内をソーティング
17. u=U(:,1:16); % 最大固有値に対する固有ベクトル
18. figure("mean")=reshape(m(:,1),[d d]); % 平均ベクトルの画像か
19. % 新しい画像の出力・保存
20. for i = 1:n
21.     %u(:,i)=u(:,i)+m;
22.     a=reshape(u(:,i),[d d]);
23.     a=uint8(a);
24.     namae=[num2str(i),"_new.png"]; %結果の保存
25.     imwrite(a,namae);
26. end
```

8.2 挑戦課題2の実装結果および考察

挑戦課題2のプログラムを実装した結果を図8に示す。図8よりも分かるように、正しい結果が得られなかった。今回は時間が足りなく、その理由や正しいやり方について論じることができなかったが、レポート提出後にも、その問題に課題として取り組む。

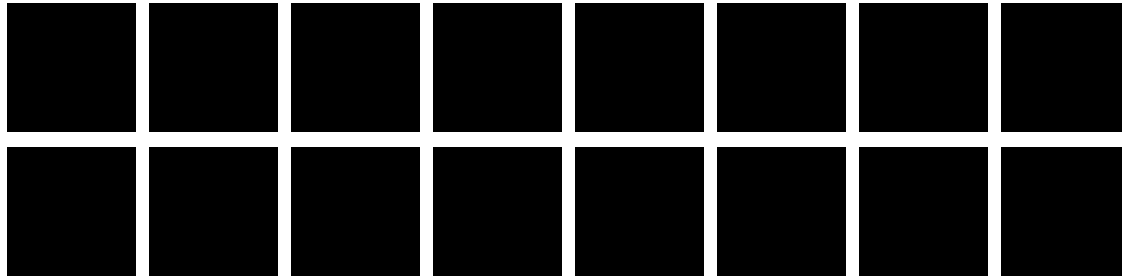


図8 挑戦課題2の結果

9. おわりに

本レポートでは主成分分析を行うことによって、その方法や目的について学んだ。また、プログラム言語としてOctaveを使用した。これはOctaveで取り組んだ初めての課題となった。

課題1によって、主成分分析で2次元のサンプルを1次元の落とし、その傾向等について観察できるようになった。また課題2では、多次元で観察できないデータに対して、主成分分析を行うことによってデータを観察・分析・分類することができた。課題3では多次元のデータの次元数を少なくすることによって、ほぼ同じデータをもっと少ない記憶容量で保存することができた。

データ分析が重要視されている現代では、主成分分析を行って何か分析できることは、非常に大切な能力・知識だと考えられる。今度も、PCAに関する知識を増やすことを、そしてPCAを今度の研究・勉学に役に立てることを期待している。

10. 参考文献

- [1] 堀田 政二，東京農工大学，情報工学科，情報工学実験2，2019年度第5回講義資料
- [2] <<https://www.kaggle.com/uciml/glass/downloads/glass-classification.zip/1>>，閲覧日：2019年6月10日
- [3] <https://octave.org/doc/v4.2.1/Two_002dDimensional-Plots.html>，閲覧日：2019年6月10日
- [4] <<https://cs.nyu.edu/~roweis/data.html>>，閲覧日：2019年6月10日