

MATLABによる次元削減

<https://web.tuat.ac.jp/~s-hotta/pca/>

概要

例えば体重と身長の数値の組み合わせで表された人物の集合（数値サンプル）が手元にあるとき、人物の分布や体重と身長の関係性を観察したい場合や、計算量を削減するために、なるべく少ない記憶容量で元のサンプルを近似的に表現したい場合がある。特に、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。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-Loève 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。

実験の目的

実験の目的は、2次元の場合、図1左に示すように、与えられた2次元空間上のサンプル集合をよく近似する1次元集合（直線）を探すことである。2次元空間上には無数の直線を引くことができるが、本実験では図1右に示すように、2次元のサンプルを直線に正射影したときに、元の分布をよく表すような直線を探すことを考える。このような近似は、先に述べたサンプル分布の可視化や圧縮に応用することができるため、非常によく用いられる。本実験では、もっとも単純な2次元サンプルを1次元空間上で表すことから始めて、高次元サンプルの圧縮や復元について実験を通して学ぶこととする。

なお本実験では、行列やベクトル演算を多用するためMATLABでの実装を行う。MATLABは主に数値計算を行うためのインタプリタ型のプログラミング言語であり、行列やベクトル同士の演算、データ解析、可視化等が得意な言語である。したがって、これらのプログラムを利用して実験を行う際には実装の手間が少なくすむという利点がある。

実験の流れは以下のように3つの必須ステップと2つの挑戦課題からなっており、必須ステップでは1ステップあたり1コマを目安としている。Step1では講義形式でMATLABの基本的な使い方と主成分分析の考え方について学ぶ。その後、このテキストを用いて各自のペースで以降のStep2とStep3へ進んでよい。分からないことや困ったことが発生した場合には、TAや教員に遠慮なく質問しよう。

- Step1: MATLABを使いながら2次元サンプルを1次元に圧縮しよう（20点）
- Step2: 高次元サンプルの分布を2次元空間で可視化してみよう（20点）
- Step3: 高次元サンプルを少ない正規直交ベクトルの線型結合で表現しよう（20点）
- 挑戦課題1: 主成分分析を実データへ適用してみよう（20点）
- 挑戦課題2: 固有顔を求めてみよう（20点）

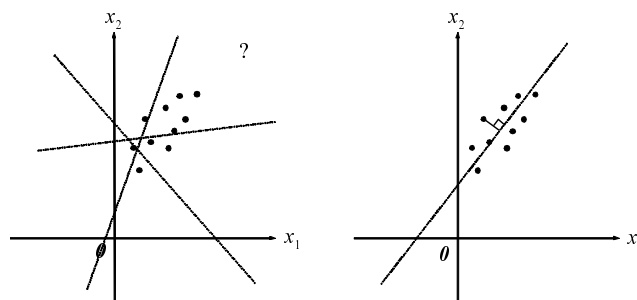


図1: 近似の概念図

1 Step1: MATLAB を使いながら 2次元サンプルを 1次元に圧縮しよう

主成分分析の概念を図 2 に示す 2次元空間上に分布するサンプル集合を用いて説明する。図 2 ではサンプルを点で表しているが、主成分分析は直感的にこのサンプルの分布を楕円で近似した場合の長軸と短軸の向きを求める処理と見なすことができる。ここで長軸を選ぶ基準は、その長軸にサンプルを正射影した場合に、射影先でのサンプル分布の分散が最も大きくなるようなものを選ぶ。そして短軸は、射影先のサンプルの分散が長軸の次に大きくなり、かつ長軸に直交するものを選ぶ。これらの軸の方向を表すベクトルは必ず原点を通ることに注意しよう。図 2 の例では、求めたい方向は \mathbf{u}_1 である。この \mathbf{u}_1 の方向で表される直線上にサンプルを正射影すれば、2次元のサンプル分布を 1次元空間上に、もとの状態を大きく損なうことなく表現することができる。

1.1 問題設定

上記の主成分分析を定式化してみる。 n 個ある d 次元のサンプルのうち、第 i サンプルを $\mathbf{x}_i = (x_{i1} \cdots x_{id})^\top$ とする。ここで \mathbf{x}_i が、ある d 次元ベクトル $\mathbf{u} = (u_1 \cdots u_d)^\top$ との内積によってスカラー値

$$z_i = \mathbf{x}_i^\top \mathbf{u} = \sum_{k=1}^d x_{ik} u_k \quad (i = 1, \dots, n) \quad (1)$$

に変換されたとする。このとき、この z_i が d 次元サンプル \mathbf{x}_i の良い近似となるように \mathbf{u} を設定したい。そのための基準として、ここでは z_i の分散の最大化という基準を用いることにする。

n 個のサンプル $\mathbf{x}_i (i = 1, \dots, n)$ が \mathbf{u} により z_i に変換されたとする。このとき、 z_i のサンプル平均は

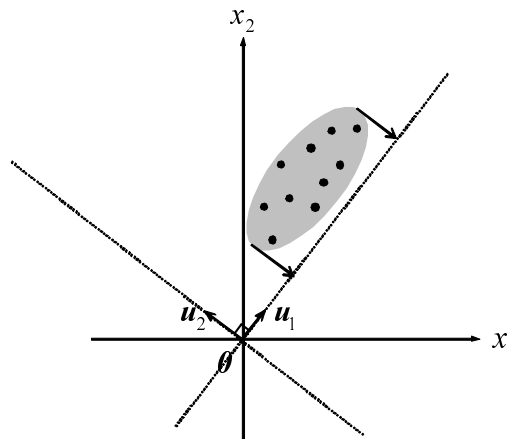


図 2: 主成分分析の概念図

$\bar{z} = \frac{1}{n} \sum_{i=1}^n z_i$ となり, z_i のサンプル分散は

$$\begin{aligned}
 & \frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2 \\
 &= \frac{1}{n} \sum_{i=1}^n z_i^2 - \bar{z}^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i^\top \mathbf{u})^2 - \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{u} \right)^2 \\
 &= \frac{1}{n} \sum_{i=1}^n (\mathbf{u}^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{u}) - (\mathbf{m}^\top \mathbf{u})^2 \\
 &= \frac{1}{n} \sum_{i=1}^n (\mathbf{u}^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{u}) - (\mathbf{u}^\top \mathbf{m} \mathbf{m}^\top \mathbf{u}) \\
 &= \mathbf{u}^\top \left(\frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top - \mathbf{m} \mathbf{m}^\top \right) \mathbf{u} \\
 &= \mathbf{u}^\top (\mathbf{R} - \mathbf{m} \mathbf{m}^\top) \mathbf{u} \\
 &= \mathbf{u}^\top \mathbf{\Sigma} \mathbf{u}
 \end{aligned} \tag{2}$$

となる. ここで $\mathbf{m} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ であり d 次元の平均ベクトルである. また, \mathbf{R} は $d \times d$ の自己相関行列と呼ばれるもので

$$\mathbf{R} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \tag{3}$$

である. さらに, $\mathbf{\Sigma}$ は $d \times d$ の共分散行列であり $\mathbf{\Sigma} = \mathbf{R} - \mathbf{m} \mathbf{m}^\top$ が成り立つ. ここまでで z_i のサンプル分散は $\mathbf{u}^\top \mathbf{\Sigma} \mathbf{u}$ と表せることを示した.

1.2 解法

次に, サンプル分散 $\mathbf{u}^\top \mathbf{\Sigma} \mathbf{u}$ を最大にする \mathbf{u} を求めるが, これは $\mathbf{u}^\top \mathbf{u} = 1$ という制約条件のもとで $\mathbf{u}^\top \mathbf{\Sigma} \mathbf{u}$ を最大化する最適化問題として書ける. このような制約付き最大化・最小化問題を解くための定番手法は Lagrange 乗数法である. すなわち, Lagrange 乗数 λ を用いてラグランジュ関数を

$$L(\mathbf{u}, \lambda) = \mathbf{u}^\top \mathbf{\Sigma} \mathbf{u} - \lambda(\mathbf{u}^\top \mathbf{u} - 1) \tag{4}$$

とおき, この $L(\mathbf{u}, \lambda)$ を \mathbf{u} で偏微分して $\mathbf{0}$ と置くと

$$\mathbf{\Sigma} \mathbf{u} = \lambda \mathbf{u} \tag{5}$$

となり, \mathbf{u} を求める問題は共分散行列 $\mathbf{\Sigma}$ の固有値問題に帰着する. 行列 $\mathbf{\Sigma}$ は半正定値対称行列であるから, d 個の実数で非負の固有値 $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \geq 0$ を持つ. 各固有値に対応する固有ベクトルを $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d$ とすれば式 (5) より

$$\mathbf{\Sigma} \mathbf{u}_j = \lambda_j \mathbf{u}_j \tag{6}$$

となる. 両辺に左から \mathbf{u}_j^\top を掛けると制約条件 $\mathbf{u}^\top \mathbf{u} = 1$ から

$$\mathbf{u}_j^\top \mathbf{\Sigma} \mathbf{u}_j = \lambda_j \tag{7}$$

となり, λ_j はちょうど $z_i = \mathbf{u}_j^\top \mathbf{x}_i$ の分散に等しくなることがわかる. したがって, サンプル分散 $\mathbf{u}^\top \mathbf{\Sigma} \mathbf{u}$ を最大にする \mathbf{u} (長軸の方向) は, 最大の固有値 λ_1 に対応する固有ベクトル $\mathbf{u}_1 = (u_{11} \dots u_{1d})^\top$ で与えられる. 同様にして, サンプル分散が二番目に大きい方向ベクトルは λ_2 に対応する固有ベクトル \mathbf{u}_2

であるが、固有値分解で得られる固有ベクトルはノルムが1で互いに直交している正規直交ベクトルであるので、 u_2 が図2の例における短軸の方向である。一般に、固有値の大きな上位 $r < d$ 個の固有ベクトルを並べた $d \times r$ の行列を $U = (u_1 \cdots u_r)$ とすると、 u_j ($j = 1, \dots, r$) によって張られる r 次元アフィン部分空間にサンプル x_i を射影したときの座標 $z_i = (z_{i1} \cdots z_{ir})^T$ は

$$z_i = U^T(x_i - m) \quad (8)$$

によって与えられる (図3左)。これをもとの d 次元空間での座標で表すと

$$\tilde{x}_i = m + Uz_i = m + UU^T(x_i - m) \quad (9)$$

と書ける (図3右)。すなわち、 d 次元サンプル x_i は r 本の固有ベクトルの線型結合によってサンプル \tilde{x}_i に変換されていると見なすことができる。

以上をまとめると、 d 次元サンプルの分布を主成分分析を利用して2次元や3次元空間に射影して観察したい場合には、共分散行列 Σ を固有値分解し、固有値の大きい上位 r 本 (2か3) の固有ベクトルを求め、式 (8) で r 次元空間でのサンプルの座標値を計算すれば良い。一方、 r 次元空間でのサンプルを d 次元空間で表すには式 (9) を計算すれば良い。

1.3 MATLAB のプログラミングの基礎と起動

ここでは MATLAB を用いて2次元サンプルの分布をよく近似する直線を求めてみよう。準備として、下記の URL から2次元サンプル `sample2d.txt` をダウンロードせよ。

<https://web.tuat.ac.jp/~s-hotta/pca/>

次に、テキストエディタを起動し、以下のようなプログラムを書いてみよ。

```
clear all;,close all; % すべてのグローバル変数/ウィンドを消去
X=load("sample2d.txt"); % sample2d.txt の読み込み
[d,n]=size(X); % Xの次元dとサンプル数nを取得
%サンプルを2次元平面に青点でプロット
figure(1),clf,hold on,plot(X(1,:),X(2:,:),"bo");
axis square %図を正方形に整形
```

上記プログラムの一行目は、グローバル変数の消去であり、MATLAB のプログラムでは記述しておいた方が無難である。; は行末、% はコメントを表す。二行目は、ファイル名が `sample2d.txt` であるデータを `load` 関数を使って読み込み、値を `X` という配列 (行列) に格納せよ、という意味である。言語 C と異なり、変数の宣言 (型指定やサイズ指定を含む) が必要ないことに注意せよ (基本的に変数は `double`

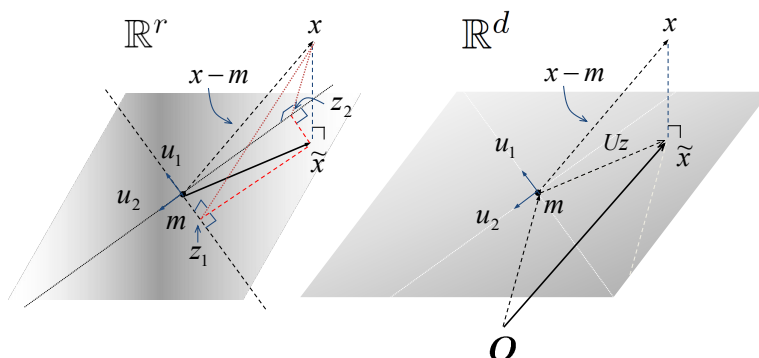


図3: r 次元と d 次元空間上での \tilde{x}

型で扱われる)。三行目は行列 X のサイズを取得する `size` 関数を用いて行数を d に、列数を n に格納している。行列 X は次元数 \times サンプル数であるので、 d が次元数、 n がサンプル数となる。図への表示は `figure` で始まる行であり、まず図を空 (`clf`) にして、上書きを許し (`hold on`)、サンプルを青点で `figure(1)` にプロットせよという命令であり、横軸の座標は X の 1 行目 $X(1,:)$ 、縦軸の座標は対応する X の 2 行目 $X(2,:)$ とせよ、という意味である。最終行は図を正方で表示するための命令である。プログラムを書き終えたら、ファイルを `step1.m` という名前で保存せよ。

次に MATLAB を起動しよう。MATLAB のアイコンをクリックして MATLAB を起動しよう。すると、図 4 のような画面が起動する。画面右側に MATLAB でプログラムを実行するためのキャラクタユーザインタフェース (CUI) が用意されており、プロンプトに続けてコマンド入力することでプログラムを動作させることができる。ここから `cd` により、プログラムを保存したフォルダに移動し、`ls` と入力して Enter キーを押せば、`step1.m` がリストアップされるはずである。なお MATLAB を終了するにはプロンプトに `exit` と入力して Enter キーを押せばよい。

次に、先程作成した `step1.m` を動作させてみよう。プログラムを動作させるには、拡張子 `.m` を除いたファイル名をプロンプトに続けて入力し Enter を押せばよい。

```
>step1
```

正しくプログラムが書かれていれば、`sample2d.txt` の中身が図にプロットされるはずである。

1.4 直線を引いてみよう

次に、画面中に直線 $y = 1/10x + 1/10$ を引いてみよう。先程作成した `step1.m` を以下のように変更して、上書き保存せよ。

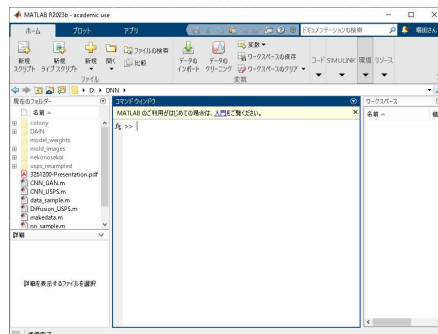


図 4: MATLAB の起動画面

```

clear all;,close all; % すべてのグローバル変数/ウィンドを消去
X=load("sample2d.txt"); % sample2d.txt の読み込み
[d,n]=size(X); % Xの次元dとサンプル数nを取得
% Xの横軸の最小値から最大値まで0.05刻みの値をxxに格納
xx=[min(X(1,:)).05:max(X(1,:))];
% サンプルを2次元平面に青点でプロット
figure(1),clf,hold on,plot(X(1,:),X(2:,:),"bo");
figure(1),plot(xx,0.1.*xx+0.1,"g-"); % 直線0.1x+0.1を緑の直線で描画
axis square

```

追加したもののうち、 $xx=[\min(X(1,:)).05:\max(X(1,:))]$; は、 X の1行目の最小値から最大値まで0.05刻みずつ値を増やしていったものを xx に格納するという意味である。すなわち、直線の描画範囲を xx に格納しているのである。この xx を用いて直線 $y = 1/10x + 1/10$ を緑の線で描画している。

1.5 レポート課題 1

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5のように図示するプログラムを作成せよ。図5における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。なお、固有値、固有ベクトルは eig 関数を用いて求めて良い（11 ページ以降の TIPS を参照せよ）。また、この固有ベクトルにサンプルを正射影して求まる z_i のサンプル分散が、最大固有値と同じ値となることを確かめよ。レポートにはプログラム、作成した図、および最大固有値を記載せよ。

ここまでの Step1 であり、以降の Step2 と Step3 は各自テキストを読みながら進めること。なお、11 ページ以降の TIPS は 1 通り試して MATLAB に慣れておくことを推奨する。

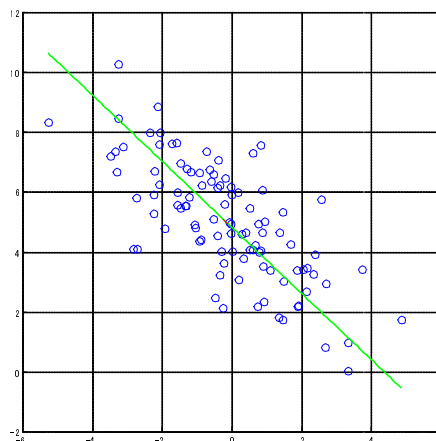


図 5: 課題 1 の結果

2 Step2: 高次元サンプルの分布を2次元空間で可視化してみよう

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よう。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 flower dataset 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, 図6 上段) を各種 50 個体選び、それぞれの花のがく片の長さや幅、花弁の長さや幅 (図6 下段) 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よう。iris flower dataset は MATLAB から標準データとして簡単に読み込むことができる：

```
x=iris_dataset;
```

x のサイズを調べれば 4 行 150 列となっていることがわかるはずである。すなわち、各列がそれぞれの花を表し、各行が 4 つの長さや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。

ところで、 d 次元の高次元サンプルを、主成分分析により r 次元空間 ($r < d$) に写像したとき、 r 次元空間のサンプル分布は元の分布のどれぐらいよい近似になっているだろうか？これは累積寄与率 (cumulative propotion) によって調べることができる。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

$$\sum_{i=1}^r \lambda_i / \sum_{j=1}^d \lambda_j \quad (10)$$

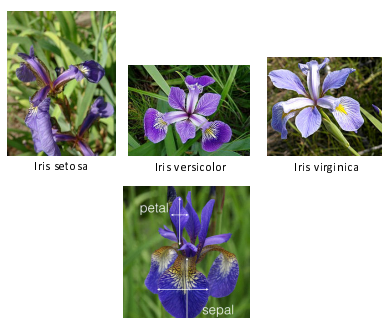


図 6: 上段 : 各アヤメ (Wikipedia より引用) . 下段 : 各花における計測部分 (Kaggle.com より引用)

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。

2.1 レポート課題 2

iris flower dataset に対して主成分分析を適用し, 固有値の大きい上位 2 本の固有ベクトルを用いて, サンプルの 2 次元散布図を作成せよ。このとき, 3 種類の花を別々の色でプロットすることで種類を区別できるようにせよ。また, 横軸を $r(r = 1, \dots, 4)$, 縦軸を累積寄与率としたグラフを作成せよ。レポートにはプログラム, 作成した散布図, グラフを記載せよ。

ヒント: あるベクトル c の特定の範囲の値を参照するには

```
c(3:7),c(1:100)
```

などとすればよい。 $c(3:7)$ は c の第 3 成分から第 7 成分までを参照し, $c(1:100)$ は c の第 1 成分から第 100 成分までを参照する。また, 色の異なる点をプロットする場合には

```
plot(x1,y1,"ro");→ 赤い丸  
plot(x2,y2,"go");→ 緑の丸  
plot(x3,y3,"bo");→ 青の丸
```

のようになる。

3 Step3: 高次元サンプルを少ない正規直交ベクトルの線型結合で表現しよう

1.2節の式(9)に示したように、主成分分析を用いれば、元のサンプルを近似ではあるが、少ない記憶容量で保存することができる。例えば iris flower dataset の場合、元の行列のサイズは 4×150 であるが、これを2次元に圧縮した場合には、 4×1 の平均ベクトル \mathbf{m} 、 4×1 の固有ベクトル2本 $\mathbf{u}_1, \mathbf{u}_2$ 、2次元空間上でのサンプル座標 $\mathbf{U}^\top(\mathbf{x}_i - \mathbf{m})$ を保存しておけば、元のサンプルをある程度復元できる。すなわち、記憶容量をおよそ半分に削減できる。このことを実験的に確認してみよう。

3.1 レポート課題 3

iris flower dataset 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(9)により $\tilde{\mathbf{x}}_i$ を求めよ。その後、元のサンプル \mathbf{x}_i との2乗誤差の平均

$$error_r = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2 = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^\top (\mathbf{x}_i - \tilde{\mathbf{x}}_i)$$

を計算せよ。レポートには横軸を r 、縦軸を $error_r$ としたグラフを記載せよ。

以上が必須課題であり、以降は挑戦課題(任意)である。ただし、成績には反映されるので、好成績を狙う場合は挑戦課題を実施することを推奨する。

4 挑戦課題 1: 主成分分析を実データへ適用してみよう

各自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論ぜよ。このとき

- どのようなデータであるか(どのサイトからダウンロードしたのか、自前で用意したものか)
- データの内訳の詳細
- 2次元空間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

について必ず記載せよ。

5 挑戦課題 2: 固有顔を求めてみよう

固有顔とは、顔画像の画素値を縦に並べたベクトル(図7左)を複数人分用意し、それらのベクトルを並べた行列を用いて主成分分析により求めた固有ベクトルのことである。その固有ベクトルを画像化すると図7右のように、顔の大域的な陰影から局所的な陰影を表すパターンが観察できる。

5.1 課題内容

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よ。ただし人間の顔でなくてもよい。このとき

- どのようなデータを使用したのか(どのサイトからダウンロードしたのか、自前で用意したものか)

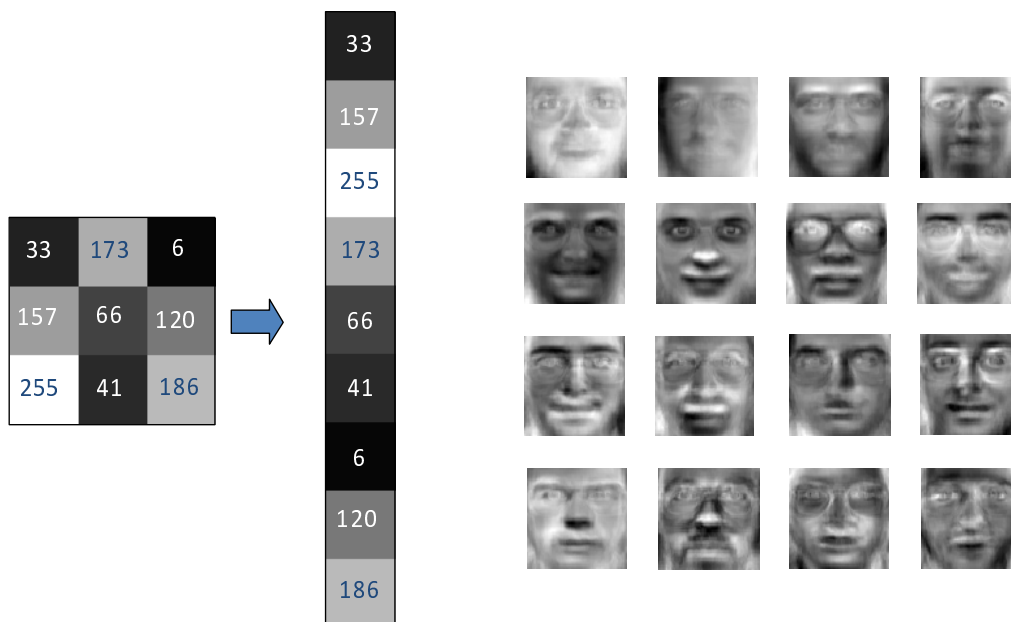


図 7: 左 : 画像のベクトル化. 右 : 固有顔の例.

- 平均ベクトルと, 最大の固有値から 16 番目に大きい固有値に対応する 16 本の固有ベクトルを 図 7 右のように画像化したもの
- この実験で使用したプログラム

について必ず記載せよ.

TIPS (行列 (2次元配列) とベクトル (1次元配列) の生成)

行列

$$A = \begin{pmatrix} 1 & 3 \\ 2 & 4 \end{pmatrix}$$

を MATLAB で作成するには

```
A=[1,3;2,4]
```

と書く. , が横への連結, ; が縦への連結である. ベクトル

$$a = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

を MATLAB で作成するには

```
a=[1;2]
```

と書く. つまり 2 つの数字を縦に連結する. サイズが同じ 2 つのベクトル

$$a = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, b = \begin{pmatrix} 3 \\ 4 \end{pmatrix}$$

を並べて行列 A を作成するには

```
A=[a,b]
```

となる. 行や列のサイズが同じであれば 2 つ以上のベクトルや行列を縦や横に連結することができる.

TIPS (行列とベクトルの値の参照)

- ベクトル a の第 1 成分を参照するには $a(1)$, 第 2 成分を参照するには $a(2)$ とする. 配列番号は 1 から始まる.
- 行列 A の i 行 j 列を参照するには $A(i,j)$ とする
- 行列 A の i 行目を参照するには $A(i,:)$ とする. 横ベクトル
- 行列 A の j 行目を参照するには $A(:,j)$ とする. 縦ベクトル

TIPS (値の表示)

MATLAB では `printf` を使わなくても, 変数名を入力するだけで中身を表示することができる. 例えば a の第 1 成分を表示させたいのであれば

```
a(1)
```

と入力すればよい.

TIPS (ベクトル・行列の転置)

ベクトルや行列の転置は良く行う操作である。MATLAB ではシングルクォーテーション'を用いる。例えば縦ベクトル $a=[1;2]$ の転置は $a'=[1,2]$ となる。

TIPS (ベクトル・行列の演算)

MATLAB や MATLAB の四則演算についてまとめておく。a および b は $d \times 1$ のベクトル、A および B は $d \times d$ の行列とする。サイズが適切でないベクトルや行列同士の計算はできないので注意。

- $a+b$ は各ベクトルの対応する成分の和をベクトルとして出力する
- $a-b$ は各ベクトルの対応する成分の差をベクトルとして出力する
- $a.*b$ は各ベクトルの対応する成分の積をベクトルとして出力する (.*はスカラー積)
- $a./b$ は a の成分を対応する b の成分で割ったベクトルを出力する (./はスカラー除算)
- $a'*b$ はベクトル a を転置したものとベクトル b の積 (内積) を出力する (*はベクトル積)
- $a*b'$ はベクトル a とベクトル b を転置したものととの積 (直積) を出力する (演算結果は $d \times d$ の行列)
- $A+B$ は各行列の対応する要素の和を行列で出力する
- $A-B$ は各行列の対応する要素の差を行列で出力する
- $A.*B$ は各行列の対応する要素の積を行列として出力する
- $A./B$ は A の要素を対応する B の要素で割った行列を出力する
- $A'*B$ は行列 A を転置したものと行列 B の積を計算する (演算結果は $d \times d$ の行列)
- $A*B'$ は行列 A と行列 B を転置したものととの積を計算する (演算結果は $d \times d$ の行列)
- $a=zeros(d,1)$ は成分がすべて 0 の $d \times 1$ のベクトルを a とする
- $A=zeros(d,d)$ は要素がすべて 0 の $d \times d$ の行列を A とする
- $a=ones(d,1)$ は成分がすべて 1 の $d \times 1$ のベクトルを a とする
- $A=ones(d,d)$ は要素がすべて 1 の $d \times d$ の行列を A とする
- $a=rand(d,1)$ は成分を 0 から 1 までの一様乱数とした $d \times 1$ のベクトルを a とする
- $A=rand(d,d)$ は要素を 0 から 1 までの一様乱数とした $d \times d$ の行列を A とする
- $A=eye(d,d)$ は $d \times d$ の単位行列を A とする
- $inv(A)$ は A が正則行列であれば A の逆行列を出力する
- $det(A)$ は A が正則行列であれば A の行列式を出力する
- $a.^2$ はベクトル a の成分の 2 乗をベクトルとして出力する

TIPS (便利な関数)

例としてベクトル

$$a = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

と行列

$$A = \begin{pmatrix} -1 & 2 \\ 3 & 4 \end{pmatrix}$$

が与えられているとする。以下は各関数の演算結果のみを示す。詳しくは MATLAB 上のプロンプトで

```
>help 関数名
```

で調べよ。

- `sum(a)=1`, `mean(a)=0.5`, `max(a)=2`, `min(a)=-1` : 総和, 平均, 最大値, 最小値
- `sum(A,1)=[2,6]`, `mean(A,1)=[1,3]` : 行方向の総和と平均, 横ベクトル
- `sum(A,2)=[1;7]`, `mean(A,2)=[0.5;3.5]` : 列方向の総和と平均, 縦ベクトル
- `max(A(:))=4`, `min(A(:))=-1` : 行列要素の最大値, 最小値
- `[V D]=eig(A)` : `D` は `A` の固有値が対角要素に格納された行列, `V` は各固有値に対応する固有ベクトルが縦に格納された行列
`D=[-2,0;0,5]`, `V=[-0.89443,-0.31623;0.44721,-0.94868]`
この場合, 固有値 `5` に対応する固有ベクトルは 2 列目の `V(:,2)=[-0.31623;-0.94868]`
- `diag(D)=[-2;5]` : `D` の対角要素を縦ベクトルで出力
- `[value index]=sort(a,"descend")` : `a` の成分を降順にソート。ソート後の値が `value` に, 対応する配列番号が `index` に格納される
`value=[2;-1]`, `index=[2;1]`
- `var(a,1)=4.5` : `a` の分散。第 2 引数の `1` はサンプル分散を指定するため
- `std(a)=2.1213` : `a` の標準偏差
- `abs(a)=[1;2]` : `a` の絶対値
- `norm(a)=2.2361` : `a` のノルム
- `cos(pi)=-1` : $\cos \pi$
- `acos(-1)=3.1416` : $\arccos(-1)$

TIPS (for 文の書き方)

```
for ii = 1 : n
    繰り返し実行したい命令
end
```

TIPS (if 文の書き方)

MATLAB では if 文を以下のように書く。二つの条件を同時に満たすように、すなわち and 条件を記述したい場合には

```
if 条件 A & 条件 B
    実行したい命令
end
```

のように条件を&を使って並べて書けば良い。一方、二つの条件のうち、どちらか片方が満たされれば良い場合、すなわち or 条件を記述したい場合には

```
if 条件 A | 条件 B
    実行したい命令
end
```

のように条件を|を使って並べて書けば良い。and や or を組み合わせる場合には、丸括弧() で優先順位を明示する必要があることは他のプログラミング言語と同じである。

TIPS (行列をベクトルに、ベクトルを行列に)

$d \times d$ の行列 A を $d^2 \times 1$ のベクトル a に変換するには

```
a=A(:)
```

とする。逆に $d^2 \times 1$ のベクトル a を $d \times d$ の行列 A に変換するには

```
A=reshape(a,[d d])
```

とする。

TIPS (ベクトルのコピーを並べて行列にする)

$d \times 1$ の縦ベクトル a を横に連続して n 本並べて $d \times n$ の行列を作りたい場合には

```
repmat(a,1,n)
```

とする。一方、 $1 \times d$ の横ベクトル a を縦に連続して n 本並べて $n \times d$ の行列を作りたい場合には

```
repmat(a,n,1)
```

とする。