

[問題] VIII-B

(1) VIII-A の問 3 にある 8 つのデータ点を最小 2 乗法を用いて 3 次多項式 $p_3(x)$ を当てはめる問題を考える。正規方程式を LU 分解で解くプログラムを作成し $p_3(x)$ の係数を有効数字 4 桁で 5 桁目を四捨五入して求めよ。作成したプログラムも提出すること。プログラミング言語は問わない。

(2) 問 1 の正規方程式をハウスホルダー法を用いて QR 分解してから解くプログラムを作成し $p_3(x)$ の係数を有効数字 4 桁で 5 桁目を四捨五入して求めよ。作成したプログラムも提出すること。プログラミング言語は問わない。

[略解] VIII-B

(1), (2) $p_3(x) = a + bx + cx^2 + dx^3$ とした時、

$$a = -2.827, \quad b = 13.33, \quad c = -4.514, \quad d = 1.055$$

※ この問題では条件数が小さく LU 分解で解いても QR 分解で解いても精度に差はない。

連立一次方程式の数値解の誤差

- ε を用いて連立一次方程式 $Ax = b$ の数値誤差を $(A + \varepsilon E)(x + \delta x) = (b + \varepsilon d)$ と表現する

▶ δx を見積もるには**条件数**が使える

- 行列 A の**条件数** (condition number)

▶ $\kappa(A) = \|A\|_2 \|A^{-1}\|_2$ ($\|A\|_2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$, $\|x\|_2 = \sqrt{\sum_{i=1}^m x_i^2}$)

- $\kappa(A)$ を用いた数値解の誤差 δx の評価式

▶ $\frac{\|\delta x\|_2}{\|x\|_2} \leq \kappa(A) \varepsilon \left(\frac{\|E\|_2}{\|A\|_2} + \frac{\|d\|_2}{\|b\|_2} \right)$

正規方程式の条件数

- 正規方程式 $(X^T X)a = X^T y$ に現れる係数行列 $X^T X$ の条件数が満たす関係式 $\kappa(X^T X) \leq \kappa(X)^2$
 - ▶ $\kappa(X^T X)$ は $\kappa(X)$ の2乗で大きくなる可能性がある
 - ▶ 一般に正規方程式をLU分解+前進・後退代入で解くと誤差が大きくなることが知られている
- 正規方程式の数値解法では旨くするための工夫が必要
 - ▶ **QR分解**により係数行列の条件数を小さくする方法
 - ▶ 特異値分解や固有値分解を行う手法
 - ▶ コレスキー法 ($X^T X$ が正定値対称であることを利用して $X = LL^T$ と変形)

QR分解—正規方程式の旨い解き方

QR分解 (QR factorization)

ある行列 X を直交行列 Q と上三角行列 R を用いて $X = QR$ の形に変形すること

- 正規方程式に出てくる正規行列 X ($\in \mathbb{R}^{m \times (n+1)}$) を直交行列 Q ($\in \mathbb{R}^{m \times (n+1)}$) と上三角行列 R ($\in \mathbb{R}^{(n+1) \times (n+1)}$) で表現する
 - ▶ 正規方程式は $Ra = Q^T y$ と書き換えることができ条件数は $\kappa(R)$ となる
- QR分解を実行する方法は複数ある
 - ▶ グラム・シュミットの直交化、ハウスホルダー変換

ハウスホルダー変換を用いたQR分解

ハウスホルダー変換 (Householder transformation)

ハウスホルダー行列 $H(\mathbf{u}) = I - \mathbf{u}\mathbf{u}^T$ ($\|\mathbf{u}\|_2 = \sqrt{2}$) を作用させることで任意のベクトルを \mathbf{u} を法線ベクトルに持つ平面に対して鏡像なベクトルに移す変換

- $H(\mathbf{u})$ は対称行列かつ直交行列 ($H(\mathbf{u})^2 = I$)
 - ▶ 鏡像変換であるため2回作用させると元の像に戻ることを意味する
- 行列 $X = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_n]$ の各列ベクトルに対してうまくハウスホルダー行列を選んで作用させれば必要な成分以外を0にできる
 - ▶ 元の行列を直交行列と下三角行列の積 (**QR 分解**) や直交行列と上ヘッセンベルグ行列の積の形に書き換えることができる

[手法] 最小2乗法の準備— X の構成

Preparation for least squares approximation—construction of normal matrix

Input: x_i ($0 \leq i \leq m-1$)

Output: X ($\in \mathbb{R}^{m \times (n+1)}$)

1: **for** $j = 0, 1, \dots, m-1$:

2: $X_{i0} = 1$

3: **for** $i = 0, 1, \dots, m-1$:

4: **for** $j = 1, \dots, n$:

5: $X_{ij} = x_i^j$

[手法] ハウスホルダー変換を用いた QR 分解

QR factorization using Householder transformation

Input: X_{ij} ($0 \leq i \leq m-1, 0 \leq j \leq n$)

Output: X_{ij} ($0 \leq i \leq n, 0 \leq j \leq n$), [上三角行列 R] y_i ($0 \leq i \leq n$) [$Q^T y$]

```
1: for  $j = 0, 1, \dots, n$  :  
2:     for  $i = 0, 1, \dots, j-1$  :  
3:          $u_i \leftarrow 0$   
4:      $t \leftarrow \sqrt{\sum_{i=j}^{m-1} X_{ij}^2}$ ;  $d \leftarrow \text{sign}(X_{jj}) \times t$ ;  $g \leftarrow \sqrt{t(t + |X_{jj}|)}$ ;  $u_j \leftarrow (X_{jj} + d)/g$ ;  $X_{jj} \leftarrow -d$   
5:     for  $i = j+1, j+2, \dots, m-1$  :  
6:          $u_i \leftarrow X_{ij}/g$ ;  $X_{ij} \leftarrow 0$   
7:     for  $k = j+1, j+2, \dots, n$  :  
8:          $sum \leftarrow \sum_{\ell=j}^{m-1} u_\ell X_{\ell k}$   
9:         for  $i = j, j+1, \dots, m-1$  :  
10:             $T_{ik} \leftarrow X_{ik} - sum \times u_i$   
11:     for  $i = j, j+1, \dots, m-1$  :  
12:         for  $k = j+1, \dots, n$  :  
13:             $X_{ik} \leftarrow T_{ik}$   
14:      $sum \leftarrow \sum_{k=j}^{m-1} u_k y_k$   
15:     for  $i = j, j+1, \dots, m-1$  :  
16:          $y_i \leftarrow y_i - sum \times u_i$ 
```


[手法] QR 分解後の正規方程式 $Ra = Q^T y$ を後退代入で解く

Backward substitution of normal equation $Ra = Q^T y$

Input: X_{ij} ($0 \leq i \leq n, 0 \leq j \leq n$), y_i ($0 \leq i \leq n$)

[上三角行列 R が X_{ij} ($0 \leq i \leq n, 0 \leq j \leq n$) として与えられ、 $Q^T y$ は y_i ($0 \leq i \leq n$) として与えられる]

Output: a_i ($0 \leq i \leq n$)

- 1: $a_n \leftarrow y_n / X_{nn}$
- 2: **for** $i = n - 1, n - 3, \dots, 0$:
- 3: $a_i \leftarrow \left(y_i - \sum_{j=i+1}^n X_{ij} a_j \right) / X_{ii}$



[数値解析 第9回]

固有値問題と特異値分解 (1)

線形変換を特徴付ける特別なベクトルと値を探す

対角行列と直交行列を用いた行列の変換

- 固有値問題 (Eigenvalue problem)
 - ▶ 行列の対角化 (Diagonalization) と呼ばれる
 - ▶ 物理の問題でよく現れる
 - ★ 慣性主軸を求める問題 (力学)
 - ★ 固有振動数と固有振動モードを求める問題 (構造力学)
 - ★ 量子系においてハミルトニアンから固有状態と固有エネルギーを求める問題 (量子力学・量子化学)
- 特異値分解 (Singular value decomposition)
 - ▶ 特殊なケースで固有値問題と一致する
 - ▶ データ解析に使える
 - ★ 最小2乗法・主成分分析

固有値問題の問題設定

固有値問題 (Eigenvalue problem)

行列 $A \in \mathbb{R}^{n \times n}$ に対して $Ax = \lambda x$ を満たす非零の固有値 λ と固有ベクトル x の組 (固有対) を求める問題

- 固有対が n 組ある場合、固有ベクトル x_1, x_2, \dots, x_n を互いに直交するようにとって正規化すると $X = [x_1, x_2, \dots, x_n]$ は直交行列となる
- 固有値 $\lambda_1, \lambda_2, \dots, \lambda_n$ を対角に並べた対角行列 Σ を用いると A は $AX = X\Sigma \Leftrightarrow A = X\Sigma X^T$ と表せる

特異値分解の問題設定

特異値分解 (Singular value decomposition)

行列 $B \in \mathbb{R}^{m \times n}$ に対し2つの直交行列 $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ および特異値を対角に並べた $\Sigma \in \mathbb{R}^{m \times n}$ を用いて $B = U\Sigma V^T$ と変形する問題

- $B^T B = V\Sigma^T \Sigma V^T$ とすると正方行列 B に対する固有値問題と同じ形になる

[手法] ベキ乗法 (全体像)

ベキ乗法 (Power method)

Input: $A, \mathbf{u} \in \mathbb{R}^n, k_{\max}, \varepsilon$ [u_i ($0 \leq i \leq n-1$) の初期値は $\mathbf{u} \neq \mathbf{0}$ となるように適当に選ぶ]

Output: λ, \mathbf{y}

```
1: for  $k = 0, 1, \dots, k_{\max}$  :  
2:    $\mathbf{y} \leftarrow \mathbf{u} / \|\mathbf{u}\|$   
3:    $\mathbf{u} \leftarrow A\mathbf{y}$   
4:    $\lambda \leftarrow (\mathbf{y}, \mathbf{u})$   
5:   if  $\|\mathbf{u} - \lambda\mathbf{y}\| \leq \varepsilon$  :  
6:     break
```

※ 上記アルゴリズムのうち *ℓℓ.* 2-5 の詳細を次ページ以降で示している

[手法] ベクトルの正規化 (p. 14 ℓ. 2; $y \leftarrow u/\|u\|$)

Normalization of u (p. 14 ℓ. 2; $y \leftarrow u/\|u\|$)

Input: u (u_i ($0 \leq i \leq n-1$))

Output: y (y_i ($0 \leq i \leq n-1$))

1: $norm \leftarrow 0$

2: **for** $i = 0, 1, \dots, n-1$:

3: $norm \leftarrow norm + u_i^2$

4: $norm \leftarrow \sqrt{norm}$

5: **for** $i = 0, 1, \dots, n-1$:

6: $y_i \leftarrow u_i/norm$

[手法] 行列演算 (p. 14 ℓ. 3; $u \leftarrow Ay$)

Matrix operation (p. 14 ℓ. 3; $u \leftarrow Ay$)

Input: y (y_i ($0 \leq i \leq n-1$))

Output: A, u (u_i ($0 \leq i \leq n-1$))

1: $sum \leftarrow 0$

2: **for** $i = 0, 1, \dots, n-1$:

3: **for** $j = 0, 1, \dots, n-1$:

4: $sum \leftarrow sum + A_{ij}y_j^{(k)}$

5: $u_i \leftarrow sum$

[手法] ベクトルの内積 (p. 14 ℓ. 4; $\lambda \leftarrow (y, u)$)

Inner product of vectors (p. 14 ℓ. 4; $\lambda \leftarrow (y, u)$)

Input: y (y_i ($0 \leq i \leq n-1$)), u (u_i ($0 \leq i \leq n-1$))

Output: λ

1: $\lambda \leftarrow 0$

2: **for** $i = 0, 1, \dots, n-1$:

3: $\lambda \leftarrow \lambda + y_i u_i$

[手法] ベクトルの長さの計算 (p. 14 ℓ. 5)

Calculation of vector norm $\|u - \lambda y\|$ (p. 14 ℓ. 5; in the condition of if statement)

Input: y (y_i ($0 \leq i \leq n-1$)), u (u_i ($0 \leq i \leq n-1$))

Output: $norm$ [$\|u - \lambda y\|$ の値を $norm$ として出力]

1: $norm \leftarrow 0$

2: **for** $i = 0, 1, \dots, n-1$:

3: $norm \leftarrow norm + (u_i - \lambda y_i)^2$

4: $norm \leftarrow \sqrt{norm}$

[問題] IX-A

VIII-A の問3にある8つのデータ点を用いて 2×2 の分散・共分散行列を構成し、ベキ乗法を用いて絶対値が最大の固有値とそれに対応する固有ベクトルを求めるプログラムを作成せよ。また分散・共分散行列が対称であり固有ベクトル同士が互いに直交する性質を利用してもう一つの固有値と固有ベクトルを求めよ。固有ベクトルは長さを1に正規化し、値は有効数字4桁で5桁目を四捨五入して求めよ。 ※ 固有ベクトルを決める際に残る向きはどちらの方向を選んでも構わない。また分散共分散行列を求める際に不変分散を

[略解] IX-A

最大固有値 2573 (不変分散・不変共分散を用いた場合¹)

対応する固有ベクトル $[0.03835, 0.9993]$

$[-0.03835, -0.9993]$ も正しい

もう一つの固有値 0.5025 (不変分散・不変共分散を用いた場合²)

対応する固有ベクトル $[-0.9993, 0.03835]$

$[0.9993, -0.03835]$ も正しい

¹ データ点数で割る定義の分散・共分散を用いた場合の値は 2251

² データ点数で割る定義の分散・共分散を用いた場合の値は 0.4397

[問題] IX-B

表 1 (次ページ) にある 8 匹のポケモンのステータスのデータから 3×3 の相関行列を構成し、ベキ乗法を用いて絶対値が最大の固有値 λ_1 とそれに対応する固有ベクトル y_1 を求めるプログラムを作成せよ。固有ベクトルは長さを 1 に正規化し、値は有効数字 4 桁で 5 桁目を四捨五入して求めよ。

※ 固有ベクトルを決める際に残る向きはどちらの方向を選んでも構わない。

[問題] IX-B 表 1

ポケモン	HP	攻撃	防御
ピカチュウ	111	112	96
ライチュウ	155	193	151
イーブイ	146	104	114
コイキング	85	29	85
ギャラドス	216	237	186
カビゴン	330	190	169
ミュウ	225	210	210
ミュウツー	214	300	182

Table: 「ポケモン GO」に出てくる 8 匹のポケモンのステータス。データは以下のサイトを参照した (<https://pokemongo.gamewith.jp/article/show/35945>)。

修正前のコード (1)

※ 2020/12/3 に配信した授業で説明した QR 分解の遅いアルゴリズムを次ページ以降に残しておく。

※ 「 Q の構成」と書かれたアルゴリズムで行列の形の Q を求めてしまうと計算量のオーダーが上がってしまう。 Q を構成するハウスホルダー行列の性質を生かしてベクトル同士の演算で済ませるのがポイント。

[手法] ハウスホルダー変換を用いた QR 分解— R の構成

QR factorization using Householder transformation—construction of R

Input: X ($\in \mathbb{R}^{m \times (n+1)}$)

Output: U, X ($\in \mathbb{R}^{m \times (n+1)}$) [上三角行列 R が X として得られる]

```
1: for  $j = 0, 1, \dots, n$  :  
2:    $t \leftarrow \sqrt{\sum_{i=j}^{m-1} X_{ij}^2}$ ;  $d \leftarrow \text{sign}(X_{jj}) \times t$ ;  $g \leftarrow \sqrt{t(t + |X_{jj}|)}$   
3:    $U_{jj} \leftarrow (X_{jj} + d)/g$ ;  $X_{jj} \leftarrow -d$   
4:   for  $i = j + 1, j + 2, \dots, m - 1$  :  
5:      $U_{ij} \leftarrow X_{ij}/g$ ;  $X_{ij} \leftarrow 0$   
6:   for  $i = j, j + 1, \dots, m - 1$  :  
7:     for  $k = j + 1, j + 2, \dots, n$  :  
8:        $T_{ik} \leftarrow X_{ik} - \left(\sum_{\ell=j}^{m-1} U_{\ell j} X_{\ell k}\right) U_{ij}$   
9:   for  $i = j, j + 1, \dots, m - 1$  :  
10:    for  $k = j + 1, \dots, n$  :  
11:       $X_{ik} \leftarrow T_{ik}$ 
```


[手法] ハウスホルダー変換を用いた QR 分解— Q の構成

QR factorization using Householder transformation—construction of Q

Input: U ($\in \mathbb{R}^{m \times (n+1)}$)

Output: Q ($\in \mathbb{R}^{m \times m}$)

1: **for** $\ell = 0, 1, \dots, n$:

2: **for** $i = 0, 1, \dots, m - 1$:

3: **for** $j = 0, 1, \dots, m - 1$:

4: $T_{ij} \leftarrow Q_{ij} - U_{i\ell} \left(\sum_{k=0}^{m-1} U_{k\ell} Q_{kj} \right)$

5: **for** $i = 0, 1, \dots, m - 1$:

6: **for** $j = 0, 1, \dots, m - 1$:

7: $Q_{ij} \leftarrow T_{ij}$

[手法] QR 分解後の正規方程式 $Ra = Q^T y$ を後退代入で解く

Backward substitution of normal equation $Ra = Q^T y$

Input: X, Q ($\in \mathbb{R}^{m \times (n+1)}$), y_i ($0 \leq i \leq m$)

Output: a_i ($0 \leq i \leq n$)

- 1: **for** $j = 0, 1, \dots, n$:
- 2: $z_j \leftarrow \sum_{k=0}^{m-1} Q_{jk} y_k$
- 3: $a_n \leftarrow z_n / X_{nn}$
- 4: **for** $i = n-1, n-3, \dots, 0$:
- 5: $a_i \leftarrow \left(z_i - \sum_{j=i+1}^n X_{ij} a_j \right) / X_{ii}$

修正前のコード (2)

※ 2020/12/3 に配信した授業で説明したベキ乗法のアルゴリズムを次ページ以降に残しておく。修正後ではベクトルや行列で表記した部分を for 文を含むアルゴリズムとして書き換えている。

[手法] ベキ乗法

ベキ乗法 (Power method)

Input: $A, \mathbf{u}_0, k_{\max}, \varepsilon$

Output: λ, \mathbf{x}

```
1: for  $k = 0, 1, \dots, k_{\max}$  :  
2:    $\mathbf{y}_k \leftarrow \mathbf{u}_k / \|\mathbf{u}_k\|_2$   
3:    $\mathbf{u}_{k+1} \leftarrow A\mathbf{y}_k$   
4:    $\lambda_k \leftarrow (\mathbf{y}_k, \mathbf{u}_{k+1})$   
5:   if  $\|\mathbf{u}_{k+1} - \lambda_k \mathbf{y}_k\| \leq \varepsilon$  :  
6:     break  
7:  $\lambda \leftarrow \lambda_k$   
8:  $\mathbf{x} \leftarrow \mathbf{y}_k$ 
```