

# MMAD - Úkol 1

Filip Ditrich

Unicorn University, Prague, Czech Republic  
27. dubna 2024

✓ Úloha 1	1
✓ Úloha 2	2
✓ Úloha 3	3
✓ Úloha 4	4
✓ Úloha 5	5
✓ Úloha 6	7

## ✓ Úloha 1

(3 body)

Dokažte, že každá ortogonální matice má determinant  $\pm 1$ .

### Řešení

Ortogonální matice je čtvercová reálná matice  $Q$ , která splňuje podmínku:

$$Q^{TQ} = QQ^T = I$$

kde  $I$  je jednotková matice.

Dále platí, že determinant součinu dvou matic je roven součinu determinantů těchto matic, tedy:

$$\det(Q^{TQ}) = \det(Q) \cdot \det(Q^T)$$

Determinant jednotkové matice je vždy 1:

$$\det(I) = 1$$

Z předchozích vztahů plyne, že:

$$\det(Q^{TQ}) = \det(I) = 1$$

Dá se také zapsat jako:

$$\det(Q) \cdot \det(Q) = \det(Q)^2 = 1$$

Odtud již plyne, že determinant ortogonální matice je roven  $\pm 1$ , jelikož:

$$\sqrt{\det(Q)^2} = \sqrt{1} = \pm 1$$

### ODPOVĚĎ

Ano, každá ortogonální matice má determinant  $\pm 1$ .

## ✓ Úloha 2

(3 body)

Pokud mají matice  $A$  i  $B$  stejná vlastní čísla, platí i, že  $A \times B$  mají ty samá vlastní čísla?

### Řešení

Neplatí. Uvažme matice

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$$

Obě matice mají stejná vlastní čísla:

$$\det(A - \lambda I) = 0 \Rightarrow (1 - \lambda)(2 - \lambda) = 0 \Rightarrow \lambda_1 = 1, \lambda_2 = 2$$

$$\det(B - \lambda I) = 0 \Rightarrow (2 - \lambda)(1 - \lambda) = 0 \Rightarrow \lambda_1 = 1, \lambda_2 = 2$$

ale jejich součin:

$$A \times B = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$$

již vlastní čísla stejná nemají:

$$\det(A \times B - \lambda I) = \det \begin{pmatrix} 2 - \lambda & 0 \\ 0 & 2 - \lambda \end{pmatrix} = (2 - \lambda)^2 = 0 \Rightarrow \lambda = 2$$

Kdyby však matice  $A$  a  $B$  byly navzájem komutativní, tedy  $A \times B = B \times A$ , pak by platilo, že  $A \times B$  mají stejná vlastní čísla jako  $B \times A$ , protože vlastní čísla matice jsou nezávislá na pořadí násobení:

$$\det(A \times B - \lambda I) = \det(B \times A - \lambda I)$$

### ODPOVĚĎ

Ne, pokud matice  $A$  a  $B$  nemají stejná vlastní čísla, pak ani jejich součin  $A \times B$  nemusí mít stejná vlastní čísla.

### ✓ Úloha 3

(3 body)

Rozhodněte, zda je následující matice diagonalizovatelná

$$A = \begin{pmatrix} 0 & 1 & 2 \\ -4 & 1 & 4 \\ -5 & 1 & 7 \end{pmatrix}$$

#### Řešení

Diagonalizovatelná matice je taková reálná čtvercová matice o rozměrech  $n \times n$ , která má  $n$  navzájem různých vlastních čísel.

Vlastní čísla matice  $A$  jsou řešením rovnice  $\det(A - \lambda I) = 0$ :

$$\det(A - \lambda I) = \begin{vmatrix} -\lambda & 1 & 2 \\ -4 & 1 - \lambda & 4 \\ -5 & 1 & 7 - \lambda \end{vmatrix} = 0$$

Po úpravě dostaneme charakteristický polynom:

$$p_A(\lambda) = -\lambda^3 + 8\lambda^2 - 17\lambda + 10 = 0$$

Tuto rovnici následovně upravíme:

$$\begin{aligned} -\lambda^3 + 8\lambda^2 - 17\lambda + 10 &= 0 \\ &\dots \\ (\lambda - 1) \times (\lambda - 2) \times (\lambda - 5) &= 0 \end{aligned}$$

A tím získáme kořeny charakteristické rovnice a tedy vlastní čísla matice  $A$ :

$$\begin{aligned} \lambda_1 &= 1 \\ \lambda_2 &= 2 \\ \lambda_3 &= 5 \end{aligned}$$

Jelikož  $\lambda_1 \neq \lambda_2 \neq \lambda_3$ , matice  $A$  má 3 různá vlastní čísla a tedy je diagonalizovatelná.

#### ODPOVĚĎ

Ano, matice  $A$  je diagonalizovatelná.

## ✓ Úloha 4

(3 body)

Nalezněte vlastní vektor příslušný k nejmenšímu a největšímu vlastnímu číslu matice  $A$  z předchozího příkladu.

### Řešení

Vlastní vektory matice  $A$  jsou řešením soustavy rovnic  $(A - \lambda I)x = 0$ , kde  $\lambda$  je vlastní číslo matice  $A$ .

Nejmenší a největší vlastní čísla matice  $A$  jsou  $\lambda_{\min} = 1$  a  $\lambda_{\max} = 5$ .

Pro  $\lambda_{\min} = 1$ :

$$(A - 1I)x = \begin{pmatrix} -1 & 1 & 2 \\ -4 & 0 & 4 \\ -5 & 1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Po úpravě dostaneme:

$$\begin{aligned} x_1 - x_3 &= 0 \\ x_2 + x_3 &= 0 \end{aligned}$$

Řešením této soustavy rovnic je vlastní vektor  $p$ :

$$p = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix}$$

Pro  $\lambda_{\max} = 5$ :

$$(A - 5I)x = \begin{pmatrix} -5 & 1 & 2 \\ -4 & -4 & 4 \\ -5 & 1 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Po úpravě dostaneme:

$$\begin{aligned} x_1 - \frac{1}{2}x_3 &= 0 \\ x_2 - \frac{1}{2}x_3 &= 0 \end{aligned}$$

Řešením této soustavy rovnic je vlastní vektor  $s$ :

$$s = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix}$$

### ODPOVĚĎ

Vlastní vektor příslušný nejmenšímu vlastnímu číslu  $\lambda_{\min} = 1$  je  $p = (1, -1, 1)^T$  a vlastní vektor příslušný největšímu vlastnímu číslu  $\lambda_{\max} = 5$  je  $s = (\frac{1}{2}, \frac{1}{2}, 1)^T$ .

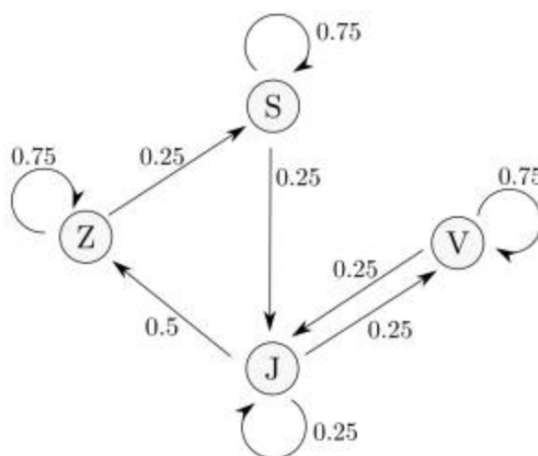
## ✓ Úloha 5

(3 body)

Mějme stálý migrační proces obyvatel mezi Severem, Jihem, Východem a Západem, jehož průběh za 1 rok lze znázornit diagramem níže (desetinná čísla udávají, jaký zlomek populace se za rok přemístí po šípce do jiné oblasti). Na počátku je počet obyvatel ve všech oblastech 10 milionů.

- 5a) Jak bude situace rozložení populace vypadat za 10 let od počátku?
- 5b) Na kterých hodnotách se populace v jednotlivých oblastech ustálí, pokud ji budeme dostatečně dlouho sledovat?

Vyžijte libovolný počítačový nástroj a své argumenty podpořte výstupy tohoto nástroje.



Obrázek (1) – Diagram stálého migračního procesu obyvatel mezi Severem, Jihem, Východem a Západem.

### Řešení

Pro řešení této úlohy využijeme jazyk Python a knihovnu `numpy` pro práci s maticemi a vektory.

#### 5a) Jak bude situace rozložení populace vypadat za 10 let od počátku?

Nejprve si vytvoříme matici přechodů  $P$  a vektor počátečního rozložení populace  $x_0$ :

```
1 import numpy as np
2
3 # matice přechodů
4 P = np.array([
5     # Sever -> S, J, V, Z
6     [0.75, 0.25, 0, 0],
7     # Jih -> S, J, V, Z
8     [0, 0.25, 0.25, 0.5],
9     # Východ -> S, J, V, Z
10    [0, 0.25, 0.75, 0],
11    # Západ -> S, J, V, Z
12    [0.25, 0, 0, 0.75]
13 ])
14
15 # počáteční rozložení populace
16 x0 = np.array([10, 10, 10, 10])
```

Následně provedeme výpočet rozložení populace za 10 let:

```
1 # výpočet rozložení populace za 10 let
2 x = x0
3 for _ in range(10):
4     x = np.dot(P, x) # P * x
5 print(x)
6 # [13.05176735  6.66666985  6.94823265 13.33333015]
```

#### ODPOVĚĎ 5A

Výsledné rozložení populace po 10 letech bude:

$$S \approx 13.05 \text{ mil.}; J \approx 6.67 \text{ mil.}; V \approx 6.95 \text{ mil.}; Z \approx 13.33 \text{ mil.}$$

**5b) Na kterých hodnotách se populace v jednotlivých oblastech ustálí, pokud ji budeme dostatečně dlouho sledovat?**

Pro řešení této otázky můžeme využít vlastních vektorů a vlastních čísel matice přechodů  $P$ . Vlastní vektor příslušný vlastnímu číslu 1 nám řekne, na kterých hodnotách se populace ustálí po dostatečně dlouhé době (tj. po nekonečně mnoho letech).

```
1 # vlastní čísla a vektory matice P
2 w, v = np.linalg.eig(P)
3 # vlastní čísla: [0.25 0.5 1. 0.75]
4 # vlastní vektory: [
5 #   [ 3.16227766e-01 -6.32455532e-01 -6.32455532e-01  7.07106781e-01]
6 #   [ 6.32455532e-01 -3.16227766e-01 -3.16227766e-01  4.44605073e-17]
7 #   [-3.16227766e-01  3.16227766e-01 -3.16227766e-01 -7.07106781e-01]
8 #   [-6.32455532e-01  6.32455532e-01 -6.32455532e-01  2.03628341e-15]
9 # ]
10 print(w)
11 print(v)
12
13 # vlastní vektor příslušný vlastnímu číslu 1
14 v1 = v[:, np.isclose(w, 1)]
15 # [[-0.63245553], [-0.31622777], [-0.31622777], [-0.63245553]]
16
17 # normalizace vlastního vektoru
18 v1 = v1 / np.sum(v1)
19 # [[0.33333333] [0.16666667], [0.16666667], [0.33333333]]
```

#### ODPOVĚĎ 5B

Pokud budeme populaci dostatečně dlouho sledovat, ustálí se na hodnotách:

$$S \approx 33.33\%; J \approx 16.67\%; V \approx 16.67\%; Z \approx 33.33\%$$

Úplný zdrojový kód je k nalezení v souboru `ukol-1-5.py`.

## ✓ Úloha 6

(3 body)

Na adrese <https://openmv.net/file/room-temperature.csv> naleznete soubor s daty. Jedná se o simulovaná data měření teploty v místnosti. Data jsou v .csv formátu. Obsahují hlavičku a pět sloupců. V prvním sloupci je datum, v následujících čtyřech sloupcích jsou teploty měření v jednotlivých čtyřech rozích místnosti.

Celkem je v souboru 144 záznamů měření. Pokud bychom chtěli s takovými daty dále pracovat, kolik hlavních komponent (a jaké) je vhodné zvolit a proč? Využijte SVD rozkladu a libovolného počítačového nástroje a své argumenty podpořte výstupy těchto nástrojů. Nezapomeňte na úvodní transformaci dat.

### Řešení

Pro vyřešení této úlohy využijeme jazyk Python a knihovny `pandas` a `numpy` pro práci s maticemi a vektory. Nejprve načteme data z CSV souboru a provedeme úvodní transformaci dat.

```
1 import pandas as pd
2 import numpy as np
3
4 # načtení dat
5 data = pd.read_csv('room-temperature.csv')
6 # zahození sloupce s datem (nepotřebné pro analýzu)
7 data = data.drop(columns=['Date'])
8 # normalizace dat (odstranění průměru)
9 data = data - data.mean()
```

Následně provedeme SVD rozklad matice dat a zjistíme, kolik hlavních komponent je vhodné zvolit.

```
1 # SVD rozklad
2 U, s, V = np.linalg.svd(data, full_matrices=False)
3 # s = [34.54408633 16.15976563 7.69262365 6.51293724]
4
5 # výpočet vysvětlené variance
6 explained_variance = np.square(s) / np.sum(np.square(s))
7 # explained_variance = [0.76688522 0.16782361 0.03803049 0.02726068]
8
9 # zjištění počtu komponent pro 95% vysvětlené variance
10 components = np.argmax(np.cumsum(explained_variance) > 0.95) + 1
11 # components = 3
```

### ODPOVĚĎ

Z výsledků SVD rozkladu zjistíme, že pro dosažení 95% vysvětlené variance je vhodné zvolit 3 hlavní komponenty.

Úplný zdrojový kód je k nalezení v souboru `ukol-1-6.py`.