Matematyczne aspekty analizy danych
semestr zimowy 2024/2025

Dr Anna Muranova
UWM w Olsztynie

Wykład 7

Rozkład Studenta (rozkład t Studenta, rozkład t)

Rozkład o gęstości:

$$, f(t) \;=\; \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi\ \nu}\ \Gamma\left(\frac{\nu}{2}\right)} \left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2} \quad ,$$

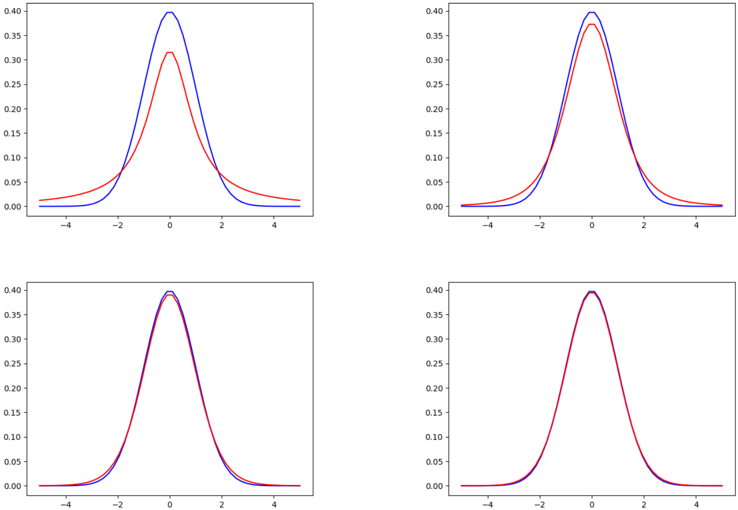
gdzie $\nu = n - 1$ – ilość stopni swobody, n – ilość próby. Używa się dla prób $n \leq 30$ zamiast rozkładu normalnego.

Srednia: 0

Wariacja: $\dfrac{\nu}{\nu - 2}$

Im mniejszy rozmiar próby, tym grubsze „ogony"w rozkładzie 1. Interesujące jest to, że kiedy zbliżamy się do liczby 31 elementów, rozkład T staje się praktycznie nieodróżnialny od rozkładu normalnego, co dobrze odzwierciedla idee stojące za centralnym twierdzeniem granicznym.

Rozkład Studenta: $t = 2, 5, 15, 35$



Kod

```
from math import *
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(-5,5)
def f(x, mu = 0, sigma2 = 1):
    sigma = sqrt(sigma2)
    return 1/(sigma*sqrt(2*pi))*exp(-(x-mu)**2/(2*sigma**2))

def f1(x, nu):
    return gamma((nu+1)/2)/(gamma(nu/2)*sqrt(nu*pi))*(1+x**2/nu)
                                                  **(-(nu+1)/2)

n = 15 #2,5,35
nu0 = n-1

y1 = np.array([f(i) for i in x])
y2 = np.array([f1(i, nu0) for i in x])
plt.plot(x,y1, color = 'b')
plt.plot(x, y2, color = 'r')
plt.show()
```

Rozdział 5. Algebra liniowa

Macierz

Macierz – to układ liczb, symboli lub wyrażeń zapisanych w postaci prostokątnej tablicy

$$\begin{array}{c c} & \begin{array}{cccc} 1 & 2 & \dots & n \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ \vdots \\ m \end{array} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \end{array}$$

Źródło: Autorstwa Svjo - Praca własna, CC BY-SA 4.0, `https://commons.wikimedia.org/w/index.php?curid=79728977`

Macierzy: działania

$A = (a_{ij}), B = (b_{ij})$

- Dodawanie: $C = A + B$, $c_{ij} = a_{ij} + b_{ij}$
- Mnożenie na liczbą: $C = cA$, $c_{ij} = c \cdot a_{ij}$
- Mnożenie: $C = A \cdot B$, $c_{ij} = \sum_k a_{ik} b_{kj}$
- Transponowanie: $C = A^T, c_{ij} = a_{ji}$

Macierzy specjalne

- *Macierz kwadratowa* – ma równą liczbą wierze i kolumn.
- *Macierz jednostkowa (tożsamościowa)* – ma jedynki na przekątnej, zero w innych miejscach `eye`.
- *Macierz zerowa (tożsamościowa)* – ma sami zera `zeros`.
- *Macierz diagonalna (przekątna)* – ma niezerowe elementy tylko na przekątnej `diag`.
- *Macierz trójkątna* – ma niezerowe elementy tylko na przekątnej i wyżej.
- *Macierz rzadka* – składa się głownie z zer.
- *Macierz symetryczna* – $a_{ij} = a_{ji}$.

Uwaga: listy jako argumenty

```
import numpy as np

print(np.eye(3))
print(np.zeros(3))#[0. 0. 0.]
print(np.zeros([3]))#[0. 0. 0.]
print(np.zeros([3,1]))#kolumna
print(np.zeros([1,3]))#wiersz
print(np.zeros([3,3]))
print(np.ones(3))#[1. 1. 1.]
print(np.ones([3]))
print(np.ones([3,1]))
print(np.ones([3,3]))
print(np.diag([4,0,7,5]))
```

Macierzy specjalne: sympy

Uwaga: NIE listy jako argumenty

```
import sympy as sp

print(sp.eye(3))
print(sp.zeros(3))#[0. 0. 0.]
print(sp.zeros(3,1))#kolumna
print(sp.zeros(1,3))#wiersz
print(sp.zeros(3,3))
print(sp.ones(3))#[1. 1. 1.]
print(sp.ones(3))
print(sp.ones(3,1))
print(sp.ones(3,3))
print(sp.diag(4,0,7,5))
```

Macierzy: wyznacznik

Niech $A \in M_{n\times n}(\mathbb{R})$ jest macierzą.

- Definicja permutacyjna

  Wówczas:

$$\det A = \sum_{\sigma\in S_n} (-1)^{\mathrm{Inv}(\sigma)}\ a_{1\sigma(1)} \cdot a_{2\sigma(2)} \cdot \ldots \cdot a_{n\sigma(n)},$$

  gdzie S_n oznacza zbiór wszystkich permutacji zbioru $\{1, 2, \ldots, n\}$, zaś $\mathrm{Inv}(\sigma)$ oznacza liczbę inwersji danej permutacji $\sigma \in S_n$.

Wyznacznik: 2×2

$$\det \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

Przykład

$$\det \begin{pmatrix} 5 & -7 \\ 3 & 8 \end{pmatrix} = 5 \cdot 8 - (-7) \cdot 3 = 61.$$

```
import numpy as np

A = np.array([[5 ,-7],[3, 8]])
print(np.linalg.det(A))#61.000000000000001
```

```
import sympy as sp

A = sp.Matrix([[5 ,-7],[3, 8]])
print(A.det())#61
```

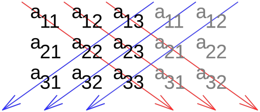
Wyznacznik: 3×3

Reguła Sarrusa, albo schemat Sarrusa

Aby obliczyć wyznacznik:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

dopisuje się z jego prawej strony dwie pierwsze kolumny, oblicza sumę iloczynów wzdłuż „czerwonych strzałek'' i odejmuje od niej sumę iloczynów wzdłuż „niebieskich strzałek'':



Źródło: Autorstwa Sarrus_rule.png: Kmhkmhderivative work: Marek M (talk) - Sarrus_rule.png, CC BY 3.0, `https://commons.wikimedia.org/w/index.php?curid=11145093`

Wzór ma postać:

$$(a_{11} \cdot a_{22} \cdot a_{33} + a_{12} \cdot a_{23} \cdot a_{31} + a_{13} \cdot a_{21} \cdot a_{32})$$
$$-(a_{13} \cdot a_{22} \cdot a_{31} + a_{11} \cdot a_{23} \cdot a_{32} + a_{12} \cdot a_{21} \cdot a_{33})$$

Wyznacznik: 3×3, przykład

$$A = \begin{pmatrix} 2 & 3 & 5 \\ -1 & 4 & 6 \\ 3 & -2 & 7 \end{pmatrix}$$

$$\det A = 105$$

Wyznacznik: 3×3, przykład

$$A = \begin{pmatrix} 2 & 3 & 5 \\ -1 & 4 & 6 \\ 3 & -2 & 7 \end{pmatrix}$$

$$\det A = 105$$

Wyznacznik: rozwinięcie Laplace'a

– wzór rekurencyjny określający wyznacznik n-tego stopnia macierzy kwadratowej o wymiarach $n \times n$.

Niech $A \in M_{n \times n}(K)$. Wówczas:

- dla każdego ustalonego $j = 1 \dots n$ zachodzi $\det A = \sum_{i=1}^{n} a_{ij} A_{ij}$,
- dla każdego ustalonego $i = 1 \dots n$ zachodzi $\det A = \sum_{j=1}^{n} a_{ij} A_{ij}$,

gdzie: A_{ij} jest dopełnieniem algebraicznym elementu a_{ij}., tzn. iloczyn $(-1)^{i+j}$ oraz wyznacznika podmacierzy stopnia $n-1$ powstałego z usunięcia i i-tego wiersza oraz j-ej kolumny macierzy A.

Przykład

Obliczmy wyznacznik następującej macierzy czwartego stopnia:

$$A = \begin{pmatrix} 0 & 1 & 2 & 7 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ -1 & 1 & -1 & 1 \end{pmatrix}.$$

$$\det A = -64$$

Przykład

Obliczmy wyznacznik następującej macierzy czwartego stopnia:

$$A = \begin{pmatrix} 0 & 1 & 2 & 7 \\ 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ -1 & 1 & -1 & 1 \end{pmatrix}.$$

$$\det A = -64$$

Macierz odwrotna

Niech A będzie macierzą kwadratową ustalonego stopnia. Macierz A jest *odwracalna*, jeśli istnieje taka macierz B, że zachodzi

$$AB = BA = I,$$

gdzie I jest macierzą jednostkową. Macierz B nazywa się wówczas macierzą *odwrotną* do macierzy A i oznacza się przez A^{-1}.

Jeżeli taka macierz B nie istnieje, to macierz A nazywamy *nieodwracalną*.

Macierz kwadratowa A jest odwracalna wtedy i tylko wtedy, kiedy $\det A \neq 0$.

Macierz odwrotną do nieosobliwej macierzy A obliczamy następująco:

$$A^{-1} = \frac{A^D}{\det A},$$

gdzie A^D jest macierzą dołączoną do macierzy A (czyli transponowaną macierzą dopełnień algebraicznych).

Macierz odwrotna: przykład

$$\begin{pmatrix} 8 & 5 \\ 13 & 8 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 3 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{pmatrix}$$

Macierz odwrotna: Python 1

```
import numpy as np
import sympy as sp

A = np.array([[8 ,5],[13, 8]])
print(np.linalg.inv(A))

A = sp.Matrix([[8 ,5],[13, 8]])
print(A.inv())
```

Wynik:
[[-8. 5.] [13. -8.]]
Matrix([[-8, 5], [13, -8]])

Macierz odwrotna: Python 2

```
import sympy as sp

a,b,c,d = sp.symbols('a,b,c,d')

A = sp.Matrix([[a ,b],[c, d]])
print(A.det())
print(A.inv())
```

Wynik:
a*d - b*c
Matrix([[d/(a*d - b*c), -b/(a*d - b*c)], [-c/(a*d - b*c), a/(a*d - b*c)]])

Układ równań

Układ n równań liniowych o m zmiennych postaci

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m = b_n \end{cases}$$

można zapisać w postaci równania macierzowego

$$\mathrm{A}\overline{x} = \overline{b}.$$

Jeżeli $\det A \neq 0$, to $\overline{x} = \mathrm{A}^{-1}\overline{b}$

Przykład 1

$$\begin{cases} 8x + 5y = 13 \\ 13x + 8y = 21 \end{cases}$$

można zapisać w postaci równania macierzowego

$$\begin{pmatrix} 8 & 5 \\ 13 & 8 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 13 \\ 21 \end{pmatrix}$$

wtedy

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 8 & 5 \\ 13 & 8 \end{pmatrix}^{-1} \begin{pmatrix} 13 \\ 21 \end{pmatrix} = \begin{pmatrix} -8 & 5 \\ 13 & -8 \end{pmatrix} \begin{pmatrix} 13 \\ 21 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Przykład 2

$$\begin{cases} 3x + 2y + 3z = 2 \\ -4x - 4y - 5z = 3 \\ x + y + z = 8 \end{cases}$$

można zapisać w postaci równania macierzowego

$$\begin{pmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 8 \end{pmatrix}$$

wtedy

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ 3 \\ 8 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 \\ -1 & 0 & 3 \\ 0 & -1 & -4 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 8 \end{pmatrix} = \begin{pmatrix} 21 \\ 22 \\ -35 \end{pmatrix}$$

Przykład 2: numpy

```
import numpy as np

A = np.array([[3,2, 3],[-4, -4, -5],[1,1, 1]])
#b = np.array([[2],[3],[8]])
b = np.array([2, 3, 8])
b.transpose()
x = np.linalg.inv(A)@b
print(x)
print(A@x==b)

print(np.linalg.solve(A, b))
```

Przykład 2: sympy

```
import sympy as sp

A = sp.Matrix([[3,2, 3],[-4, -4, -5],[1,1, 1]])
#b = sp.Matrix([[2],[3],[8]])
b = sp.Matrix([2, 3, 8])
b.transpose()
x=A.inv()*b
print(x)
print(A*x==b)

x, y, z = sp.symbols('x, y, z')
print(sp.solve([3*x+2*y + 3*z - 2, -4*x  -4*y-5*z -3, x + y+ z-8],
x, y,z, dict=True))
#[{x: 21, y: 22, z: -35}]

x, y, z = sp.symbols('x, y, z')
X = sp.Matrix([[x],[y],[z]])
print(sp.solve([A*X-b],x,y,z, dict=True))
print(A.solve(b))
```

Wzory Cramera

Niech dany będzie układ równań liniowych

$$x_1 \mathrm{a}_1 + \ldots + x_n \mathrm{a}_n = \mathrm{b},$$

gdzie $\mathrm{x} = (x_1, \ldots, x_n)$ oraz $\mathrm{b} = (b_1, \ldots, b_n)$.
Jeśli wyznacznik $\det(\mathrm{a}_1, \ldots, \mathrm{a}_n) \neq 0$, to układ jest *oznaczony* (ma jedno i tylko jedno rozwiązanie) dane wzorami:

$$x_1 = \frac{\det(\mathrm{b},\ \mathrm{a}_2, \ldots, \mathrm{a}_n)}{\det(\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_n)},$$

$$\vdots$$

$$x_n = \frac{\det(\mathrm{a}_1, \ldots, \mathrm{a}_{n-1},\ \mathrm{b})}{\det(\mathrm{a}_1, \ldots, \mathrm{a}_{n-1}, \mathrm{a}_n)}.$$

W przeciwnym przypadku, gdy $\det(\mathrm{a}_1, \ldots, \mathrm{a}_n) = 0$, układ jest

- *sprzeczny* (nie ma rozwiązań), gdy choć jeden wyznacznik we wzorach Cramera zawierający b jest różny od zera;
- *nieoznaczony* (ma więcej niż jedno rozwiązanie) lub sprzeczny, gdy wszystkie wyznaczniki we wzorach Cramera zawierające b są równe zeru.

Przykład 1

$$\begin{cases} 8x + 5y = 13 \\ 13x + 8y = 21 \end{cases}$$

można zapisać w postaci równania macierzowego

$$\begin{pmatrix} 8 & 5 \\ 13 & 8 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 13 \\ 21 \end{pmatrix}$$

wtedy

$$x = \frac{\begin{vmatrix} 13 & 5 \\ 21 & 8 \end{vmatrix}}{\begin{vmatrix} 8 & 5 \\ 13 & 8 \end{vmatrix}}, y = \frac{\begin{vmatrix} 8 & 13 \\ 13 & 21 \end{vmatrix}}{\begin{vmatrix} 8 & 5 \\ 13 & 8 \end{vmatrix}}.$$

Przykład 2

$$\begin{cases} 3x + 2y + 3z = 2 \\ -4x - 4y - 5z = 3 \\ x + y + z = 8 \end{cases}$$

można zapisać w postaci równania macierzowego

$$\begin{pmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 8 \end{pmatrix}$$

$$x = \frac{\begin{vmatrix} 2 & 2 & 3 \\ 3 & -4 & -5 \\ 8 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{vmatrix}}, y = \frac{\begin{vmatrix} 3 & 2 & 3 \\ -4 & 3 & -5 \\ 1 & 8 & 1 \end{vmatrix}}{\begin{vmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{vmatrix}}, z = \frac{\begin{vmatrix} 3 & 2 & 2 \\ -4 & -4 & 3 \\ 1 & 1 & 8 \end{vmatrix}}{\begin{vmatrix} 3 & 2 & 3 \\ -4 & -4 & -5 \\ 1 & 1 & 1 \end{vmatrix}}$$

Przykład Python numpy

```
import numpy as np

A = np.array([[3,2, 3],[-4, -4, -5],[1,1, 1]])
#b = sp.Matrix([[2],[3],[8]])
b = np.array([2, 3, 8])
X = A.copy()#Uwaga
Y = A.copy()
Z = A.copy()
X[:,0] = b#Uwaga
Y[:,1] = b
Z[:,2] = b
x = np.linalg.det(X)/np.linalg.det(A)
y = np.linalg.det(Y)/np.linalg.det(A)
z = np.linalg.det(Z)/np.linalg.det(A)

print(x,y,z)#21.0 22.000000000000004 -35.00000000000001
```

Przykład Python sympy

```
import sympy as sp

A = sp.Matrix([[3,2, 3],[-4, -4, -5],[1,1, 1]])
#b = sp.Matrix([[2],[3],[8]])
b = sp.Matrix([2, 3, 8])
X = A.copy()#Uwaga
Y = A.copy()
Z = A.copy()
X[0] = b
Y[1] = b
Z[2] = b
x = X.det()/A.det()
y = Y.det()/A.det()
z = Z.det()/A.det()

print(x,y,z)#21 22 -35
```

Matrix vs. array

```
import sympy as sp
import numpy as np
A = sp.Matrix([[1,2,3],[4, 5, 6],[7,8,9]])
B = np.array([[1,2,3],[4, 5, 6],[7,8,9]])
print(A[0])
print(A[8])
print(B[0])#print(B[8])
print(A[0,:])
print(B[0,:])
print(A[:,0])
print(B[:,0])
A[0] = sp.Matrix([1,1])
print(A)
A[0] = sp.Matrix([[1,1]])
print(A)
A[0] = sp.Matrix([[2],[2]])
print(A)
#B[0] = np.array([1,1])
B[0] = np.array([1,1,1])
print(B)
#B[0] = np.array([[1],[1],[1]])
```

Wektory w $\mathbb{R}^n$

Zbiór

$$\mathbb{R}^n = \{\overline{x} = (x_1, \ldots, x_n) \mid x_i in \mathbb{R}\}$$

z dodawaniem definiowanym dla każdych $\overline{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ oraz $\overline{y} = (y_1, \ldots, y_n) \in \mathbb{R}^n$ jako:

$$\overline{x} + \overline{y} = (x_1 + y_1, \ldots, x_n + y_n)$$

i mnożeniem na liczby, definiowanym jako:

$$a\overline{x} = (a \cdot x_1, \ldots, a \cdot x_n) \text{ dla każdego } a \in \mathbb{R},$$

jest *przestrzenią wektorową*.

Uwaga! Nie można dodawać do siebie wektorów z rożnych przestrzeni wektorowych!

Wektor

Właściwości:

1. Dodawanie wektorów jest łączne, tj. dla dowolnych $\overline{u}, \overline{v}, \overline{w} \in V$ jest $\overline{u}+(\overline{v}+\overline{w}) = (\overline{u}+\overline{v})+\overline{w}$.
2. Dodawanie wektorów jest przemienne, tj. dla dowolnych $\overline{v}, \overline{w} \in V$ jest $\overline{v}+\overline{w} = \overline{w}+\overline{v}$.
3. Dodawanie wektorów ma element neutralny, tj. istnieje taki element $\overline{0} \in V$, nazywany wektorem zerowym, że dla dowolnego $\overline{v} \in V$ jest $\overline{v}+\overline{0} = \overline{v}$.
4. Dodawanie wektorów ma elementy przeciwne, tj. dla każdego $\overline{v} \in V$ istnieje element $\overline{w} \in V$, nazywany wektorem przeciwnym do $\overline{v}$, taki że $\overline{v}+\overline{w} = \overline{0}$.
5. Mnożenie przez skalar jest rozdzielne względem dodawania wektorów, tj. dla każdego $a \in \mathbb{R}$ oraz $\overline{v}, \overline{w} \in V$ jest $a(\overline{v}+\overline{w}) = a\overline{v}+a\overline{w}$.
6. Mnożenie przez wektor jest rozdzielne względem dodawania skalarów, tj. dla każdych $a, b \in \mathbb{R}$ oraz $\overline{v} \in V$ zachodzi $(a+b)\overline{v} = a\overline{v}+b\overline{v}$.
7. Mnożenie przez skalar jest zgodne z mnożeniem liczb, tj. dla dowolnych $a, b \in \mathbb{R}$ oraz $\overline{v} \in V$ jest $a(b\overline{v}) = (a \cdot b)\overline{v}$.
8. Dla dowolnego $\overline{v} \in V$ zachodzi $1 \cdot \overline{v} = \overline{v}$.

Wektor

Przykłady na tablice.

Wektory w numpy

```
import numpy as np

# vector as row
vector1 = np.array([1, 2, 3])

# vector as column
vector2 = np.array([[10],
                    [20],
                    [30]])

print(vector1)
print(vector2)
print(2*vector1)
print(vector1+vector1)
print(2*vector2)
print(vector2+vector2)
```

Wektory w sympy

```
import sympy as sp

# vector as row
vector1 = sp.Matrix([1, 2, 3])

# vector as column
vector2 = sp.Matrix([[10],
                     [20],
                     [30]])
```

```
print(vector1)
print(vector2)
print(2*vector1)
print(vector1+vector1)
print(2*vector2)
print(vector2+vector2)
```

Wektory: liniowa zależność i baza

Wektory $\overline{v}_1, \overline{v}_2, \ldots \overline{v}_n$ przestrzeni V nazywany są *liniowo niezależnymi*, gdy zachodzi wynikanie:
Jeśli $a_1\overline{v}_1 + a_2\overline{v}_2 + \cdots + a_n\overline{v}_n = \overline{0}$, to $a_i = 0$ dla $i = 1, 2, \ldots, n,$.

Bazą przestrzeni V nazywa się liniowo niezależny zbiór wektorów $v_1, v_2, \ldots, v_n$, który rozpina przestrzeń V, tzn. każdy wektor $v \in V$ może być zapisany jako $\sum_i^n a_i v_i$ dla niektorych $a_i \in \mathbb{R}^n$.

Na przykład, wektory

$$e_1 = (1, 0, \ldots, 0)$$
$$e_2 = (0, 1, \ldots, 0)$$
$$e_n = (0, 0, \ldots, 1)$$

są baza w $\mathbb{R}^n$.

Theorem
W $\mathbb{R}^n$ każde n liniowo niezależnych wektorów są bazą.

Przykład 1

Wektory $(1, 1)$ i $(-3, 2)$ z $\mathbb{R}^2$ są liniowo niezależne. Rzeczywiście, niech a_1 oraz a_2 będą takimi liczbami rzeczywistymi, że

$$(1, 1)a_1 + (-3, 2)a_2 = (0, 0).$$

Biorąc każdą współrzędną z osobna, uzyskuje się układ równań z niewiadomymi a_1 i a_2:

$$\begin{cases} a_1 - 3a_2 = 0, \\ a_1 + 2a_2 = 0. \end{cases}$$

Jedynymi rozwiązaniami tego układu są $a_1 = 0$ i $a_2 = 0$, co potwierdza liniową niezależność wektorów.

Przykład 2

Podzbiór przestrzeni $\mathbb{R}^4$ złożony z wektorów jest liniowo zależny.

$$\begin{pmatrix}1\\4\\2\\-3\end{pmatrix}, \quad \begin{pmatrix}7\\10\\-4\\-1\end{pmatrix}, \quad \begin{pmatrix}-2\\1\\5\\-4\end{pmatrix}$$

Należy znaleźć takie liczby rzeczywiste a_1, a_2, a_3, nie wszystkie równe zeru, że

$$a_1\begin{pmatrix}1\\4\\2\\-3\end{pmatrix} + a_2\begin{pmatrix}7\\10\\-4\\-1\end{pmatrix} + a_3\begin{pmatrix}-2\\1\\5\\-4\end{pmatrix} = \begin{pmatrix}0\\0\\0\\0\end{pmatrix}.$$

Rozwiązując układ równań

$$\begin{cases} a_1 + 7a_2 - 2a_3 = 0, \\ 4a_1 + 10a_2 + a_3 = 0, \\ 2a_1 - 4a_2 + 5a_3 = 0, \\ -3a_1 - a_2 - 4a_3 = 0, \end{cases}$$

uzyskuje się

$$\begin{cases} a_1 = -\frac{3}{2}a_3, \\ a_2 = \frac{1}{2}a_3. \end{cases}$$

Wektory: liniowa zależność i wyznacznik

Wektory $\overline{v}_1, \overline{v}_2, \ldots \overline{v}_n$ przestrzeni $-\mathbb{R}^n$ są liniowo niezależnymi, wtedy i tylko wtedy, kiedy

$$\det(v_1, \ldots v_n) = 0$$

Np. dla wektorów $(1, 1)$ i $(-3, 2)$ z $\mathbb{R}^2$ odpowiednia macierz ma postać

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

Ponieważ

$$\det A = 1 \cdot 2 - 1 \cdot (-3) = 5 \neq 0,$$

więc wektory te są liniowo niezależne.

Macierz przejścia między bazami

Niech $U = \{\overline{u}_1, \ldots, \overline{u}_n\}$ i $V = \{\overline{v}_1, \ldots, \overline{v}_n\}$ będą dwoma bazami tej samej przestrzeni $\mathbb{R}^n$. Niech $t : \mathbb{R}^n \to \mathbb{R}^n$ będzie przekształceniem liniowym, które przekształca bazę E na bazę V, tzn. $t(\overline{u}_i) = \overline{v}_i$ Macierz T przekształcenia t w bazie U nazywamy macierzą przejścia od bazy U do bazy V.

Macierz tę wyznaczamy zatem z równości:

$$
\begin{aligned}
v_1 &= \alpha_1^1 u_1 + \alpha_1^2 u_2 + \cdots + \alpha_1^n u_n,\\
v_2 &= \alpha_2^1 u_1 + \alpha_2^2 u_2 + \cdots + \alpha_2^n u_n,\\
&\vdots\\
v_n &= \alpha_n^1 u_1 + \alpha_n^2 u_2 + \cdots + \alpha_n^n u_n.
\end{aligned}
$$

Czyli mamy

$$
S = \begin{pmatrix}
\alpha_1^1 & \alpha_2^1 & \cdots & \alpha_n^1\\
\alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2\\
\vdots & \vdots & \ddots & \vdots\\
\alpha_1^n & \alpha_2^n & \cdots & \alpha_n^n
\end{pmatrix}.
$$

Przykład cz.1

W przestrzeni $\mathbb{R}^3$, znaleźć macierz przejścia z bazy U do bazy V, gdzie

$$U = \{(0,1,0),(-1,0,0),(0,0,2)\},$$

$$V = \{(1,1,1),(0,1,2),(0,0,-1)\}.$$

Rozwiązanie: Zgodnie z powyższą definicją, obliczamy:

$$v_1 = (1,1,1) = \alpha u_1 + \beta u_2 + \gamma u_3 = \alpha(0,1,0) + \beta(-1,0,0) + \gamma(0,0,2).$$

Zatem dostajemy równość

$$(1,1,1) = (-\beta, \alpha, 2\gamma),$$

co daje nam

$$\alpha = 1, \quad \beta = -1, \quad \gamma = \frac{1}{2}.$$

Obliczone współczynniki tworzą pierwszą kolumnę szukanej macierzy przejścia S.

Przykład cz.2

Następnie:

$$v_2 = (0, 1, 2) = \alpha u_1 + \beta u_2 + \gamma u_3 = \alpha(0, 1, 0) + \beta(-1, 0, 0) + \gamma(0, 0, 2).$$

Zatem dostajemy równość

$$(0, 1, 2) = (-\beta, \alpha, 2\gamma),$$

co daje nam

$$\alpha = 1, \quad \beta = 0, \quad \gamma = 1.$$

Obliczone współczynniki tworzą drugą kolumnę szukanej macierzy przejścia S.

Przykład cz.3

I na koniec:

$$v_3 = (0,0,-1) = \alpha u_1 + \beta u_2 + \gamma u_3 = \alpha(0,1,0) + \beta(-1,0,0) + \gamma(0,0,2).$$

Zatem dostajemy równość

$$(0,0,-1) = (-\beta, \alpha, 2\gamma),$$

co daje nam

$$\alpha = 0, \quad \beta = 0, \quad \gamma = -\frac{1}{2}.$$

Obliczone współczynniki tworzą trzecią kolumnę szukanej macierzy przejścia S. Składając razem powyższe rozwiązania, dostajemy szukaną macierz przejścia z bazy U do bazy V:

$$S = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 0 & 0 \\ \frac{1}{2} & 1 & -\frac{1}{2} \end{pmatrix}.$$

Współrzędne wektora w dwóch bazach

Twierdzenie. Niech w przestrzeni $\mathbb{R}^n$ będą dane dwie bazy $U = \{u_i\}$ i $V = \{v_i\}$. Niech wektor $\overline{x} \in \mathbb{R}^n$ ma współrzędne $\{\alpha_i\}$ w bazie U oraz współrzędne $\{\beta_i\}$ w bazie V. Jeśli S jest macierzą przejścia od bazy U do bazy V, to zachodzi wzór:

$$\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix} = S \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}.$$

Macierz odwrotna S^{-1} (zawsze istnieje) jest macierzą przejścia od bazy V do U, oraz zachodzi wzór:

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix} = S^{-1} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}.$$

- S_{UV} — macierz przejścia z bazy U do bazy V;
- S_{VU} — macierz przejścia z bazy V do bazy U;
- $X_U = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{pmatrix}$ — współrzędne wektora $\overline{x}$ w bazie U;
- $X_V = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}$ — współrzędne wektora $\overline{x}$ w bazie V;

mamy następujące wzory:

$$X_V = S_{VU} X_U,$$
$$X_V = S_{UV}^{-1} X_U,$$
$$X_U = S_{UV} X_V,$$
$$X_U = S_{VU}^{-1} X_V.$$

Ponadto, zachodzi:

$$S_{UV} = S_{VU}^{-1},$$
$$S_{VU} = S_{UV}^{-1}.$$

Przykład

W przestrzeni $\mathbb{R}^2$ mamy dane dwie bazy:

$$U = \{(0,1),(1,1)\},$$

$$V = \{(2,2),(2,0)\}.$$

Mając współrzędne wektora $X_U = [1,2]$ w bazie U przejść do jego współrzędnych X_V w bazie V. Zrobić to na dwa sposoby: wprost oraz licząc macierz przejścia S_{VU}. Wprost:

$$\overline{x} = {\color{blue}1} \cdot {\color{red}(0,1)} + {\color{blue}2} \cdot (1,1) = (2,3) = a(2,2) + b(2,0), *$$

skąd $a = \frac{3}{2}, b = \frac{-1}{2}$.

$$*\overline{x} = \begin{bmatrix} {\color{red}0} & 1 \\ {\color{red}1} & 1 \end{bmatrix} \begin{bmatrix} {\color{blue}1} \\ {\color{blue}2} \end{bmatrix} = \begin{bmatrix} {\color{red}0} \cdot {\color{blue}1} + 1 \cdot {\color{blue}2} \\ {\color{red}1} \cdot {\color{blue}1} + 1 \cdot {\color{blue}2} \end{bmatrix}.$$

Rozwiązanie

Obliczamy macierz przejścia S_{VU}:

$$u_1 = (0,1) = \alpha v_1 + \beta v_2 = \alpha(2,2) + \beta(2,0) = (2\alpha + 2\beta, 2\alpha).$$

Zatem mamy równość

$$(0,1) = (2\alpha + 2\beta, 2\alpha),$$

co daje:

$$\alpha = \frac{1}{2}, \quad \beta = -\frac{1}{2}.$$

Podobnie dla u_2:

$$u_2 = (1,1) = \alpha v_1 + \beta v_2 = \alpha(2,2) + \beta(2,0) = (2\alpha + 2\beta, 2\alpha).$$

Zatem mamy równość

$$(1,1) = (2\alpha + 2\beta, 2\alpha),$$

co daje: $\alpha = \frac{1}{2}, \quad \beta = 0.$

Zatem

$$S_{VU} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix}.$$

Ostatecznie obliczamy:

$$X_V = S_{VU} X_U = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ -\frac{1}{2} \end{bmatrix}.$$

Rozwiązanie w numpy

```
import numpy as np

xu = np.array([1,2])
u = np.array([[0,1],[1,1]])
v = np.array([[2,2],[2,0]])
x = u@xu
print(x)
#u @ xu = x = v @ xv
xv = np.linalg.inv(v)@x
print(xv)
svu =  np.linalg.inv(v)@u
print(svu)
```

Rozwiązanie w sympy

```
import sympy as sp

xu = sp.Matrix([1,2])
xu.transpose()
u = sp.Matrix([[0,1],[1,1]])
v = sp.Matrix([[2,2],[2,0]])
x = u*xu
print(x)
#u * xu = x = v * xv
xv = v.inv()*x
print(xv)
svu =  v.inv()*u
print(svu)
```

Przekształcenie liniowe

Przekształcenie liniowe jest to funkcja $f\colon \mathbb{R}^n \to \mathbb{R}^m$ między przestrzeniami liniowymi zachowująca ich działania w tym sensie, że:

- odwzorowanie sumy wektorów z jednej przestrzeni w drugą jest równe sumie odwzorowań poszczególnych wektorów tej sumy,
- odwzorowanie iloczynu wektora przez skalar jest równe iloczynowi skalara przez odwzorowanie danego wektora.

Powyższe warunki można połączyć w jeden, równoważny z nimi warunek liniowości

$$f(c\,\overline{x} + d\,\overline{y}) = c\,f(\overline{x}) + d\,f(\overline{y}).$$

Przekształcenie liniowe

Każde przekształcenie liniowe między $\mathbb{R}^n$ oraz $\mathbb{R}^m$ można przedstawić za pomocą macierzy $m \times n$,. Np.: jeśli

$$(1,0,0) \to (1,2), (0,1,0) \to (1,1), (0,0,1) \to (1,0),$$

to dla

$$\overline{x} = (2,3,1) = 2 \cdot (1,0,0) + 3 \cdot (0,1,0) + 1 \cdot (0,0,1)$$

mamy

$$f(\overline{x}) = 2 \cdot (1,2) + 3 \cdot (1,1) + 1 \cdot (1,0) = (6,7)$$

Co można zapisać jako:

$$f(\overline{x}) = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}.$$

Tzn. macierz

$$\begin{pmatrix} 1 & 1 & 1 \\ 2 & 1 & 0 \end{pmatrix}$$

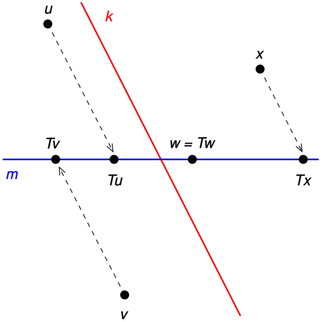
odpowiada przekształceniu f.

Rzut

Niech dana będzie przestrzeń liniowa $\mathbb{R}^n$ (nad ustalonym ciałem). Przekształcenie liniowe $P\colon \mathbb{R}^n \to \mathbb{R}^n$ tej przestrzeni w siebie spełniające warunek idempotentności

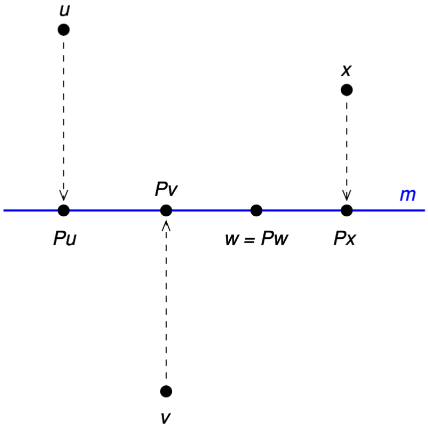
$$P^2 = P,$$

czyli $P\big(P(\overline{v})\big) = P(\overline{v})$ dla każdego $\overline{v} \in V$ nazywa się rzutem (ukośnym) lub projekcją.



Rzut T wzdłuż prostej k na prostą m.
Źródło: `https://commons.wikimedia.org/w/index.php?curid=1329674`

Rzut



Rzut ortogonalny P na prostą *m*.
Źródło: `https://commons.wikimedia.org/w/index.php?curid=1329670`
W tym kontekście rzut ukośny nazywa się *operatorem idempotentnym*, a *rzut ortogonalny* znany jest jako operator rzutowy.

Przykłady

- Przekształcenie tożsamościowe jest rzutem ortogonalnym reprezentowanym przez macierz jednostkową, np. $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ (operator jednostkowy jest operatorem rzutowym).
- Przekształcenie liniowe, którego macierz ma postać $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$, jest rzutem ortogonalnym, podczas gdy zadane macierzą $\left(\begin{smallmatrix} 0 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ jest rzutem (ukośnym), ale nie ortogonalnym (pierwsza macierz opisuje operator rzutowy, druga – tylko idempotentny).

Przykład

Ortogonalny rzut na prostą $x = y$ ma macierz

$$\begin{pmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

Niezerowy wektor v nazywamy *wektorem własnym* macierzy M, jeśli $Mv = \lambda v$ dla pewnej liczby λ. Liczbę λ nazywamy *wartością własną* macierzy M. Mówimy tez, ze wektor v należy do wartości własnej λ.

Inaczej, wektor własny macierzy, to taki niezerowy wektor, który nie zmienia kierunku po przekształceniu.

Przykład 1

Znaleźć wartości i wektory własne macierzy

$$\begin{pmatrix} 6 & 3 \\ 2 & 5 \end{pmatrix}.$$

Rozwiązanie. Wartości własne:

$$\begin{pmatrix} 6 & 3 \\ 2 & 5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} x \\ y \end{pmatrix}$$

jest równoważne układowi równań

$$\begin{cases} 6x + 3y = \lambda x, \\ 2x + 5y = \lambda y, \end{cases}$$

czyli

$$\begin{cases} (6-\lambda)x + 3y = 0, \\ 2x + (5-\lambda)y = 0. \end{cases}$$

Układ równań ma niezerowe rozwiązanie, o ile równania są do siebie proporcjonalne (liniowo zależne), co można ująć w postaci równania

$$0 = \begin{vmatrix} 6-\lambda & 3 \\ 2 & 5-\lambda \end{vmatrix} = (6-\lambda)(5-\lambda) - 2 \cdot 3 = \lambda^2 - 11\lambda + 24,$$

skąd $\lambda = 3 \quad \text{lub} \quad \lambda = 8$.

Przykład 1: wektory własne

Dla każdej z wartości własnych znajdziemy wektor własny (tutaj wystarczy jeden).
Dla $\lambda = 3$:

$$6x + 3y = 3x \implies 3x + 3y = 0 \implies x = -y,$$

więc wektor własny to

$$v = \begin{pmatrix} 1 \\ -1 \end{pmatrix}.$$

Dla $\lambda = 8$:

$$6x + 3y = 8x \implies -2x + 3y = 0 \implies 2x = 3y,$$

więc wektor własny to

$$v = \begin{pmatrix} 3 \\ 2 \end{pmatrix}.$$

Algorytm: wartości własne

Szukamy λ taki ze $|A - \lambda I| = 0$. Szukamy wektory własne $|A - \lambda I|\overline{x} = \overline{b}$.

Znaleźć wartości i wektory własne macierzy

$$\begin{pmatrix} 5 & 4 & 4 \\ -3 & -3 & -5 \\ 1 & 2 & 4 \end{pmatrix}.$$

(na tablice, $\lambda = 1, 2, 3$)

```
import sympy as sp

A = sp.Matrix([[5,4, 4],[-3, -3, -5],[1,2, 4]])
x = sp.symbols('x')
print(sp.solve([(A-x*sp.eye(3)).det()],x, dict=True))
print((A-x*sp.eye(3)).det())
```

Python

```
import numpy as np

A = np.array([[5,4, 4],[-3, -3, -5],[1,2, 4]])
print(np.linalg.eigvals(A))
```

```
import sympy as sp

A = sp.Matrix([[5,4, 4],[-3, -3, -5],[1,2, 4]])
print(A.eigenvals())
```

Algorytm: wektory własne

Szukamy λ taki ze $|A - \lambda I| = 0$. Szukamy wektory własne $|A - \lambda I|\overline{x} = \overline{b}$.

Znaleźć wartości i wektory własne macierzy

$$\begin{pmatrix} 5 & 4 & 4 \\ -3 & -3 & -5 \\ 1 & 2 & 4 \end{pmatrix}.$$

(na tablice)

```
import sympy as sp

A = sp.Matrix([[5,4, 4],[-3, -3, -5],[1,2, 4]])
y,z = sp.symbols('y,z')
X = sp.Matrix([[1],[y],[z]])
for i in 1,2,3:
    print(sp.solve ([(A-i*sp.eye(3))*X],y,z,dict=True))
X = sp.Matrix([[y], [1], [z]])#uwaga
print(sp.solve([(A - 2 * sp.eye(3)) * X], y, z, dict=True))
```

Python

```
import numpy as np

A = np.array([[5,4, 4],[-3, -3, -5],[1,2, 4]])
print(np.linalg.eig(A))
```

```
import sympy as sp

A = sp.Matrix([[5,4, 4],[-3, -3, -5],[1,2, 4]])
print(A.eigenvects())
#(eigenvalue, algebraic_multiplicity, [eigenvectors])
```

Czy zawsze istnieją wektory własne?

Macierz może nie mieć wektorów własnych. Rozważmy macierz

$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

Aby znaleźć wartości własne, obliczamy wyznacznik:

$$0 = \begin{vmatrix} -\lambda & -1 \\ 1 & -\lambda \end{vmatrix} = \lambda^2 + 1.$$

Rozwiązań możemy jednak szukać wśród liczb zespolonych:

$$\lambda = \pm i, \quad v = \begin{pmatrix} \pm i \\ 1 \end{pmatrix}.$$

Dopuszczając liczby zespolone, zawsze będziemy mieli co najmniej jeden wektor własny.

Uwagi

- W zastosowaniach wygodnie jest mieć bazę złożoną z wektorów własnych rozważanej macierzy.
- Wektory własne należące do różnych wartości własnych są liniowo niezależne. Dlatego, gdy wszystkie wartości własne są różne, istnieje baza złożona z wektorów własnych.
- W przypadku macierzy symetrycznej zawsze znajdziemy bazę złożoną z wektorów własnych.
- Czasem taka baza nie istnieje. Ale dla większości macierze – istnieje ($\Leftrightarrow$ macierz jest diagonalizowalna $\Leftrightarrow$ $A = BDB^{-1}$).

Rozważmy macierz $M = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$. Dla tej macierzy mamy $M^2 = 0$, dlatego wartości własne mogą być tylko zerami (można to sprawdzić metodą zastosowaną w zadaniach). Gdyby wektory własne u i v tworzyły bazę, to dla dowolnego wektora w mielibyśmy:

$$Mw = M(au + bv) = aMu + bMv = 0.$$

Oznaczałoby to, że $M = 0$, co jest sprzeczne z definicją macierzy M.