

project_matrix_norm

April 17, 2024

Rachunek macierzowy Projekt 3 Dominik Tomalczyk, Piotr Van-selow

```
[1]: from numpy.linalg import inv, eigvals
import numpy as np
```

$$\text{cond}(A) = \|A\| \|A^{-1}\|$$

$$\text{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2$$

1. Współczynnik uwarunkowania macierzy

```
[2]: def cond_matrix(A, norm) -> float:
    return norm(A) * norm(inv(A))
```

$$\|A\|_1 = \max_{x \neq 0} \frac{\|Ax\|_1}{\|x\|_1}$$

2. Norma macierzowa M1

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1,\dots,n} |a_{ij}|$$

Można ją również wyliczyć równoznacznie:

```
[3]: A1 = [[1,2,3],
          [3,5,33],
          [8,9,6],]

def m1_norm(A) -> float:
    num_columns = len(A[0])
    column_sums = [0] * num_columns

    for row in A:
        for i in range(num_columns):
            column_sums[i] += row[i]
```

```

return max(column_sums)

print("Norma M1 dla macierzy A1: {:.2f}".format(m1_norm(A1)))
print(f"Współczynnik uwarunkowania macierzy A1 z normą M1: {cond_matrix(A1, m1_norm)}")

```

Norma M1 dla macierzy A1: 42.00

Współczynnik uwarunkowania macierzy A1 z normą M1: 5.870967741935485

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}$$

2. Norma macierzowa M2

Inny sposób obliczania

$$\|A\|_2 = |\lambda_1|$$

Można ją również wyliczyć:

gdzie $|\lambda_1|$ to największa (na moduł) wartość własna macierzy

```

[4]: A2 = [
        [11,23,2],
        [7,5,7],
        [8,6,6],
    ]
def m2_norm(A) -> float:
    eigenvalues, eigenvectors = np.linalg.eig(A)
    return max(eigenvalues)

print("Norma M2 dla macierzy A2: {:.2f}".format(m2_norm(A2)))
print(f"Współczynnik uwarunkowania macierzy A2 z normą M2: {cond_matrix(A2, m2_norm)}")

```

Norma M2 dla macierzy A2: 25.03+0.00j

Współczynnik uwarunkowania macierzy A2 z normą M2: (1.000000000000001+0j)

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1,\dots,n} |a_{ij}|$$

3. Norma macierzowa M inf

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1,\dots,n} |a_{ij}|$$

Można ją również wyliczyć:

```

[5]: A3 = [
        [7,1,2],
    ]

```

```

    [3,0,8],
    [13,1,1],
]
def minf_norm(A) -> float:
    row_sums = []
    for row in A:
        row_sums.append(sum(row))

    return max(row_sums)

print("Norma Minf dla macierzy A3: {:.2f}".format(minf_norm(A3)))
print(f"Współczynnik uwarunkowania macierzy A3 z normą Minf: {cond_matrix(A3, minf_norm)}")

```

Norma Minf dla macierzy A3: 15.00

Współczynnik uwarunkowania macierzy A3 z normą Minf: 9.411764705882353

4. Norma macierzowa M_p ???
5. Rozkład SVD dla macierzy

Każdą macierz $A \in \mathcal{R}^{n \times m}$ da się zdekomponować

$$A = U \Sigma V^T$$

Postać rozkładu SVD:

gdzie: U - macierz wektorów własnych AA^T V - macierz wektorów własnych $A^T A$ Σ - macierz która na przekątnej ma pierwiastki wartości własnych macierzy AA^T

```

[6]: def svd_eigen(A):
    # Oblicz macierz  $A^T A$ 
    ATA = np.dot(A.T, A)

    # Oblicz wartości własne i wektory własne macierzy  $A^T A$ 
    eigenvalues, eigenvectors = np.linalg.eig(ATA)

    # Sortowanie malejące wartości własnych i wektorów własnych
    idx = eigenvalues.argsort()[::-1]
    eigenvalues = eigenvalues[idx]
    eigenvectors = eigenvectors[:,idx]

    # Macierz  $V$  - prawych wektorów własnych
    V = eigenvectors

    # Macierz  $\Sigma$  - diagonalna macierz wartości osobliwych
    Sigma = np.diag(np.sqrt(eigenvalues))

    # Macierz  $U$  - lewych wektorów własnych
    # Ze wzoru  $A = USVT$  wyznaczamy  $U$ :  $U = A * V * \Sigma(-1)$ 
    U = np.dot(A, np.dot(V, np.linalg.inv(Sigma)))

```

```

    return U, Sigma, V.T

# Przykładowa macierz A
A = np.array([[1, 2, 3], [2, 0, 7], [4, 5, 6]])

# Oblicz SVD
U, Sigma, VT = svd_eigen(A)
print("Macierz U:")
print(U)
print("\nMacierz Sigma:")
print(Sigma)
print("\nMacierz V^T:")
print(VT)
print()
print("Rekonstrukcja macierzy U*Sigma*V.T :")
print(U @ Sigma @ VT)

```

Macierz U:

```

[[ 0.32192765 -0.11663948 -0.93955192]
 [ 0.58234916  0.80684247  0.09937142]
 [ 0.74647977 -0.57913768  0.32766981]]

```

Macierz Sigma:

```

[[11.39327489  0.          0.          ]
 [ 0.          3.71293842  0.          ]
 [ 0.          0.          0.63825973]]

```

Macierz V^T:

```

[[ 0.39256009  0.38410854  0.83567769]
 [-0.22071609 -0.84271997  0.49102694]
 [ 0.89284992 -0.37720509 -0.24603929]]

```

Rekonstrukcja macierzy U*Sigma*V.T :

```

[[ 1.00000000e+00  2.00000000e+00  3.00000000e+00]
 [ 2.00000000e+00 -1.44554742e-15  7.00000000e+00]
 [ 4.00000000e+00  5.00000000e+00  6.00000000e+00]]

```

Wszystkie obrazki pochodzą z: Maciej Paszyński Wydział Informatyki Akademia Górniczo-Hutnicza w Krakowie home.agh.edu.pl/paszynsk

[]: