Metoda potęgowa i dekompozycja SVD
Raport z implementacji i analizy numerycznej

Bartosz Hanc, Paweł Surdyka

18 maja 2025

Wstęp teoretyczny

Celem niniejszego raportu jest analiza numeryczna dwóch ważnych metod związanych z teorią macierzy: metody potęgowej oraz dekompozycji SVD (Singular Value Decomposition). Obie techniki są fundamentalnymi narzędziami w algebrze liniowej numerycznej, a ich zastosowania obejmują m.in. analizę danych, kompresję informacji, metody iteracyjne oraz rozwiązywanie układów równań.

Metoda potęgowa

Metoda potęgowa (ang. *Power Method*) jest prostą i efektywną iteracyjną techniką znajdowania dominującej (co do modułu największej) wartości własnej macierzy kwadratowej $A \in \mathbb{R}^{n \times n}$ oraz odpowiadającego jej wektora własnego. Algorytm bazuje na wielokrotnym przemnażaniu początkowego wektora przez macierz A, co asymptotycznie prowadzi do zbieżności w kierunku wektora własnego związanego z największą wartością własną.

Zbieżność metody potęgowej jest gwarantowana w przypadku, gdy:

- macierz A posiada jednoznacznie największą wartość własną (dominującą),
- wektor startowy posiada składową w kierunku tej wartości własnej.

Algorytm jest prosty w implementacji, lecz jego skuteczność zależy m.in. od wyboru normy wykorzystywanej do normalizacji oraz od struktury spektrum macierzy.

Dekompozycja SVD

Dekompozycja według wartości osobliwych (ang. *Singular Value Decomposition*) jest jedną z najbardziej wszechstronnych i stabilnych metod analizy macierzy. Każdą macierz $A \in \mathbb{R}^{m \times n}$ można przedstawić jako iloczyn trzech macierzy:

$$A = UDV^T,$$

gdzie:

- $U \in \mathbb{R}^{m \times m}$ — macierz ortogonalna zawierająca wektory własne macierzy AA^T,
- $D \in \mathbb{R}^{m \times n}$ — macierz diagonalna zawierająca tzw. wartości osobliwe (pierwiastki z wartości własnych macierzy A^TA lub AA^T),

- $V \in \mathbb{R}^{n\times n}$ — macierz ortogonalna zawierająca wektory własne macierzy $A^T A$.

W praktyce dekompozycja SVD jest stosowana m.in. do:

- redukcji wymiarowości (np. w metodzie PCA),
- wykrywania rangi macierzy,
- kompresji obrazów,
- rozwiązywania układów równań nadokreślonych i niedookreślonych.

W niniejszym projekcie dekompozycja SVD została przeprowadzona zarówno przy użyciu własnej implementacji (na podstawie wartości własnych macierzy AA^T), jak i z wykorzystaniem biblioteki numerycznej, co umożliwia porównanie dokładności tych metod.

a) Pseudokod algorytmu potęgowego

- Losuj wektor z_0 o współrzędnych z przedziału $(0, 1)$
- Oblicz $w_0 = Az_0$
- Jeśli $||Az_0 - \max(w_0)z_0||_p < 10^{-8}$, wróć do początku
- Normalizuj z_0 do wybranej normy p
- Iteruj:
  - $w = Az$
  - $z_{\text{new}} = w/||w||_p$
  - $\text{error} = ||Az - \max(w)z||_p$
  - jeśli $\text{error} < \epsilon$, zakończ
  - w przeciwnym razie $z = z_{\text{new}}$

b) Fragmenty kodu źródłowego (Python)

```
def power_method(A, z0, epsilon=1e-4, p=2):
    errors = []
    z = z0 / norm(z0, ord=p)
    while True:
        w = A @ z
        w_norm = norm(w, ord=p)
        z_new = w / w_norm
        error = norm(A @ z - np.max(w) * z, ord=p)
        errors.append(error)
        if error < epsilon:
            break
        z = z_new
    return errors
```

c) Wylosowana macierz A

$$A = \begin{bmatrix} 73 & 70 & 19 \\ 67 & 88 & 1 \\ 34 & 42 & 48 \end{bmatrix}$$

d) Macierze U, D, V z dekompozycji SVD

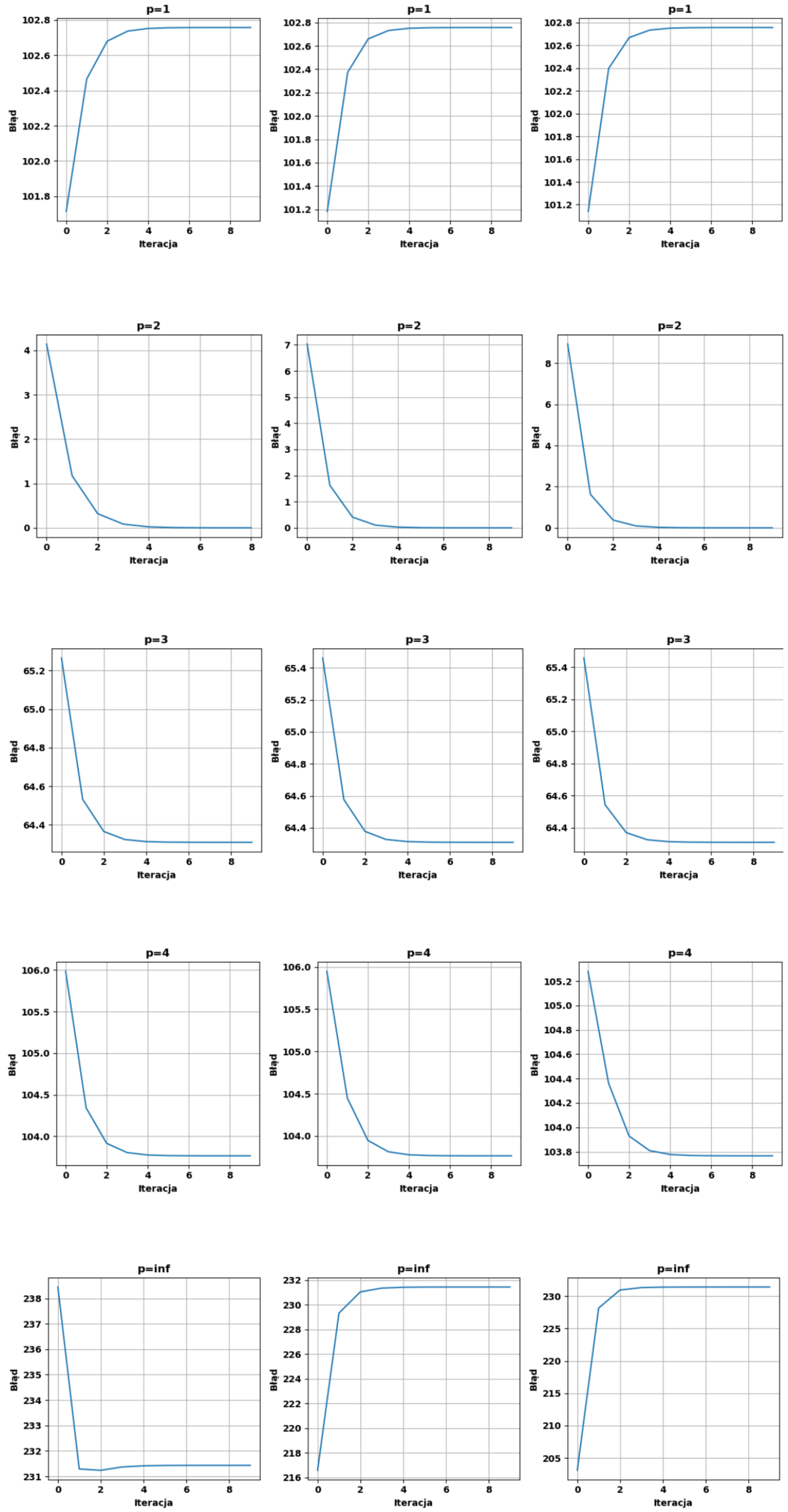
$$U = \begin{bmatrix} 0.63391397 & 0.77334425 & -0.00957834 \\ 0.67081645 & -0.5559508 & -0.49084009 \\ 0.38491344 & -0.30472508 & 0.87119703 \end{bmatrix},$$

$$D = \begin{bmatrix} 161.7109745 & 0 & 0 \\ 0 & 11.66394013 & 0 \\ 0 & 0 & 41.97038511 \end{bmatrix},$$

$$V = \begin{bmatrix} 0.64502412 & 0.75830068 & -0.09446674 \\ 0.73941915 & -0.65055432 & -0.17331592 \\ 0.19288133 & -0.04194243 & 0.98032527 \end{bmatrix}$$

e) Wykresy zbieżności błędu dla różnych norm i wektorów

Poniżej przedstawiono wykresy przedstawiające błąd w kolejnych iteracjach dla 3 różnych wektorów startowych i norm: $p = 1, 2, 3, 4, \infty$. Na osiach: X — numer iteracji, Y — błąd.



Rysunek 1: Wykresy zbieżności błędu dla różnych norm i wektorów początkowych

f) Porównanie dokładności dekompozycji SVD

Wartości błędu oszacowania dekompozycji SVD na podstawie wzoru $||UDV - \text{SVD}(A)||_p$ dla różnych norm:

- $||UDV - \text{SVD}(A)||_1 = 1.1369e - 13$
- $||UDV - \text{SVD}(A)||_2 = 1.1612e - 13$
- $||UDV - \text{SVD}(A)||_\infty = 1.6342e - 13$

Wnioski

W trakcie eksperymentów zaobserwowano zbieżność metody potęgowej dla różnych norm $||\cdot||_p$. Dla każdej z analizowanych norm wykresy błędu wskazują na szybki spadek wartości błędu w pierwszych kilku iteracjach, a następnie "zamarznięcie" błędu na stałym poziomie, co oznacza, że zmiany wektora były niewielkie lub numerycznie niezauważalne.

Dla normy $p = 2$ — będącej najczęściej stosowaną normą euklidesową — wartości błędu zamarzały w pobliżu zera (rzędu 10^{-4} lub mniejszego), co świadczy o dobrej zbieżności metody w tym przypadku i osiągnięciu przybliżonego wektora własnego.

Natomiast dla norm $p = 1$, $p = 3$, $p = 4$ oraz $p = \infty$, wartości błędu zamarzały na znacznie wyższych poziomach, rzędu około 65, 100 lub nawet 200, w zależności od konkretnego przypadku. Może to wynikać z większej wrażliwości tych norm na wybrane składowe wektora lub ograniczeń precyzji numerycznej w tych normach.

Dodatkowo, podczas obliczania błędu porównującego dekompozycję SVD z własnej implementacji (na podstawie AA^T) z dekompozycją uzyskaną za pomocą biblioteki scipy.linalg.svd, zauważono, że biblioteka numpy.linalg.norm nie obsługuje norm $p = 3$ oraz $p = 4$ dla macierzy. W związku z tym, w podpunkcie **f** przedstawiono porównanie błędów jedynie dla $p = 1$, $p = 2$ oraz $p = \infty$.

Ogólnie metoda potęgowa działa zgodnie z oczekiwaniami, jednak jej efektywność i precyzja końcowa są istotnie zależne od wybranej normy, co należy wziąć pod uwagę przy analizie numerycznej oraz implementacjach praktycznych.