**Algorytmy macierzowe**

**Laboratorium 3**

**Sprawozdanie**

**Łukasz Stępień, Szymon Urbański**

1. **Temat zadania**

Laboratorium polegało na zaimplementowaniu i przetestowaniu rekurencyjnego algorytmu kompresji macierzy z wykorzystaniem częściowego SVD dla wybranych parametrów.

1. **Rozwiązanie**

Zaimplementowano rekurencyjną kompresje macierzy.

* **Pseudokod algorytmu:**

1. **Tworzenie drzewa kompresji**

create\_tree(, , , , r, ε):

[U, D, V] = truncatedSVD(A[:, :], r+1)

if D[r+1] < ε then

v = compress\_matrix(, , , , U, D, V, r)

else

create node v

v.append(create\_tree(, , , , r, ε)

v.append(create\_tree(, , , , r, ε)

v.append(create\_tree(, , , , r, ε)

v.append(create\_tree(, , , , r, ε)

return v

1. **Kompresja SVD**

compress\_matrix(, , , , U, D, V, r):

create node v

if A[:, :] is zeros then

v.U <- zeros( - , - )

v.V <- zeros( - , - )

else

v.size <- (, , , )

v.rank <- r

v.singular\_values <- D[0 : r + 1]

v.U = U[\*, 0 : r + 1]

v.V = diag(D[0: r + 1]) \* V[0 : r + 1, \*]

return v

1. **Dekonstrukcja macierzy z macierzy skompresowanej**

decompress(B, v):

if v has no children then:

B[v.size] = v.U \* v.V

else

for child in v.children:

B = decompress(B, child)

return B

* **Najważniejsze fragmenty kodu:**

Poniżej przedstawiono fragmenty kodu realizujące powyższe pseudokody oraz fragment implementujący rysowacz.

**Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie**

**Kod 1.** Implementacja klasy Node

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

**Kod 2.** Implementacja tworzenia drzewa.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Kod 3.** Implementacja kompresji SVD macierzy

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

**Kod 4.** Implementacja dekonstrukcji macierzy

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

**Kod 5.** Implementacja rysowacza

1. **Wyniki**

Algorytm przetestowano dla 5 macierzy x dla k=10, odpowiednio zawierających 1,2,5,10,20 procent wartości niezerowych. Do funkcji przekazywano maksymalny rząd równy 1 oraz za najmniejszą wartość osobliwą przyjmowano tą ostatnią (czyli rzeczywiście najmniejszą). Dla każdej z macierzy podano czas kompresji, uruchomiono SVD i znaleziono wartości osobliwe, dokonano dekonstrukcji i obliczenia MSE. Wyniki przedstawiono na wykresach.

Obraz zawierający tekst, zrzut ekranu, diagram, Czcionka

Opis wygenerowany automatycznie

**Wykres 3.1** Czas kompresji w zależności od ilości wartości niezerowych macierzy

**Obraz zawierający tekst, zrzut ekranu, wyświetlacz, linia

Opis wygenerowany automatycznie**

**Wykres 3.2** Wartości osobliwe poszczególnych macierzy

**Obraz zawierający tekst, zrzut ekranu, diagram, Prostokąt

Opis wygenerowany automatycznie**

**Wykres 3.3** Błąd dekonstrukcji macierzy dla poszczególnych przypadków

Na poniższych wykresach przedstawiono narysowane rekurencyjne dekompozycje macierzy, w zależności od obranego maksymalnego rzędu macierzy i minimalnej wartości osobliwej. Testy przeprowadzono dla macierzy o wymiarach x w celu uzyskanie przejrzystości wyniku.

1. Ilość wartości niezerowych: 1%

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, Prostokąt, kwadrat

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający krzyżówka, kwadrat, zrzut ekranu, Prostokąt

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający kwadrat, wzór, zrzut ekranu, krzyżówka

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, wyświetlacz, tekst, Prostokąt

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający Prostokąt, kwadrat, zrzut ekranu, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, kwadrat, Prostokąt, tekst

Opis wygenerowany automatycznie

1. Ilość wartości niezerowych: 2%

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, Prostokąt, kwadrat, linia

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający kwadrat, zrzut ekranu, Prostokąt, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający czarne i białe, kwadrat, zrzut ekranu, ścieg

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, wyświetlacz, tekst, Prostokąt

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, kwadrat, Prostokąt, Symetria

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, kwadrat, wzór, Prostokąt

Opis wygenerowany automatycznie

1. Ilość wartości niezerowych: 5%

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, Prostokąt, kwadrat, linia

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający kwadrat, zrzut ekranu, Prostokąt, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający ścieg, wzór, czarne i białe, zrzut ekranu

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, wyświetlacz, tekst, Prostokąt

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający kwadrat, Prostokąt, wzór, zrzut ekranu

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, ścieg, czarne i białe, kwadrat

Opis wygenerowany automatycznie

1. Ilość wartości niezerowych: 10%

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, Prostokąt, kwadrat, linia

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający kwadrat, Prostokąt, zrzut ekranu, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający ścieg, tekst, czarne i białe, zrzut ekranu

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, wyświetlacz, tekst, Prostokąt

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający kwadrat, zrzut ekranu, Prostokąt, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, czarne i białe, wzór, kwadrat

Opis wygenerowany automatycznie

1. Ilość wartości niezerowych: 20%

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, Prostokąt, kwadrat, linia

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający kwadrat, Prostokąt, zrzut ekranu, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 1, minimalna wartość osobliwa:

Obraz zawierający tekst, ścieg, czarne i białe, zrzut ekranu

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający zrzut ekranu, wyświetlacz, tekst

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający kwadrat, zrzut ekranu, Prostokąt, wzór

Opis wygenerowany automatycznie

* Maksymalny rząd: 4, minimalna wartość osobliwa:

Obraz zawierający ścieg, tekst, czarne i białe, monochromatyzm

Opis wygenerowany automatycznie

1. **Wnioski**

Analizując wykres 3.1 można stwierdzić, że czas działania rekurencyjnej kompresji SVD rośnie wraz z ilością niezerowych wartości (pomimo faktu że macierz ma takie same rozmiary). Jest to spowodowane tym, że w momencie kompresji sprawdzane jest, czy macierz nie jest zerowa. Jeżeli tak, to jej kompresja też będzie składała się z samych zer, więc pomija się obliczenia związane z mnożeniem macierzy (co w oczywisty sposób skraca czas działania algorytmu).

Biorąc pod uwagę dane z wykresu 3.2 można stwierdzić, że w każdym przypadku wartości osobliwe maleją oraz w pewnym momencie są bardzo bliskie zeru. To dlatego można czasem używać stratnej kompresji macierzy, ponieważ niewielkie wartość osobliwa nie wpłynie w znaczącym stopniu na błąd. Należy tu jednak pamiętać, że błąd ten może się kumulować i przy większych rozmiarach macierzy może on już być bardziej znaczący.

Na podstawie wykresu 3.3 można stwierdzić, że zaimplementowany algorytm zadziałał poprawnie. Błąd kwadratowy jest rzędu , a co za tym idzie można powiedzieć, że uzyskane wyniki są prawidłowe.

Porównując uzyskane rysunki podczas kompresji macierzy można zauważyć, że w każdym przypadku ilość rekurencyjnych wywołań rośnie wraz ze wzrostem minimalnej wartości osobliwej (dopuszczalnego błędu), ale maleje wraz ze wzrostem rządu macierzy. Dodatkowo dla większej ilości niezerowych wartości liczba wywołań rekurencyjnych także wzrasta, co tłumaczy również fakt dłuższego czasu działania. Warto również podkreślić, że rysunki te nie są symetryczne, a co za tym idzie zależą one od ułożenia wartości w macierzy.

1. **Bibliografia**

* wykład z przedmiotu „Algorytmy macierzowe” przygotowany przez prof. dr hab. Macieja Paszyńskiego
* https://en.wikipedia.org/wiki/Singular\_value\_decomposition