

SPRAWOZDANIE

Zajęcia: Matematyka Konkretna

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium Nr 1 Data 3.10.2023 Temat: SVD Wariant 6	Rafał Klinowski Informatyka II stopień, stacjonarne, 2 semestr, gr. a
--	--

1. Polecenie:

Ćwiczenie polegało na stworzeniu notatnika Jupyter w języku Python do opanowania analizy macierzowej na podstawie obrazu w metodzie SVD.

Wariant zadania: 6

Zadanie dotyczy kompresji obrazu metodą SVD zgodnie z wariantem zadania. Jaka powinna być użyta liczba wartości singularnych żeby zachować 90% informacji na obrazie



2. Napisany program, uzyskane wyniki

Podczas implementacji skorzystano z domyślnej wersji implementacji SVD w instrukcji laboratoryjnej. W pierwszym kroku przeprowadzono identyczną

analizę wstępną, a następnie zbadano krok po kroku dla jakich wartości uzyskana ilość informacji na obrazie wynosi 90%.

```
from matplotlib.image import imread
import matplotlib.pyplot as plt
import numpy as np

# Załadowanie obrazu
A = imread('6.webp')
X = np.mean(A, -1) # convert RGB to grayscale

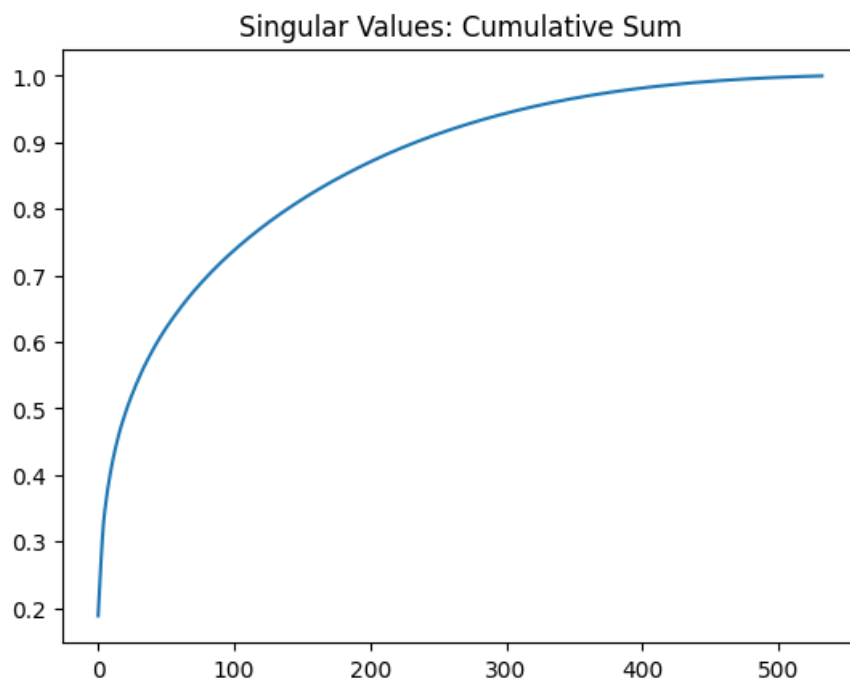
# Pokazanie obrazu
img = plt.imshow(X)
img.set_cmap('gray')
plt.axis('off')
plt.show()

# SVD - zgodnie z instrukcją laboratoryjną (wstęp)
U, S, VT = np.linalg.svd(X, full_matrices=False)
print(S.shape)
S = np.diag(S)

j=0
for r in (5, 20, 100, 650):
    Xapprox = U[:, :r] @ S[0:r, :r] @ VT[:, r, :]
    plt.figure(j+1)
    j += 1
    img = plt.imshow(Xapprox)
    img.set_cmap('gray')
    plt.axis('off')
    plt.title('r='+str(r))
    plt.show()

plt.figure(1)
plt.plot(np.cumsum(np.diag(S)) / np.sum(np.diag(S)))
plt.title('Singular Values: Cumulative Sum')
plt.show()
```

Rysunek 1. Domyślna implementacja w celu oszacowania przedziału oraz sprawdzenia działania SVD.



Rysunek 2. Uzyskane wyniki – procent w zależności od ilości wartości singularnych.

Następnie przeprowadzono bardziej szczegółową analizę dla przedziału <200; 400> odczytanego z powyższego wykresu.

```

# Przeprowadzenie dokładnej analizy w celu zobaczenia, dla jakiej wartości
uzyskamy 90%.
# Weźmy zawężony przedział <200; 400> - na podstawie wykresu wyżej

U, S, VT = np.linalg.svd(X, full_matrices=False)
S = np.diag(S)

bestR = 200
bestX = None
bestInfo = 0
for r in range(200, 400):
    # Obraz dla tego r
    Xapprox = U[:, :r] @ S[0:r, :r] @ VT[:, :]
    bestR += 1

    # Sprawdamy jaki jest %
    percentage = (np.cumsum(np.diag(S)) / np.sum(np.diag(S))) [r]
    if percentage > 0.90:
        bestInfo = percentage
        bestX = Xapprox
        break

plt.figure(1)
img = plt.imshow(bestX)
img.set_cmap('gray')
plt.axis('off')
plt.title('r='+str(bestR))
plt.show()
print("Dla r = " + str(bestR) + " uzyskano " + str(bestInfo * 100) + "%
informacji (pierwsza wartość powyżej 90%).")

```

Rysunek 3. Kod wyszukujący wartość, dla której przekroczony zostanie przedział 90%, zapisujący również tę wartość wraz z ilością wartości singularnych i uzyskanym obrazem.

r=235



Rysunek 4. Uzyskany obraz.

Wynikiem jest $r=235$ (ilość wartości singularnych) oraz 90.07554125658388% jako ilość zachowanych informacji (jest to pierwsza uzyskana wartość wyższa od 90%).

Repozytorium zawierające uzyskane wyniki wraz z niezbędnymi plikami:

<https://github.com/Stukeley/MatematykaKonkretna/tree/master/Lab1>