



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Лабораторная работа № 9
по курсу «Численные методы линейной алгебры»
«Реализация сингулярного разложения матрицы (SVD)»

Студентка группы ИУ9-72Б Самохвалова П. С.

Преподаватель Посевин Д. П.

Москва 2023

1 Цель

Целью работы является самостоятельное изучение сингулярного разложения матрицы или SVD разложения.

2 Задание

1. Реализовать SVD разложение матрицы.
2. Проверить корректность реализации прямым перемножением матриц.
3. Проверить через numpy.

3 Практическая реализация

Исходный код программы представлен в листинге 1.

Листинг 1: SVD разложение

```
1 import numpy as np
2 from num_methods import *
3
4
5 def svd_decomposition(matrix):
6     covariance_matrix = np.dot(matrix.T, matrix)
7
8     eigenvalues, eigenvectors = np.linalg.eig(covariance_matrix)
9
10    singular_values = np.sqrt(eigenvalues)
11    singular_vectors = eigenvectors
12    left_singular_vectors = matrix.dot(singular_vectors) /
    singular_values
13    right_singular_vectors = singular_vectors
14
15    ns = len(singular_values)
16    s = [[0] * ns for _ in range(ns)]
17    for i in range(ns):
18        s[i][i] = singular_values[i]
19
20    return left_singular_vectors, s, right_singular_vectors.T
21
22
23 matrix = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])
```

```

24 U, S, Vt = svd_decomposition(matrix)
25 print("Result:")
26 print("U:")
27 print_matr(U)
28 print()
29 print("S:")
30 print_matr(S)
31 print()
32 print("Vt:")
33 print_matr(Vt)
34 print()
35
36 res = mult_matr_matr(mult_matr_matr(U, S), Vt)
37 print("Result after multiplication")
38 print_matr(res)
39 print()
40
41 print("Result numpy")
42 U, singular_values, Vt = np.linalg.svd(matrix)
43 ns = len(singular_values)
44 s = [[0] * ns for _ in range(ns)]
45 for i in range(ns):
46     s[i][i] = singular_values[i]
47 print("U:")
48 print_matr(U)
49 print()
50 print("S:")
51 print_matr(s)
52 print()
53 print("Vt:")
54 print_matr(Vt)
55 print()

```

4 Результаты

Результаты работы программы представлены на рисунках 1 – 3.

```

Result:
U:
-0.21483723836839624 0.8872306883463643 1.5155842424895793e-07
-0.5205873894647369 0.2496439529882966 4.2930712448202676e-08
-0.8263375405610777 -0.38794278236977064 -6.829886071304972e-08

S:
16.848103352614213 0 0
0 1.0683695145547163 0
0 0 8.534067506295021e-08

Vt:
-0.4796711778777712 -0.5723677939720625 -0.6650644100663531
-0.776690990321564 -0.0756864701045489 0.6253180501124379
0.4082482904638524 -0.816496580927727 0.4082482904638715

```

Рис. 1 — Результат SVD разложения

```

Result after multiplication
0.9999999999999994 1.9999999999999991 3.00000000000000018
3.9999999999999964 5.0000000000000002 6.0000000000000003
6.999999999999993 8.0000000000000004 9.0000000000000002

```

Рис. 2 — Проверка результата перемножением матриц

5 Выводы

В результате выполнения лабораторной работы было реализовано SVD разложение матрицы, результат был проверен прямым перемножением матриц и через numpy.

```
Result numpy
U:
-0.21483723836839674  0.8872306883463706  0.40824829046386246
-0.5205873894647373  0.24964395298829728  -0.8164965809277261
-0.8263375405610782  -0.3879427823697744  0.40824829046386324

S:
16.84810335261421  0  0
0  1.06836951455471  0
0  0  4.418424751193368e-16

Vt:
-0.47967117787777147  -0.5723677939720624  -0.6650644100663531
-0.7766909903215596  -0.07568647010455848  0.6253180501124427
-0.4082482904638631  0.8164965809277263  -0.4082482904638632
```

Рис. 3 — Результат, полученный через numpy