Соковнин Игорь Леонидович

Линейная алгебра

Урок 5. Сингулярное разложение матриц

Практическое задание

1. Найти с помощью NumPy SVD для матрицы

$$\begin{pmatrix} 1 & 2 & 0\\ 0 & 0 & 5\\ 3 & -4 & 2\\ 1 & 6 & 5\\ 0 & 1 & 0 \end{pmatrix}.$$

2. Для матрицы из предыдущего задания найти:

   а) евклидову норму;

   б) норму Фробениуса.

В [ ]:

Ортогона́льная ма́трица — квадратная матрица F с вещественными элементами, результат умножения которой на транспонированную матрицу F^T равен единичной матрице:
$$FF^T = F^TF = E, FF^T = F^TF = E,$$
или, что эквивалентно, её обратная матрица (которая обязательно существует) равна транспонированной матрице:
$$F^{-1} = F^T.$$

Представление матрицы A в виде

$$A = UDV^T$$

называется *сингулярным разложением (Singular Values Decomposition, SVD)*.

В [ ]:

В [1]:
```python
import numpy as np
np.set_printoptions(precision=2, suppress=True)
```

Задача 1

1. Найти с помощью NumPy SVD для матрицы

$$\begin{pmatrix} 1 & 2 & 0\\ 0 & 0 & 5\\ 3 & -4 & 2\\ 1 & 6 & 5\\ 0 & 1 & 0 \end{pmatrix}.$$

В [2]:
```python
A = np.array([[1, 2, 0],
              [0, 0, 5],
              [3,-4, 2],
              [1, 6, 5],
              [0, 1, 0]])
print(f'Матрица A:\n{A}')
```

```
Матрица A:
[[ 1  2  0]
 [ 0  0  5]
 [ 3 -4  2]
 [ 1  6  5]
 [ 0  1  0]]
```

В [3]:
```python
print(np.dot(A, A.T))
```

```
[[  5   0  -5  13   2]
 [  0  25  10  25   0]
 [ -5  10  29 -11  -4]
 [ 13  25 -11  62   6]
 [  2   0  -4   6   1]]
```

В [4]:
```
U, s, W = np.linalg.svd(A)

# Транспонируем матрицу W
V = W.T

# s - список диагональных элементов, его нужно привести к виду диагональной матрицы для наглядности
D = np.zeros_like(A, dtype=float)
D[np.diag_indices(min(A.shape))] = s
```

В [5]:
```
print(f'Матрица D:\n{D}')
```

```
Матрица D:
[[8.82 0.   0.  ]
 [0.   6.14 0.  ]
 [0.   0.   2.53]
 [0.   0.   0.  ]
 [0.   0.   0.  ]]
```

В [6]:
```
print(f'Матрица U:\n{U}')
```

```
Матрица U:
[[ 0.17  0.16 -0.53 -0.8  -0.16]
 [ 0.39 -0.53  0.61 -0.43  0.03]
 [-0.14 -0.82 -0.52  0.14  0.07]
 [ 0.89  0.06 -0.25  0.38 -0.06]
 [ 0.08  0.11 -0.08 -0.11  0.98]]
```

В [7]:
```
# Убедимся, что она действительно ортогональна
print(np.dot(U.T, U))
```

```
[[ 1.  0. -0. -0. -0.]
 [ 0.  1.  0. -0.  0.]
 [-0.  0.  1. -0. -0.]
 [-0. -0. -0.  1. -0.]
 [-0.  0. -0. -0.  1.]]
```

В [8]:
```
print(f'Матрица V:\n{V}')
```

```
Матрица V:
[[ 0.07 -0.37 -0.93]
 [ 0.72  0.67 -0.21]
 [ 0.69 -0.65  0.31]]
```

В [9]:
```
# Убедимся, что она действительно ортогональна
print(np.dot(V.T, V))
```

```
[[ 1.  0.  0.]
 [ 0.  1. -0.]
 [ 0. -0.  1.]]
```

В [10]:
```
# Проведем проверку
print(np.dot(np.dot(U, D), V.T))
```

```
[[ 1.  2.  0.]
 [ 0. -0.  5.]
 [ 3. -4.  2.]
 [ 1.  6.  5.]
 [-0.  1.  0.]]
```

Задача 2

2. Для матрицы из предыдущего задания найти:

   а) евклидову норму;

   б) норму Фробениуса.

а) Найти евклидову норму

Евклидова норма матрицы равна евклидовой норме диагональной матрицы из ее сингулярных чисел D. Максимальное значение полученного отношения будет равно максимальному сингулярному числу μ_{max}, и, принимая во внимание факт сортировки по убыванию сингулярных чисел, получим

$$\|A\|_E = \mu_1.$$

В [11]:
```
n_ev=D[0, 0]
print(f'Евклидова норама матрицы A:\n{n_ev}')
```

```
Евклидова норама матрицы A:
8.824868854820442
```

б) Найти норму Фробениуса

В случае, когда известно сингулярное разложение матрицы, ее норма Фробениуса вычисляется как

$$\|A\|_F = \sqrt{\sum_{k=1}^{r} \mu_k^2}.$$

Диагональные элементы матрицы D имели вид

$$\mu_1 \geqslant \mu_2 \geqslant \ldots \geqslant \mu_r > \mu_{r+1} = \ldots = \mu_n = 0,$$

В [12]:
```
n_f=np.linalg.norm(A, ord=None, axis=None, keepdims=False)
print(f'Норма Фрабениуса матрицы A:\n{n_f}')
```

```
Норма Фрабениуса матрицы A:
11.045361017187261
```

В [13]:
```
# Проверка
(D[0,0]**2+D[1,1]**2+D[2,2]**2)**(1/2)
```

Out[13]:
```
11.045361017187261
```

1. numpy.linalg.norm - https://pyprog.pro/linear_algebra_functions/linalg_norm.html (https://pyprog.pro/linear_algebra_functions/linalg_norm.html)