

BÀI THỰC HÀNH LAB 3

Giảng viên: TS. Dương Việt Hằng

Trợ giảng: Trần Hà Sơn

Ngày 2 tháng 5 năm 2023

Mục tiêu:

- Nắm vững được Python để viết được các đoạn lệnh.
- Nâng cao kỹ năng lập trình cùng với tư duy toán học.
- Hiểu và tính toán được phân tích kỳ dị với một số thư viện phổ biến.

1 Sử dụng thư viện NumPy để phân tích kỳ dị

Ví dụ 1. Cho ma trận $A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$. Sử dụng thư viện Numpy, code phân tích ma trận A như sau:

```
1 import numpy as np
2
3 #Khoi tao ma tran A
4 A = np.array([[1,0,1, 0],[0,1,0,1]])
5
6 #Thuc hien phan tich ky di ma tran A
7 # U la ma tran ky di trai
8 # VT la ma tran chuyen vi cua ma tran ky di phai
9 # D la ma tran duong cheo
10 U, D, VT = np.linalg.svd(A)
```

Lần lượt thực hiện các lệnh:

```
1 print(U)
```

Kết quả là

```
1 [[1.  0.]
2  [0.  1.]]
```

```
1 print(D)
```

Kết quả là

```
1 [1.41421356 1.41421356]
```

```
1 print(VT)
```

Kết quả là

```
1 [[ 0.70710678  0.          0.70710678  0.          ]
2 [-0.          0.70710678  0.          0.70710678]
3 [-0.70710678  0.          0.70710678  0.          ]
4 [ 0.          -0.70710678  0.          0.70710678]]
```

2 Phân tích kỳ dị với thư viện SciPy

Vấn với ma trận A ở trên, sử dụng thư viện SciPy để phân tích kỳ dị ma trận A ta cũng được kết quả tương tự.

```
1 import scipy.linalg
2 U, D, VT = scipy.linalg.svd(A)
```

3 Đồ án

Định nghĩa 1. Cho ma trận $M \in M_{m \times n}(\mathbb{R})$ và có phân tích kỳ dị là $M = U\Sigma V^T$ (full SVD hoặc reduced SVD), khi đó ma trận giả nghịch đảo của M được định nghĩa như sau:

$$M^+ = V\Sigma^+U^T,$$

trong đó Σ^+ là ma trận được xây dựng từ ma trận Σ bằng cách chuyển vị Σ rồi lấy nghịch đảo tất cả các phần tử khác 0 nằm trên đường chéo chính của Σ .

Ví dụ 2. Nếu $\Sigma = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ thì $\Sigma^+ = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Ta có ứng dụng của SVD trong việc tìm nghiệm bình phương tối thiểu từ định lý sau:

Định lý 1. Cho ma trận $A \in M_{m \times n}(\mathbb{R})$ và ma trận $b \in \mathbb{R}^m$, khi đó

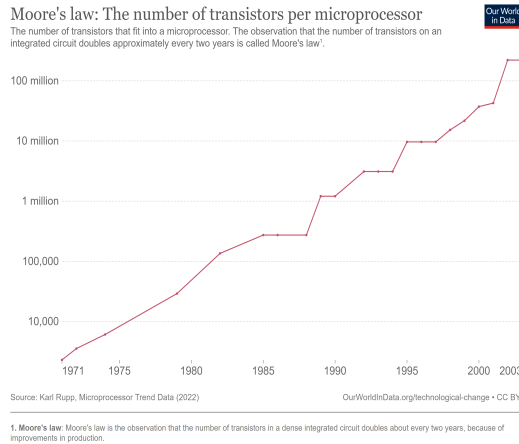
$$\bar{x} = A^+b$$

là nghiệm bình phương tối thiểu của $Ax = b$.

Áp dụng định lý 1, giải bài toán sau bằng phân tích kỳ dị.

Bài 1. Hình sau cho biết số bóng bán dẫn N trong 13 bộ vi xử lý cùng với năm chúng ra đời.

Năm	Số bóng bán dẫn
1971	2 250
1972	2 500
1974	5 000
1978	29 000
1982	120 000
1985	275 000
1989	1 180 000
1993	3 100 000
1997	7500 000
1999	24 000 000
2000	42 000 000
2002	220 000 000
2003	410 000 000



- a) Dùng phương pháp phân tích kỳ dị, tìm nghiệm bình phương tối thiểu, từ đó xác định đường thẳng khớp với mô hình

$$\log_{10} N \approx \theta_1 + \theta_2(t - 1970).$$

- b) Với mô hình tính được ở câu (a), hãy dự đoán số bóng bán dẫn trong bộ vi xử lý được giới thiệu vào năm 2015. So sánh dự đoán với bộ vi xử lý IBM Z13 ra đời vào năm 2015 có số bóng bán dẫn khoảng 4×10^9 .

3.1 Quy định bài nộp

- Thực hiện toàn bộ bài làm trên một tập tin Python (.py) hoặc một tập tin Jupyter Notebook (.ipynb).
- Nộp tập tin **MSSV.zip** được nén từ thư mục **MSSV** chứa các tập tin sau:
 - Báo cáo toàn bộ bài làm: **MSSV.pdf**.
Nội dung báo cáo gồm có: Thông tin cá nhân (họ và tên, mã số sinh viên); ý tưởng thực hiện, mô tả các hàm.
 - Mã nguồn: **MSSV.py** hoặc **MSSV.ipynb**.

3.2 Quy định chấm bài

Những trường hợp sau sẽ bị 0 điểm toàn bộ đề án:

- Nộp sai quy định.
- Không có báo cáo.
- Thực thi mã nguồn báo lỗi.

LƯU Ý: SAO CHÉP BÀI LÀM CỦA NHAU SẼ BỊ 0 ĐIỂM TOÀN BỘ PHẦN THỰC HÀNH.