TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC

KHAI TRIỂN KÌ DỊ CỦA MA TRẬN VÀ ỨNG DỤNG

BÁO CÁO
GIẢI TÍCH SỐ

Giảng viên hướng dẫn:	TS. Hà Thị Ngọc Yến
Sinh viên thực hiện:	Nguyễn Quang Huy
MSSV:	20173179
Lớp:	KSTN Toán Tin K62

HÀ NỘI, 1/2022

Mục lục

LỜI MỞ ĐẦU

Phương pháp khai triển giá trị kì dị (Singular Value Decomposition) được viết tắt là SVD là một trong những phương pháp thuộc nhóm matrix factorization được phát triển lần đầu bởi những nhà hình học vi phân. Ban đầu mục đích của phương pháp này là tìm ra một phép xoay không gian sao cho tích vô hướng của các vector không thay đổi. Từ mối liên hệ này khái niệm về ma trận trực giao đã hình thành để tạo ra các phép xoay đặc biệt. Phương pháp SVD đã được phát triển dựa trên những tính chất của ma trận trực giao và ma trận đường chéo để tìm ra một ma trận xấp xỉ với ma trận gốc. Phương pháp này sau đó đã được ứng dụng rộng rãi trong các lĩnh vực như hình học vi phân, hồi qui tuyến tính, xử lý hình ảnh, các thuật toán nèn và giảm chiều dữ liệu, và trong các bài toán recommendation.

Chương 1

Kiến thức chuẩn bị

1.1 Trị riêng và vector riêng của ma trận

Cho $A = (a_{ij})$ là ma trận vuông kích thước $n \times n$. Nếu $Au = \lambda u$, khi đó λ và u tương ứng được gọi là giá trị riêng và vector riêng của ma trận A.
Giá trị riêng của A là nghiệm của phương trình :

$$\det(\lambda I_n - A) = 0$$

Vector riêng λ của A là nghiệm của hệ:

$$(\lambda I_n - A)X = 0$$

1.2 Hệ trực chuẩn và hệ trực giao

Hệ trực chuẩn là một hệ cơ sở $u_1, u_2, \ldots, u_n \in \mathbb{R}^n$ được gọi là hệ trực chuẩn nếu:

$$\begin{cases} u_i^T u_j = 0 & \forall 1 \leq i \neq j \leq n \\ u_i^T u_j = 1 \end{cases}$$

Ma trận trực giao $U = [u_1, u_2, \ldots, u_n] \in \mathbb{R}^{n \times n}$ là một ma trận trực giao nếu $u_1, u_2, \ldots, u_n$ là 1 hệ trực chuẩn.
Nếu U là ma trận trực giao thì:

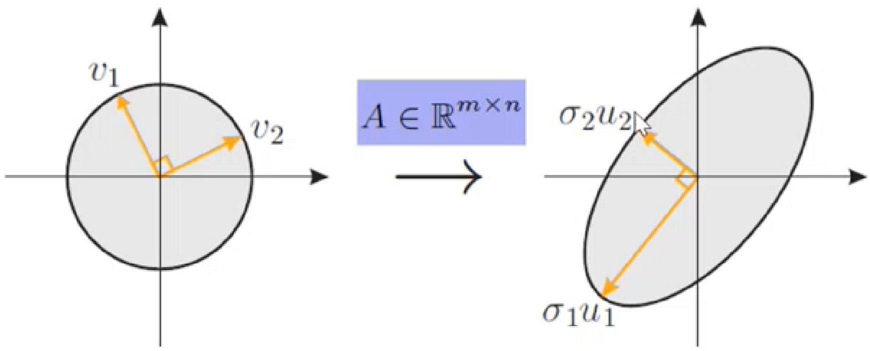
- $U^T U = I$
- U^T là ma trận trực giao
- $U^T = U^-1$

Chương 2

Giá trị kì dị và vector kì dị

2.1 Ánh xạ hình cầu đơn vị

2.1.1 Sự tồn tại của khai triển kì dị



$$S = \left\{ x \in \mathbb{R}^n \mid \|x\| \le 1 \right\} \qquad AS = \left\{ Ax \mid x \in S \right\}$$

Ta xét một phép biến đổi tuyến tính hoặc một ánh xạ tương ứng với ma trận A đi từ không gian $\mathbb{R}^n$ vào không gian $\mathbb{R}^m$ được định nghĩa như ở hình minh họa phía trên.
Khi ta tác động A vào vector v_1 với $||v_1|| = 1$ ta được vector u_1. Tương tự trong không gian $\mathbb{R}^n$ ta hoàn toàn có thể tìm được một vector v_2 sao cho $v_2 \perp v_1$ và $||v_2|| = 1$, và tác động ma trận A vào vector v_2 ta thu được vector u_2 và $u_2 \perp u_1$.
Vì $v \in \mathbb{R}^n$ nên để tìm được đầy đủ các vector trực giao còn lại của $\mathbb{R}^n$ bằng phương pháp trực chuẩn Gram-Schmidt đã học trong môn Đại số.
Khi đó, với $v_1, v_2, \cdots, v_n$ là một hệ cơ sở trực chuẩn trong $\mathbb{R}^n$, ta hoàn toàn có thể tìm được m vector trực giao trong $\mathbb{R}^m$ bằng phép tính:

$$Av_i = u_i$$

Để u_i là một vector trực giao, ta hoàn toàn có thể nhân vector u_i với một giá trị σ_i nào đó để thỏa mãn u_i là một vector trực chuẩn.
Hay:

$$Av_i = \sigma_i u_i$$

Ta có thể viết lại công thức trên dưới dạng ma trận:

$$A \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_n \end{bmatrix}$$

Hay

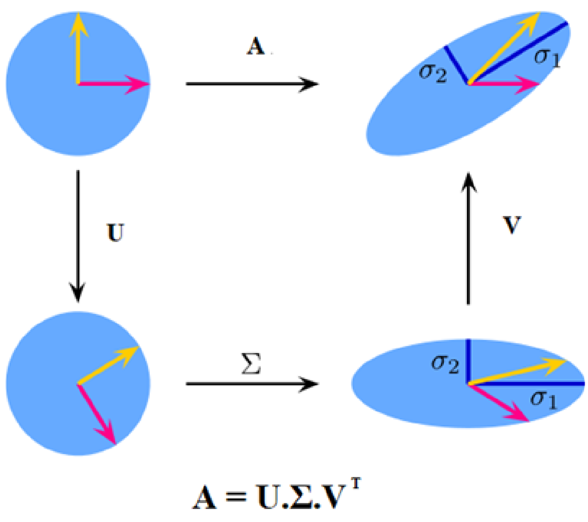
$$AV = U\Sigma$$

Vì V là ma trận vuông gồm các cơ sở trực giao nên $\exists V^{-1}$.
Do $V^T = V^{-1}$ nên:

$$A = U\Sigma V^T$$

Vì vậy với mọi ma trận A, ta đều có thể phân tích ma trận A về dạng như trên và được gọi là khai triển kì dị của ma trận A.

2.1.2 Bản chất của khai triển kì dị



Về mặt hình học phép phân tích suy biến sẽ lần lượt trải qua:

- Phép xoay (rotation)
- Phép nới rộng (scaling)
- Phép xoay

Nếu ta coi các dòng của ma trận A là các điểm dữ liệu và các chiều dữ liệu là các cột. Khi đó nhân các điểm dữ liệu với ma trận trực giao U chính là việc ta thực hiện một phép xoay và phép xoay này không làm thay đổi tích vô hướng của các vector. Phép nhân với ma trận đường chéo Σ sẽ co dãn độ lớn các chiều theo giá trị của các trị riêng trên đường chéo chính. Và cuối cùng phép nhân với ma trận trực giao V^T lại thực hiện phép biến xoay một lần nữa.

2.2 Biểu diễn ma trận A thông qua các vector và giá trị kì dị

Xét ma trận $A_{m\times n}$ với $m > n$ và A là ma trận có hạng đầy đủ hay rank$A = n$.
Với

$$A = U\Sigma V^T$$

Trong đó:

$$U = [u_1 \quad u_2 \quad \dots \quad u_n]$$

$$V = [v_1 \quad v_2 \quad \dots \quad v_n]$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}$$

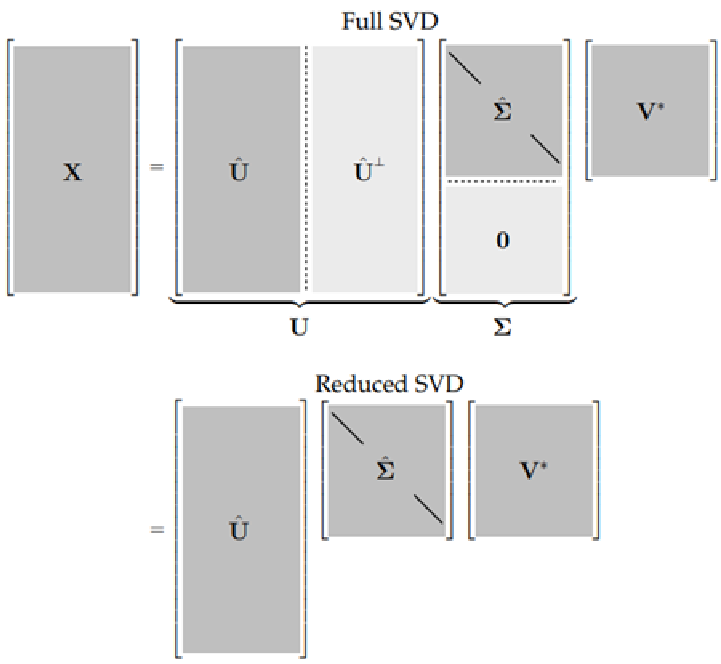
Với:

- $u_1, u_2, \dots, u_n$ lần lượt là các vector kì dị trái của A
- $v_1, v_2, \dots, v_n$ lần lượt là các vector kì dị phải của A
- $\sigma_1, \sigma_2, \dots, \sigma_n$ lần lượt là các giá trị kì dị của A

Khi đó:

$$
\begin{aligned}
A &= U\Sigma V^T \\
&= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \begin{bmatrix} v_1 & v_2 & \dots & v_n \end{bmatrix}^T \\
&= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} \\
&= \begin{bmatrix} u_1 & u_2 & \dots & u_n \end{bmatrix} \begin{bmatrix} \sigma_1 v_1^T \\ \sigma_2 v_2^T \\ \vdots \\ \sigma_n v_n^T \end{bmatrix} \\
&= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_n u_n v_n^T
\end{aligned}
$$

Đây là biểu thức rút gọn của ma trận A dưới dạng tích của các giá trị kì dị và các vector kì dị của A với dạng hạng đầy đủ.



Trong trường hợp ma trận có hạng đầy đủ như ma trận A, vector cột từ vị trí $n+1$ đến ứng với phần ma trận $U^{\perp}$ ta có thể bỏ qua và không cần thêm vào ma trận U. Bởi vì:

$$
\begin{bmatrix} U & U^{\perp} \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix} = \begin{bmatrix} U.\Sigma + U^{\perp}.0 \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix} = U\Sigma V^T
$$

Tương tự, ta xét ma trận $A_{m\times n}$ với $m > n$ và A là ma trận có hạng không đầy đủ hay rank$A = r < n$.
Trong trường hợp hạng không đầy đủ, thì từ vector cột thứ $r+1$ trở đi, u_i và v_i là các vector không và $\sigma_i = 0$ với $i > r+1$. Khi đó

$$A = U\Sigma V^T$$

Trong đó:

$$\begin{aligned} U &= [u_1 \quad u_2 \quad \dots \quad u_r] \\ V &= [v_1 \quad v_2 \quad \dots \quad v_r] \\ \Sigma &= \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} \end{aligned}$$

Với:

- $u_1, u_2, \dots, u_r$ lần lượt là các vector kì dị trái của A
- $v_1, v_2, \dots, v_r$ lần lượt là các vector kì dị phải của A
- $\sigma_1, \sigma_2, \dots, \sigma_r$ lần lượt là các giá trị kì dị của A

Khi đó:

$$\begin{aligned} A &= U\Sigma V^T \\ &= [u_1 \quad u_2 \quad \dots \quad u_r] \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r \end{bmatrix} [v_1 \quad v_2 \quad \dots \quad v_r]^T \\ &= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_r u_r v_r^T \end{aligned}$$

Đây là biểu thức rút gọn của ma trận A dưới dạng tích của các giá trị kì dị và các vector kì dị của A với dạng hạng không đầy đủ.

2.3 Tìm giá trị kì dị và các giá trị kì dị

Như đã nói ở phía trước, mọi ma trận A đều có thể khai triển dưới dạng tích của 3 ma trận đặc biệt như sau:

$$A = U\Sigma V^T$$

Ta xét ma trận sau:

$$\begin{aligned} AA^T &= U\Sigma V^T (U\Sigma V^T)^T \\ &= U\Sigma V^T V \Sigma^T V \\ &= U\Sigma\Sigma^T U^T \\ &= U\Sigma\Sigma^T U^{-1} \end{aligned}$$

Do U, V là các ma trận trực giao nên $U^T = U^{-1}$ và $V^T V = I$

Quan sát thấy rằng $\Sigma\Sigma^T$ là một ma trận đường chéo với các phần tử trên đường chéo chính là $\sigma_1^2, \sigma_2^2, \ldots$

Vậy $AA^T = U\Sigma\Sigma^T U^{-1}$ chính là khai triển giá trị riêng của ma trận AA^T. Hơn nữa $\sigma_1^2, \sigma_2^2, \ldots$ chính là các giá trị riêng của AA^T.

Vì AA^T luôn là ma trận nửa xác định dương nên các giá trị riêng của ma trận AA^T là không âm, khi đó $\sigma_1, \sigma_2, \ldots$ sẽ là các giá trị thực mà các giá trị này chính là các giá trị kì dị của ma trận A.

Cũng theo đó, mỗi cột của U chính là một vector riêng của AA^T. Ta gọi mỗi cột này là các vector kì dị trái của A.

Tương tự như thế, $A^T A = V\Sigma^T \Sigma V^T$ và các cột của V còn được gọi là các vector kì dị phải của A.

Ta không nhất thiết phải tìm ma trận U. Ta có:

$$\begin{aligned}
A &= U\Sigma V^T \\
\leftrightarrow AV &= U\Sigma \\
\leftrightarrow Av_i &= u_i\sigma_i \\
\leftrightarrow \frac{1}{\sigma_i} Av_i &= u_i
\end{aligned}$$

Từ đó các cột của ma trận U có thể tính được từ các giá trị kì dị và các vector kì dị phải của A

Chương 3

Ứng dụng của giá trị kì dị

3.1 Nén ảnh

Phương pháp nén ảnh hay còn gọi là giảm chiều dữ liệu hình ảnh. Giả sử ta có một tập dữ liệu nhiều ảnh có kích thước rất lớn. Khả năng lưu trữ của chúng ta là có hạn. Trong tình huống này, ta làm thể nào để giảm kích thước của bộ ảnh vừa với dung lượng vừa đủ mà thông tin của các bức ảnh vẫn giữ được một lượng lớn.

Trong nội dung của bài báo cáo này, ta chỉ xét đến ảnh đen trắng là một ma trận có kích thước là $m \times n$ với mỗi giá trị của ma trận là một phần tử nằm trong đoạn $[0, 255]$.

Coi $A_{m\times n}$ là ma trận biểu diễn cho hình ảnh và A có các giá trị kì dị là $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$. Khi ta muốn nén bức ảnh đó lại, ta chỉ cần giữ 1 lượng nhỏ các giá trị σ_i mang giá trị lớn, còn các giá trị còn lại thường là nhỏ và gần đến 0 và ta có thể lược bỏ các giá trị đó đi.

Khi đó ta có thể xấp xỉ ma trạn A bằng tổng của $k < r$ ma trận:

$$A \cong A_k = U_k \Sigma_k V_k^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T$$

Dưới đây là định lý Frobineous nói rằng sai số do cách xấp xỉ trên chính là căn bậc hai của tổng bình phương của các giá trị kì dị mà ta đã bỏ qua ở phần cuối của ma trận Σ. Ở đây sai số được định nghĩa là chuẩn Frobineous của hiệu hai ma trận:

Định lý Frobenious:

$$||A - A_k||_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Với A_k là ma trận không thì:

$$||A||_F^2 = \sum_{i=1}^{r} \sigma_i^2$$

Một cách khác để tính chuẩn Frobenious của ma trận như sau:

$$||A - A_k||_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} \sigma_{ij}^2 = ||A||_2^2 = \mathrm{tr}(AA^T)$$

Trong đó $\text{tr}(AA^T)$ được gọi là vết của ma trận được định nghĩa như sau:

$$\text{tr}(AA^T) = \sum_i a_{ii}$$

Ta có tính chất sau: với A, B là 2 ma trận bất kì thì:

$$\text{tr}(AB) = \text{tr}(BA)$$

Chứng minh:

Với $\bar{A}_i, B_j$ lần lượt là vector dòng thứ i của ma trận A và cột thứ j của ma trận B. Khi đó ta có phần tử $(AB)_{ij}$ của ma trận AB là:

$$(AB)_{ij} = \bar{A}_i B_j = \sum_k (a_{ik} b_{kj})$$

$$\rightarrow \text{tr}(AB) = \sum_i (AB)_{ii} = \sum_i \sum_k (a_{ik} b_{ki})$$

Biến đổi tương tự, ta được

$$\text{tr}(BA) = \sum_i (BA)_{ii} = \sum_i \sum_k (b_{ik} a_{ki})$$

Ta nhận thấy chỉ số i, k bình đẳng trong cả 2 biểu thức trên nên nếu hoán vị i và k cho nhau không làm thay đổi kết quả tổng.
Mặt khác phép hoán vị này sẽ biến biểu thức trên thành dưới nên suy ra giá trị của 2 biểu thức là bằng nhau.

Khi đó:

$$\begin{aligned} ||A||_F^2 &= \text{tr}(^T) \\ &= \text{tr}(U\Sigma\Sigma^T U^T) \\ &= \text{tr}(\Sigma\Sigma^T U U^T) \\ &= \text{tr}(\Sigma\Sigma) \\ &= \sum_{i=1}^{r} \sigma_i^2 \end{aligned}$$

Vậy ta được:

$$||A - A_k||_F^2 = \sum_{i=k+1}^{r} \sigma_i^2$$

Từ đó, ta xét:

$$\frac{||A - A_k||_F^2}{||A||_F^2} =$$

Như vậy, sai số do xấp xỉ càng nhỏ nếu phần giá trị kì dị bị lược bỏ có giá trị càng nhỏ so với phần giá trị kì dị được giữ lại. Đây là một định lý quan trọng giúp xác định việc xấp xỉ ma trận dựa trên lượng thông tin muốn giữ lại.

Ví dụ, nếu ta muốn giữ lại ít nhất 90% lương thông tin trong A, trước hết ta tính $\sum_{j=1}^{r} \sigma_J^2$ sau đó chọn k là số nhỏ nhất sao cho:

$$\frac{\sum_{i=k+1}^{r} \sigma_i^2}{\sum_{j=1}^{r} \sigma_j^2} \geq 0.9$$

Khi k nhỏ, ma trận A_k có hạng là k, là một ma trận có rank nhỏ. Vì vậy, phương pháp nén ảnh này còn được coi là một phương pháp xấp xỉ ma trận với hạng nhỏ hơn.

3.2 Nghịch đảo suy rộng

Như ta đã biết, mọi ma trận A vuông không suy biến đều tồn tại ma trận nghịch đảo, vậy với ma trận A không vuông thì nó có tồn tại ma trận nghịch đảo không? Trong phần này ta sẽ tập trung làm rõ điều đó.
Giả sử A là ma trận với khai triển kì dị là $A = U\Sigma V^T$
Khi đó ma trận $A^\dagger = V\Sigma^{-1}U^T$ được gọi là ma trận nghịch đảo suy rộng của A.

Có hai loại ma trận nghịch đảo suy rộng:

- Ma trận nghịch đảo suy rộng trái:

$$A^\dagger A = I$$

- Ma trận nghịch đảo suy rộng phải:

$$AA^\dagger = I$$

Xét một ma trận $A_{m\times n}$, khi đó ma trận A sẽ xảy ra 2 trường hợp.

- A là ma trận hình chữ nhật nằm hay $n > m$ và ta chỉ xét trường hợp ma trận A có hạng đầy đủ tức là $\text{rank}(A) = m$, Xét:

$$(A^TA)_{m\times m} \text{ có hạng } = m \rightarrow \exists (A^TA)^{-1}$$

$$(AA^T)_{n\times n} \text{ có hạng} = m \rightarrow \text{không } \exists (A^TA)^{-1}$$

  Khi đó: $A^\dagger$ là ma trận nghịch đảo suy rộng phải và được xác định bởi:

$$A^\dagger = A^T(AA^T)^{-1}$$

- A là ma trận hình chữ nhật đứng hay $m > n$ và ta cũng chỉ xét trường hợp ma trận A có hạng đầy đủ tức là $\text{rank}(A) = n$. Tương tự như phía trên, ta sẽ có $A^\dagger$ là ma trận nghịch đảo suy rộng trái và mà trận $A^\dagger$ được xác định bởi:

$$A^\dagger = (A^TA)^{-1}A^T$$

Ứng dụng của nghịch đảo suy rộng vào giải hệ phương trình:

$$Ax = y$$

Khi $n > m$ thì $A^\dagger$ là ma trận nghịch đảo suy rộng phải của A, do đó trong trường hợp này ma trận nghịch đảo suy rộng sẽ không có tác dụng.

Khi $m > n$ thì $A^\dagger$ sẽ là ma trận nghịch đảo suy rộng trái của A. Như vậy:

$$\begin{aligned} A^\dagger A x &= A^\dagger y \\ x &= A^\dagger y \end{aligned}$$

Xét hệ sau:

$$\begin{cases} x_1 + 3x_2 &= 17 \\ 5x_1 + 7x_2 &= 19 \\ 11x_1 + 13x_2 &= 23 \end{cases}$$

Ta có:

$$A = \begin{bmatrix} 1 & 3 \\ 5 & 7 \\ 11 & 13 \end{bmatrix}, y = \begin{bmatrix} 17 \\ 19 \\ 23 \end{bmatrix}$$

Trong thực tế, khi giải hệ phương trình trên, ta chắc chắn một điều rằng hệ phương trình trên cho kết quả là vô nghiệm. Nhưng vì A là tồn tại ma trận nghịch đảo suy rộng hay trong thực nghiệm thì sẽ cho ta một kết quả cụ thể.

Thực chất, ma trận nghịch đảo suy rộng không giải chính xác bài toán $Ax = y$ mà là lời giải cho bài toán bình phương tối thiểu. Thông thường sẽ không có lời giải chính xác cho bài toán $Ax = y$ với A là ma trận hình chữ nhật đứng.

Xét bài toán bình phương tối thiểu:
Với mỗi $y = [y_1, y_2, \dots, y_n], A = [\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n]$, xét hàm:

$$\begin{aligned} L(x) &= \frac{1}{2}\sum_{i=1}^{n}(\bar{A}_i x - y_i)^2 \\ &= \frac{1}{2}||Ax - y||^2 \end{aligned}$$

Như vậy: $\min L(x) = \min \frac{1}{2}||Ax - y||^2$

Xét:

$$\begin{aligned} L(x) &= \frac{1}{2}||Ax - y||^2 \\ \rightarrow \frac{\partial L(x)}{\partial x} &= A^T(Ax - y) \end{aligned}$$

Cho $\frac{\partial L(x)}{\partial x} = 0$, ta được

$$\begin{aligned} L(x) &= 0 \\ \leftrightarrow A^T(Ax - y) &= 0 \\ \leftrightarrow A^T A x &= A^T y \\ \leftrightarrow x &= (A^T A)^{-1} A^T y \\ \leftrightarrow x &= A^\dagger y \end{aligned}$$

Do đó $A^\dagger$ là nghiệm của bài toán $\min L(x)$ và cũng là nghiệm duy nhất của phương trình

3.3 Số điều kiện của ma trận

Trong lĩnh vực giải tích số, số điều kiện của ma trận đo lường giá trị đầu ra của phương trình $Ax = y$ có thể thay đổi bao nhiêu đối với một thay đổi nhỏ trong đối số đầu vào. Điều này được sử dụng để đo lường mức độ nhạy cảm của một ma trận đối với các thay đổi hoặc sai số trong đầu vào và mức độ sai số trong đầu ra.

Với A là ma trận khả nghịch. Xét phương trình:

$$Ax = y$$

Trong đó:

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 0.001 \end{bmatrix}$$

Xét hai trường hợp:

- $y = [2; 2]$
  Ta có:
  $$\begin{bmatrix} 1 & 1 \\ 1 & 0.001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$
- $y = [2; 2.001]$
  Ta có:
  $$\begin{bmatrix} 1 & 1 \\ 1 & 0.001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2.001 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Từ 2 trường hợp nêu trên, ta có nhận xét rằng y có sự thay đổi nhỏ và dẫn đến sự thay đổi lớn của x. Ta đưa ra 2 khái niệm sau:
Ma trận A được gọi là ma trận tốt nếu có sự thay đổi nhỏ ở y dẫn đến sự thay đổi nhỏ ở x và ngược lại ma trận xấu là ma trận nếu có sự thay đổi nhỏ ở y dẫn đến thay đổi lớn ở nghiệm x.

$$
\begin{aligned}
A(x+\delta x) &= y+\delta y \\
\leftrightarrow Ax + A\delta x &= y+\delta y \\
\leftrightarrow A\delta x &= \delta y \\
\leftrightarrow \delta x &= A^{-1}\delta y \\
\rightarrow ||\delta x|| &= ||A^{-1}\delta y|| \\
\rightarrow ||\delta x|| &\le ||A^{-1}||||\delta y||
\end{aligned}
$$

Lại có:

$$
\begin{aligned}
||y|| &\le ||A||||x|| \\
\rightarrow \frac{1}{||x||} &\le \frac{||A||}{||y||} \\
\rightarrow \frac{||\delta x||}{||x||} &\le ||A||||A^{-1}||\frac{||\delta y||}{||y||}
\end{aligned}
$$

Khi đó số điều kiện của A được xác định bởi:

$$\text{cond}(A) = ||A||||A^{-1}||$$

Mà:

$$
\begin{aligned}
||Ax|| &= ||U\Sigma V^T x|| \\
\leftrightarrow ||Ax|| &= ||\Sigma x|| \\
\leftrightarrow ||Ax|| &\le \sigma_{max}||x||
\end{aligned}
$$

Như vậy: $||A|| = \sigma_{max}$
Tương tự ta tính được $||A^{-1}|| = \frac{1}{\sigma_{min}}$
Hay số điều kiện của A được xác định bởi:

$$\text{cond}(A) = ||A||||A^{-1}|| = \frac{\sigma_{max}}{\sigma_{min}}$$

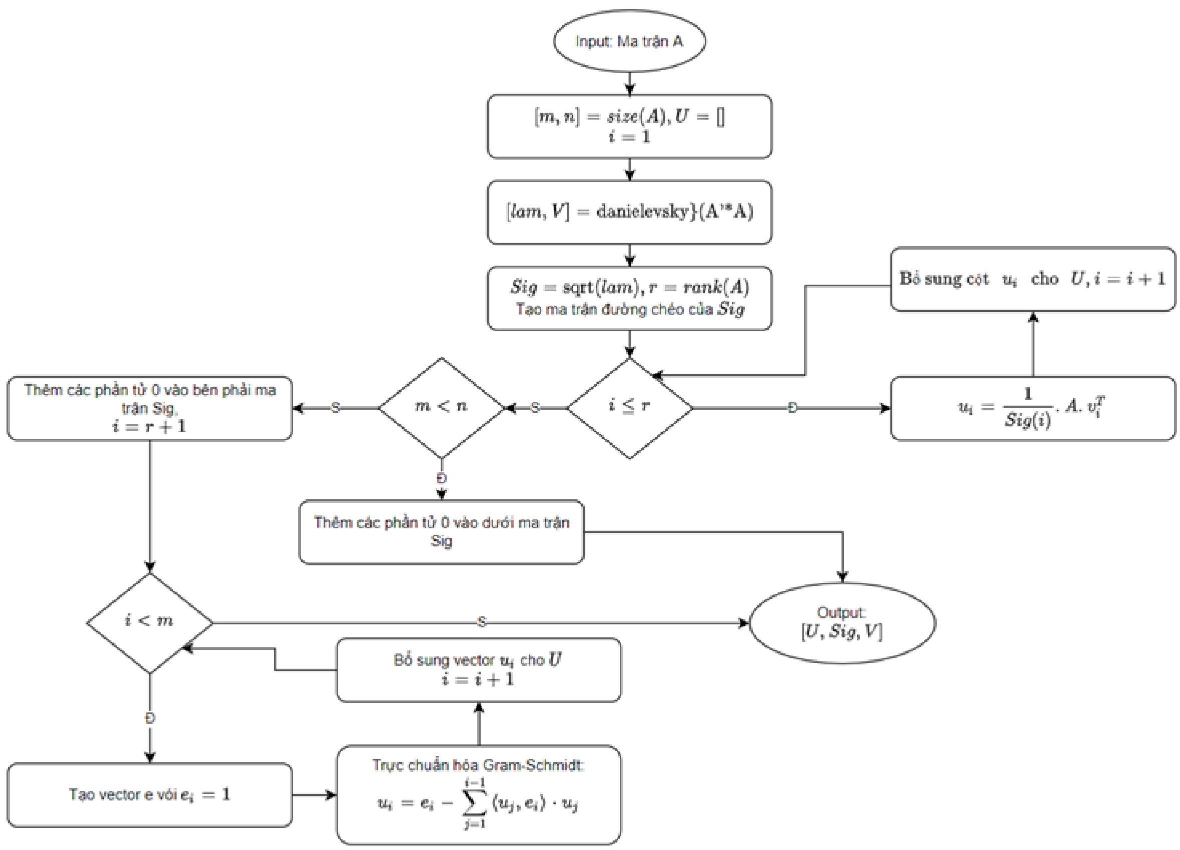
Chương 4

Thuật toán và các ví dụ

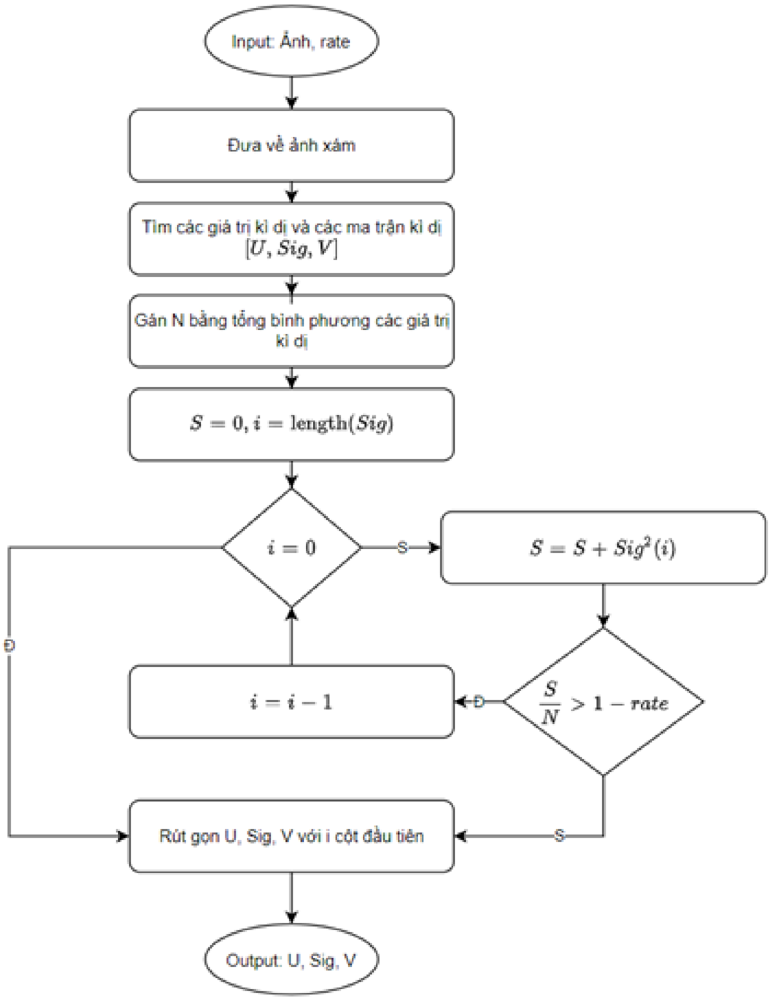
4.1 Thuật toán

Áp dụng các kiến thức lý thuyế đã được trình bày ở phía trước, ta xây dựng các thuật thuật toán như sau:

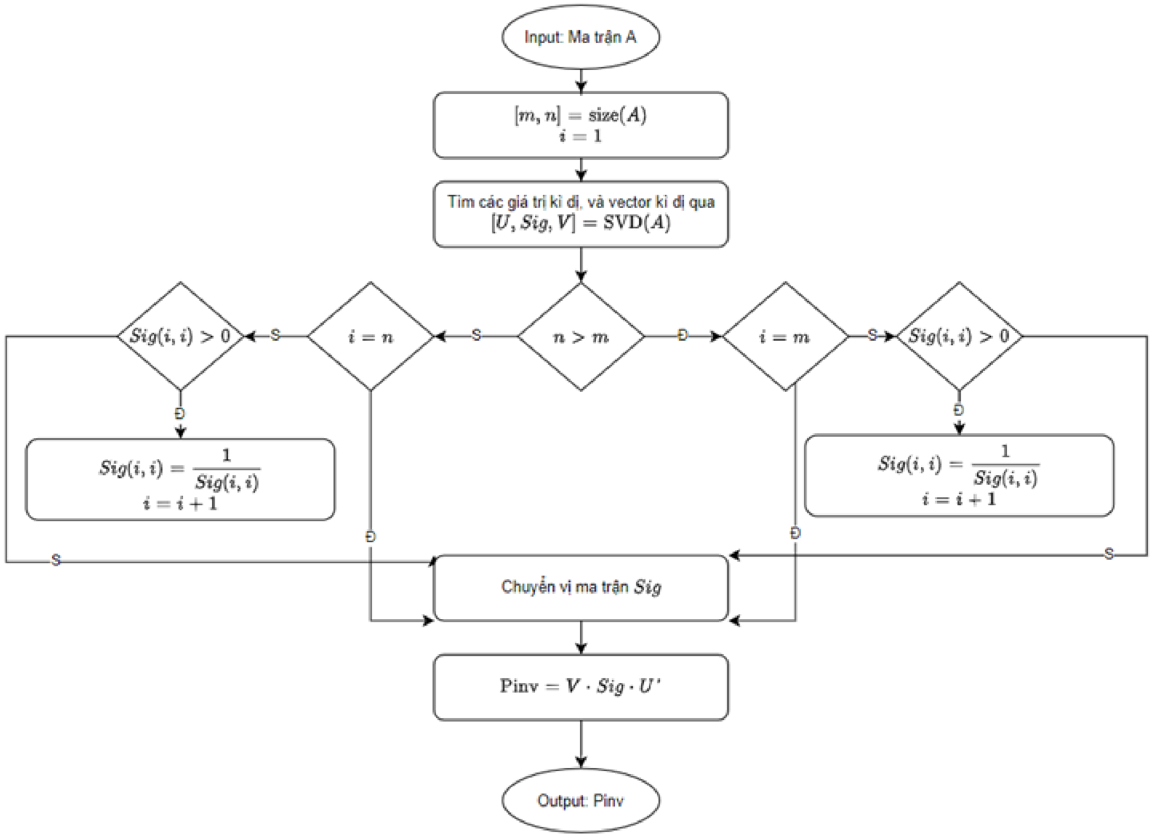
4.1.1 Thuật toán tìm giá trị riêng và vector riêng của ma trận



4.1.2 Thuật toán nén ảnh



4.1.3 Thuật toán tìm ma trận nghịch đảo suy rộng



4.2 Ví dụ minh họa

Ví dụ 4.2.1

Tìm các giá trị kì dị và vector kì dị của ma trận sau:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Ta thu được kết quả:

```
>> [U, Sig, V] = SVD(A)

U =

    1.0000         0
         0    1.0000


Sig =

    1.4142         0         0
         0    1.0000         0


V =

    0.7071         0   -0.7071
    0.7071         0    0.7071
         0    1.0000         0
```

Ví dụ 4.2.2

Tìm các giá trị kì dị và vector kì dị của ma trận sau:

$$G = \begin{pmatrix} 2 & -3 & 6 & 2 & 5 \\ -2 & 3 & -3 & -3 & -4 \\ 4 & -6 & 9 & 5 & 9 \\ -2 & 3 & 3 & -4 & 1 \end{pmatrix}$$

Ta thu được kết quả:

```
G =

     2    -3     6     2     5
    -2     3    -3    -3    -4
     4    -6     9     5     9
    -2     3     3    -4     1

>> [U, Sig, V] = SVD(G)

U =

   -0.0152   -0.0232   -0.0015   -0.0000
    0.0152    0.0254   -0.0016   -0.0000
   -0.0304   -0.0486    0.0001    0.0000
    0.0152    0.0298    0.0006    0.0000


Sig =

   18.9984         0         0         0         0
         0    6.4529         0         0         0
         0         0    0.6485         0         0
         0         0         0    0.0000         0


V =

    0.7849   -0.8569   -0.8326   -0.8296    0.0020
    0.6196   -0.5154   -0.5539   -0.5583   -0.6574
    0.0000    0.0032   -0.0011   -0.0023   -0.2928
    0.0000   -0.0035    0.0003   -0.0052   -0.6587
    0.0000    0.0011    0.0017    0.0017    0.2196
```

Ví dụ 4.2.3

Tìm các giá trị kì dị và vector kì dị của ma trận sau:

$$H = \begin{pmatrix} 12 & 19 & 18 & 5 & 17 & 3 & 10 & 10 & 20 \\ 14 & 1 & 2 & 12 & 13 & 19 & 15 & 12 & 9 \\ 14 & 14 & 7 & 19 & 1 & 2 & 3 & 10 & 7 \\ 15 & 2 & 1 & 6 & 4 & 7 & 19 & 2 & 19 \\ 7 & 18 & 1 & 14 & 11 & 3 & 5 & 1 & 1 \\ 18 & 4 & 5 & 15 & 5 & 12 & 2 & 6 & 19 \\ 14 & 19 & 2 & 12 & 16 & 1 & 12 & 3 & 9 \\ 6 & 5 & 12 & 20 & 7 & 15 & 17 & 7 & 7 \\ 17 & 2 & 4 & 20 & 12 & 5 & 6 & 4 & 17 \\ 14 & 4 & 15 & 9 & 10 & 17 & 20 & 5 & 14 \end{pmatrix}$$

Ta thu được kết quả:

```
>> [U, Sig, V] = SVD(H)

U =

    0.3744   -0.3492    0.6099    0.2486    0.3867   -0.2194    0.0202   -0.1895    0.2376    0.1168
    0.3347    0.3555   -0.0820   -0.2659   -0.2265   -0.6084   -0.4055   -0.2951    0.0414    0.0977
    0.2673   -0.3159   -0.4117   -0.0308    0.3622    0.3513   -0.4908   -0.2715   -0.2832    0.0945
    0.2860    0.2999    0.1572    0.3521   -0.4432    0.5531   -0.1987   -0.0904    0.2900    0.2131
    0.2117   -0.4522   -0.1311   -0.3448   -0.2698   -0.0370    0.0631    0.4481    0.1857    0.5489
    0.3154    0.1185   -0.3648    0.4179    0.2144   -0.1928   -0.1267    0.5701    0.2600   -0.2953
    0.3127   -0.4249    0.1056   -0.0578   -0.5045    0.0170   -0.0003   -0.0239   -0.1938   -0.6435
    0.3266    0.2061   -0.0787   -0.5672    0.2661    0.2902    0.3204   -0.1195    0.4258   -0.2646
    0.3291    0.0182   -0.3816    0.3175   -0.0911   -0.1255    0.6545   -0.3189   -0.2223    0.2034
    0.3709    0.3464    0.3375   -0.1526    0.1384    0.1172    0.0613    0.3909   -0.6382    0.0922


Sig =

   97.6631         0         0         0         0         0         0         0         0
         0   31.2158         0         0         0         0         0         0         0
         0         0   23.7358         0         0         0         0         0         0
         0         0         0   21.8590         0         0         0         0         0
         0         0         0         0   17.5244         0         0         0         0
         0         0         0         0         0   12.7046         0         0         0
         0         0         0         0         0         0   10.1180         0         0
         0         0         0         0         0         0         0    7.1027         0
         0         0         0         0         0         0         0         0    4.4667
         0         0         0         0         0         0         0         0         0


V =

    0.4247    0.0089   -0.2306    0.3782   -0.1834   -0.0142   -0.3183    0.1178   -0.6886
    0.2720   -0.7493    0.1870   -0.1699   -0.0329    0.1418   -0.3469    0.3299    0.2248
    0.2332   -0.0145    0.3766   -0.0969    0.7588    0.1273    0.2698    0.0581   -0.3551
    0.4152   -0.1195   -0.7170   -0.2763    0.0895    0.2154    0.3704   -0.1269    0.1219
    0.3169   -0.2078    0.2608   -0.0992   -0.3193   -0.6673    0.4500   -0.1474   -0.0833
    0.2837    0.5090   -0.0617   -0.3429    0.0928   -0.3411   -0.2201    0.5729    0.1954
    0.3622    0.3207    0.4056   -0.3329   -0.4049    0.5037   -0.0464   -0.2655   -0.0175
    0.1999    0.0143   -0.0455   -0.0816    0.3183   -0.3072   -0.5435   -0.6607    0.1536
    0.4084    0.1371    0.1323    0.7053    0.0756    0.0862    0.1408    0.0306    0.5152
```

Ví dụ 4.2.4

Nén bức ảnh sau

Ta có kích thước của ảnh là:

```
>> size(img)

ans =

         612         1086            3
```

Thực hiện nén ảnh theo chương trình với lượng thông tin là 99,99%, Ta thu được độ dài của vector Σ:

```
>> size(Sig)

ans =

   197     1
```

Khi đó ta thu được ảnh sau khi nén với lượng thông tin ở trên là:

Ta nhận xét rằng với lượng thông tin là 99,99% lượng thông tin, ta không cần thiết phải lưu 612×1086 giá trị, mà thay vào đó, ta chỉ cần lưu 197 vector có độ dài 612×1, 197 vector có độ dài 1086×1, 1 vector có độ dài 197×1, giảm rất nhiều so với số lượng gốc mà vẫn giữ hầu hết các đặc trưng của ảnh sẵn có.

Ví dụ 4.2.5

Nén bức ảnh sau

Ta có kích thước của ảnh là:

```
>> size(img)

ans =

   337   599     3
```

Thực hiện nén ảnh theo chương trình với lượng thông tin là 99,99%, Ta thu được độ dài của vector Σ:

```
>> size(Sig)

ans =

   136     1
```

Khi đó ta thu được ảnh sau khi nén với lượng thông tin ở trên là:

Ta nhận xét rằng với lượng thông tin là 99,99% lượng thông tin, ta không cần thiết phải lưu 337×599 giá trị, mà thay vào đó, ta chỉ cần lưu 136 vector có độ dài 337×1, 136 vector có độ dài 599×1, 1 vector có độ dài 136×1, giảm đáng kể so với số lượng gốc mà vẫn giữ hầu hết các đặc trưng của ảnh sẵn có.

Ví dụ 4.2.6

Giải hệ phương trình sau:

$$\begin{cases} x_1 + 3x_2 &= 17 \\ 5x_1 + 7x_2 &= 19 \\ 11x_1 + 13x_2 &= 23 \end{cases}$$

Ta đã biết rằng phường trình trên không có nghiệm, ta sẽ tìm được nghiệm x sao cho khoảng cách giữa Ax với y là nhỏ nhất, chạy ví dụ với chương trình, ta thu được kết quả như sau:

```
A =

     1     3
     5     7
    11    13


y =

    17
    19
    23

>> [Pinv] = pseudo_inv(A)
Ma tran nghich dao suy rong trai.

Pinv =

   -0.5197   -0.2171    0.2368
    0.4276    0.2039   -0.1316

>> x = Pinv*y

x =

   -7.5132
    8.1184
```

Ví dụ 4.2.7

Giải hệ phương trình sau:

$$\begin{cases} x_1 + x_2 & = 4 \\ x_1 - 2x_2 & = 5 \\ x_1 + 3x_2 & = -1 \\ x_1 + 4x_2 & = 1 \end{cases}$$

Ta thu được kết quả sau

```
A =

     1     1
     1    -2
     1     3
     1     4


y =

     4
     5
    -1
     1

>> [Pinv] = pseudo_inv(A)
Ma tran nghich dao suy rong trai.

Pinv =

    0.2857    0.5000    0.1429    0.0714
   -0.0238   -0.1667    0.0714    0.1190

>> x = Pinv*y

x =

    3.5714
   -0.8810
```

Ví dụ 4.2.8

Giải hệ phương trình sau:

$$\begin{cases} x_1 & = 2000 \\ x_1 + x_2 + x_3 & = 195 \\ x_1 + 2x_2 + 4x_3 & = 180 \\ x_1 + 4x_2 + 16x_3 & = 120 \\ x_1 + 6x_2 + 36x_3 & = 25 \end{cases}$$

Ta thu được kết quả sau

```
A =

     1     0     0
     1     1     1
     1     2     4
     1     4    16
     1     6    36


y =

        2000
         195
         180
         120
          25

>> [Pinv] = pseudo_inv(A)
Ma tran nghich dao suy rong trai.

Pinv =

    0.7878    0.3326    0.0128   -0.2207    0.0874
   -0.5229   -0.0043    0.3268    0.4259   -0.2255
    0.0677   -0.0107   -0.0581   -0.0602    0.0613

>> x = Pinv*y

x =

   1.0e+03 *

    1.6186
   -0.9424
    0.1172
```

Ví dụ 4.2.9

Xác định số điều kiện của ma trận trong hệ phương trình sau

$$\begin{cases} 10x_1 + 7x_2 + 8x_3 + 7x_4 & = 32 \\ 7x_1 + 5x_2 + 6x_3 + 5x_4 & = 23 \\ 8x_1 + 6x_2 + 10x_3 + 9x_4 & = 33 \\ 7x_1 + 5x_2 + 9x + 3 + 10x_4 & = 31 \end{cases}$$

Nghiệm của phương trình đã cho là:

```
>> x = inv(A)*y

x =

    1.0000
    1.0000
    1.0000
    1.0000
```

Số điều kiện của ma trận A là:

```
>> c = condition(A)

c =

   2.9841e+03
```

Ta thấy rằng số điều kiện của ma trận A khá là lớn, như vậy ma trận A là ma trận xấu hay ma trận không ổn định. Khi ta làm nhiễu giá trị của y rồi giải lại ta thu được nghiệm và nhiễu của x như sau:

```
>> y1 = y + [0.01,-0.01,0.01,+0.01]'

y1 =

   32.0100
   22.9900
   33.0100
   31.0100

>> x1 = inv(A)*y1

x1 =

    1.7000
   -0.1600
    1.2900
    0.8300
```

Ta nhận xét rằng, khi thay làm nhiễu y khoảng 1% thì x đã xuất hiện sự thay đổi rất là lớn.

Ví dụ 4.2.10

Xác định số điều kiện của ma trận trong hệ phương trình sau

$$\begin{cases} -13x_1 + 3x_2 & = -13 \\ 10x_1 - 12x_2 & = 10 \end{cases}$$

Dễ dàng nhận ra nghiệm của phương trình đã cho là: $x = [1, 1]'$ Số điều kiện của ma trận A là:

```
>> c = condition(A)

c =

    3.0178
```

Ta thấy rằng số điều kiện của ma trận A rất bé, như vậy ma trận A là ma trận tốt hay ma trận ổn định. Khi ta làm nhiễu giá trị của y rồi giải lại ta thu được nghiệm và nhiễu của x như sau:

```
>> y1 = y + [0.1,-0.1]'

y1 =

  -12.9000
    9.9000

>> x1 = inv(A)*y1

x1 =

    0.9929
    0.0024
```

Ta nhận xét rằng, khi thay làm nhiễu y khoảng 10% thì x đã xuất hiện sự thay đổi không đáng kể.

Tài liệu tham khảo

[1] Mechanical Engineering Analysis - Steve Brunton - Washington

[2] School of Computing (SoC) and Scientific Computing and Imaging Institute - PhD.Guido Gerig - University of Utah