

PHÂN RÃ MA TRẬN LU VÀ PHƯƠNG PHÁP LẬP TRÌNH GIẢI MÔ HÌNH TUYẾN TÍNH TRONG PHÂN TÍCH KINH TẾ

DISCUSSION AND PROGRAMMING METHODS IN ECONOMIC ANALYSIS

NGUYỄN VĂN LỘC^(*)

TÓM TẮT: Bài viết trình bày cách ứng dụng phương pháp sử dụng phân rã ma trận LU và phương pháp lập trình để giải hệ phương trình có hàng trăm phương trình tuyến tính trong phân tích kinh tế mà các phương pháp thông thường trong chương trình toán cao cấp khó giải được.

Từ khóa: lập trình; phân rã ma trận; hệ phương trình tuyến tính.

ABSTRACT: The paper presents the method of using LU matrix decomposition and programming method to solve the system of linear equations in economic analysis.

Key words: matrix decomposition; system of linear equations.

1. ĐẶT VẤN ĐỀ

Trong một nền kinh tế có hàng trăm sản phẩm, mỗi sản phẩm có hai phương trình cung và phương cầu nhằm xác định sản lượng cân bằng. Để xác định tất cả các sản lượng cân bằng của nền kinh tế, phục vụ cho việc nghiên cứu sản phẩm tiềm năng Yp, ta phải giải một hệ có đến hàng trăm phương trình tuyến tính. Các phương pháp toán học như: Phương pháp ma trận nghịch đảo; Phương pháp Cramer; Phương pháp Gauss... các phương pháp lập trình như: Hàm solve(); hàm inv(); hàm pinv()... là không thể giải. Dùng phân rã ma trận kết hợp với phương pháp lập trình để chuyển việc giải các hệ tuyến tính “phức tạp” về giải các hệ tuyến tính đơn giản hơn là việc làm rất cần thiết trong nghiên cứu kinh tế. Trong bài viết này, chúng tôi minh họa phương pháp trên thông qua cách sử dụng phân rã ma trận LU và phương pháp lập trình để giải các hệ phương trình tuyến tính trong phân tích kinh tế.

2. NỘI DUNG

2.1. Phân rã ma trận $A=LU$ (LU - Decomposition) trong Toán học

2.1.1. Phân rã ma trận (Matrix Decomposition)

Phân rã ma trận là cách giảm ma trận thành các thành phần cấu thành của nó. Phân rã ma trận là một cách tiếp cận có thể đơn giản hóa các tính toán ma trận phức tạp vì được thực hiện trên ma trận phân tách thay vì trên chính ma trận gốc. Phân rã ma trận tương tự như phân tách (factor) các số: ví dụ như factor 10 thành 2×5 . Giống như phân tích các giá trị thực, có nhiều cách để phân tách một ma trận và một loạt các kỹ thuật phân rã ma trận khác nhau. Trong bài viết này, chúng tôi trình bày phương pháp phân rã ma trận đơn giản, được sử dụng rộng rãi là phân rã ma trận LU. Phép phân rã LU - Decomposition dùng cho ma trận vuông, nó phân tách ma trận thành các thành phần L và U.

Công thức: $A = L.U$ hoặc $A = LU$

Với A là ma trận vuông muốn phân rã; L là ma trận tam giác dưới; U là ma trận tam giác trên. Điều kiện áp dụng phân rã LU: A khả nghịch, $|A| \neq 0, \forall k \leq n$

2.1.2. Phương pháp phân rã ma trận

Biểu diễn A dưới dạng tích hai ma trận đồng cấp với A: L là ma trận tam giác trên với các phần tử trên đường chéo chính bằng 1, các

^(*) PGS.TS. Trường Đại học Văn Lang, loc.nv@vlu.edu.vn, Mã số: TCKH27-07-2021

phần tử đối diện “phần 0” cần xác định và U là ma trận tam giác dưới với các phần tử trên đường chéo chính và các phần tử đối diện “phần 0” cần xác định;

Thực hiện phép nhân LU; + Đồng nhất hóa các phần tử hai ma trận cùng cấp LU và ma trận A, ta được hệ đẳng thức về sự bằng nhau của các phần tử tương ứng của hai ma trận.

Từ hệ đẳng thức được xác định, ta tìm được các phần tử của L và U.

Ví dụ: Phân rã LU - Decomposition ma trận:

$$A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = LU; L = \begin{bmatrix} 1 & 0 & 0 \\ L_{21} & 1 & 0 \\ L_{31} & L_{32} & 1 \end{bmatrix}$$

$$U = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

Giải: Ta có:

$$LU = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ L_{21}U_{11} & L_{21}U_{12} + U_{22} & L_{21}U_{13} + U_{23} \\ L_{31}U_{11} & L_{31}U_{12} + L_{32}U_{22} & L_{31}U_{13} + L_{32}U_{23} + U_{33} \end{bmatrix} \\ = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} \quad (1)$$

Đồng nhất hóa các phần tử tương ứng của hai ma trận từ (1), ta có:

$$U_{11} = 1; U_{12} = 2; U_{13} = 4.$$

$$\Rightarrow L_{21} = 3; U_{22} = 2; U_{23} = 2; L_{31} = 2; L_{32} = 1; U_{33} = 3.$$

$$\Rightarrow A = \begin{bmatrix} 1 & 2 & 4 \\ 3 & 8 & 14 \\ 2 & 6 & 13 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 4 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{bmatrix}$$

2.2. Phân rã ma trận PA = LU bằng phương pháp lập trình

Việc phân rã LU được thực hiện bằng cách sử dụng một quá trình số lặp và có thể thất bại đối với các ma trận không thể phân tích. Do vậy, để phân tích ma trận ổn định về mặt số, ta dùng phân tích LU xoay vòng một phần (Partial Pivoting) [1, tr.6].

Công thức: PA = LU

Với mọi ma trận vuông A: PA = LU. Với L là ma trận tam giác dưới; U là ma trận tam giác trên.

P là ma trận hoán vị (permutation matrix). Mỗi dòng, mỗi cột có một hệ số bằng 1, tất cả các hệ số khác bằng 0. Hoán vị các dòng của I hay các cột chuẩn.

Tính chất của ma trận P:

$$a). |P| = \pm 1; b). P.P^T = P^T.P = I; c). P^{-1} = P^T$$

$$\text{Ví dụ: Cho ma trận } A = \begin{bmatrix} 3 & 2 & 5 & 9 \\ 1 & 7 & 5 & 7 \\ 5 & 10 & 5 & 3 \\ 6 & 9 & 7 & 2 \end{bmatrix}$$

a) Phân tích ma trận A thành các thành phần P, L, U. In ra kết quả.

b) Tái tạo (phục hồi) ma trận B từ các thành phần P, L, U.

Giải:

```
import numpy
Entrée[1]: import scipy
           from scipy.linalg import lu
A = numpy.array([[3., 2., 5., 9.], [1.,
Entrée[2]: 7., 5., 7.], [5., 10., 5., 3.], [6., 9., 7.,
           2.]])
Entrée[3]: P, L, U = lu(A)
Entrée[4]: P
Out [4]: array([[0., 0., 1., 0.], [0., 1., 0., 0.],
           [0., 0., 0., 1.], [1., 0., 0., 0.]])
Entrée[5]: L
           array([[1., 0., 0., 0.], [0.16666667, 1., 0.,
Out [5]: 0.], [0.5, -0.45454545, 1., 0.], [0.83333333,
           0.45454545, -0.79439252, 1.]])
Entrée[6]: U
           array([[6., 9., 7., 2.], [0., 5.5, 3.83333333,
Out [6]: 6.66666667], [0., 0., 3.24242424,
           11.03030303], [0., 0., 0., 7.06542056]])
B = P.dot(L).dot(U)
Entrée[7]: B
           array([[3., 2., 5., 9.],
Out [7]: [1., 7., 5., 7.],
           [5., 10., 5., 3.],
           [6., 9., 7., 2.]])
```

2.3. Ứng dụng phân rã ma trận PA = LU và phương pháp lập trình giải hệ phương trình tuyến tính

$$AX = (1) \Leftrightarrow PAX = LUX = PB = B'$$

$$\Leftrightarrow \begin{cases} LY = B'(2) \Leftrightarrow Y = L^{-1}.B' \\ UX = Y(3) \Leftrightarrow X = U^{-1}.Y \end{cases}$$

Thay vì giải hệ phương trình (1), ta lần lượt:

Bước 1: Giải hệ phương trình (2), tìm Y,

với $Y = L^{-1}.B'$

Bước 2: Giải hệ phương trình (3), tìm X,

với $X = U^{-1}.Y$

Ví dụ: Giải hệ phương trình:

$$\begin{cases} x_1 + 2x_2 + 3x_3 - 2x_4 = 6 \\ 2x_1 - x_2 - 2x_3 - 3x_4 = 8 \\ 3x_1 + 2x_2 - x_3 + 2x_4 = 4 \\ 2x_1 - 3x_2 + 2x_3 + x_4 = -8 \end{cases} \text{ Theo [2, tr.86]}$$

Chuyển hệ phương trình về dạng phương trình ma trận: $AX = B$, với:

$$A = \begin{bmatrix} 1 & 2 & 3 & -2 \\ 2 & -1 & -2 & -3 \\ 3 & 2 & -1 & 2 \\ 2 & -3 & 2 & 1 \end{bmatrix}; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}; B = \begin{bmatrix} 6 \\ 8 \\ 4 \\ -8 \end{bmatrix}$$

Giải: (Sử dụng phân rã $PA = LU$).

```
Entrée[1]: from numpy import array
           from scipy.linalg import lu
           A = array([[1, 2, 3, -2], [2, -1, -2, -
Entrée[2]: 3], [3, 2, -1, 2], [2, -3, 2, 1]])
           print(A)
           [[ 1  2  3 -2]
            [ 2 -1 -2 -3]
            [ 3  2 -1  2]
            [ 2 -3  2  1]]
           P, L, U = lu(A)
Entrée[3]: C=P.dot(L).dot(U)
           print(C)
           [[ 1.  2.  3. -2.]
            [ 2. -1. -2. -3.]
            [ 3.  2. -1.  2.]
            [ 2. -3.  2.  1.]]
Entrée[4]: print(P)
           [[ 1.  2.  3. -2.]
            [ 2. -1. -2. -3.]
            [ 3.  2. -1.  2.]
            [ 2. -3.  2.  1.]]
Entrée[5]: print(L)
           [[ 1.  0.  0.  0.] [0.66666667 1.  0.  0.]
            [0.33333333-0.30769231 1.  0.]
            [0.66666667 0.53846154-0.66666667 1.]]
```

```
Entrée[6]: print(U)
           [[ 3.  2. -1.  2.] [ 0. -4.33333333
            2.66666667-0.33333333] [0.  0.
            4.15384615-2.76923077] [0.  0.  0. -6.]]
           import numpy as np
Entrée[7]: B = np.array([[6], [8], [4], [-8]])
           B
Out [7]: array([[ 6], [ 8], [ 4], [-8]])
           D = P @ B
Entrée[8]: D
           D
Out [8]: array([[ 4.], [-8.], [ 6.], [ 8.]])
           from numpy.linalg import inv
Entrée[9]: K = inv(L)
           K
           array([[ 1.,  0.,  0.,  0.], [-0.66666667,
            1.,  0.,  0.], [-0.53846154, 0.30769231, 1.,
            0.], [-0.66666667, -0.33333333,
            0.66666667, 1.]])
Out [9]: Y = K @ D
           Y
Entrée[10]: array([[4.], [-10.66666667], [1.38461538],
            [12.]])
Out [10]: E = inv(U)
           E
Entrée[11]: array([[ 0.33333333, 0.15384615, -
            0.01851852,  0.11111111], [-0., -
            0.23076923, 0.14814815, -0.05555556],
            [ 0., 0., 0.24074074, -0.11111111], [-0., -
            0., -0., -0.16666667]])
Out [11]: X = E.dot(Y)
           X
Entrée[12]: array([[ 1.], [ 2.], [-1.], [-2.]])
Out [12]:
```

Vậy nghiệm của hệ phương trình là :

$$\begin{cases} x_1 = 1 \\ x_2 = 2 \\ x_3 = -1 \\ x_4 = -2 \end{cases}$$

2.4. Ứng dụng phân rã ma trận $PA = LU$ và phương pháp lập trình giải mô hình tuyến tính kinh tế

Mô hình tuyến tính trong phân tích kinh tế giải bằng phân rã LU và phương pháp lập trình theo quy trình 3 bước như mô tả trong ví dụ sau:

Ví dụ: Xét thị trường chè, cà phê, ca cao có hàm cung và hàm cầu tương ứng như sau:

$$Q_{S_1} = -10 + P_1; \dots Q_{D_1} = 20 - P_1 - P_3 \dots (\text{che})$$

$$Q_{S_2} = 2P_2; \dots Q_{D_2} = 40 - 2P_2 - P_3 \dots (\text{cafe})$$

$$Q_{S_3} = -5 + 3P_3; \dots Q_{D_3} = 10 + P_2 - P_3 - P_1 \dots (\text{caocao})$$

Hãy thiết lập mô hình cân bằng thị trường của ba loại hàng hóa trên. Xác định giá và lượng hàng hóa của mỗi loại ở trạng thái cân bằng thị trường [3, tr.84].

Giải: Bước 1: Thiết lập mô hình cân bằng thị trường:

$$\begin{cases} Q_{S_1} = Q_{D_1} \\ Q_{S_2} = Q_{D_2} \\ Q_{S_3} = Q_{D_3} \end{cases} \Leftrightarrow \begin{cases} -10 + P_1 = 20 - P_1 - P_3 \\ 2P_2 = 40 - 2P_2 - P_3 \\ -5 + 3P_3 = 10 + P_2 - P_3 - P_1 \end{cases}$$

$$\Leftrightarrow \begin{cases} 2P_1 + P_3 = 30 \\ 4P_2 + P_3 = 40 \\ P_1 - P_2 + 4P_3 = 15 \end{cases}$$

Dạng ma trận $AX = B$ với:

$$A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 4 & 1 \\ 1 & -1 & 4 \end{bmatrix}; X = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix}; B = \begin{bmatrix} 30 \\ 40 \\ 15 \end{bmatrix}$$

Bước 2: Giải mô hình bằng phương pháp lập trình, sử dụng phân rã ma trận $PA = LU$

Entrée [1]: # Giải mô hình tuyến tính bậc 3_trong kinh tế :

$$\# \begin{cases} 2P_1 + P_3 = 30 \\ 4P_2 + P_3 = 40 \\ P_1 - P_2 + 4P_3 = 15 \end{cases}$$

Entrée[2]:

```
from numpy import array
from scipy.linalg import lu
A = array([[2, 0, 1], [0, 4, 1], [1, -1, 4]])
```

Entrée[3]:

```
print(A)
[[ 2  0  1]
 [ 0  4  1]
 [ 1 -1  4]]
P, L, U = lu(A)
```

Entrée[4]:

```
C = P.dot(L).dot(U)
print(C)
[[ 2.  0.  1.]
 [ 0.  4.  1.]
 [ 1. -1.  4.]]
```

Entrée[5]:

```
print(P)
```

```
[[1.  0.  0.]
 [0.  1.  0.]
 [0.  0.  1.]]
Entrée[6]: print(L)
[[ 1.  0.  0.]
 [ 0.  1.  0.]
 [ 0.5 -0.25  1. ]]
Entrée[7]: print(U)
[[2.  0.  1.]
 [0.  4.  1.]
 [0.  0.  3.75]]
import numpy as np
Entrée[8]: B = np.array([[30], [40], [15]])
B
Out [8]: array([[30], [40], [15]])
D = P @ B
Entrée[9]: D
Out [9]: array([[30.], [40.], [15.]])
from numpy.linalg import inv
Entrée[10]: K = inv(L)
K
Out [10]: array([[ 1.,  0.,  0.], [0.,  1.,  0.], [-0.5, 0.25, 1.]])
Y = K @ D
Entrée[11]: Y
Out [11]: array([[30.], [40.], [10.]])
E = inv(U)
Entrée[12]: E
Out [12]: array([[ 0.5,  0., -0.13333333], [0.,  0.25, -0.06666667], [0.,  0.,  0.26666667]])
X = E.dot(Y)
Entrée[13]: X
Out [13]: array([[13.66666667], [9.33333333], [2.66666667]])
```

Bước 3: Phát biểu kết quả “kinh tế” của bài toán: Giá và lượng cân bằng tương ứng của mỗi loại hàng hóa là:

$$\overline{P_1} = 13,66, \overline{Q_{D_1}} = \overline{Q_{S_1}} = -10 + 13,66 = 3,66;$$

$$\overline{P_2} = 9,33, \overline{Q_{D_2}} = \overline{Q_{S_2}} = 2.9,33 = 18,66;$$

$$\overline{P_3} = 2,66, \overline{Q_{D_3}} = \overline{Q_{S_3}} = -5 + 3.2,66 = 2,98$$

3. KẾT LUẬN

Hệ phương trình tuyến tính có ứng dụng rộng rãi trong các bài toán tìm vec-tơ riêng, giá trị riêng của ánh xạ tuyến tính, trong các hệ mô

hình hóa tuyến tính trong kinh tế... Do vậy, việc tìm các phương pháp hữu hiệu giải các hệ tuyến tính có số ẩn và số phương trình “rất lớn” luôn là nhu cầu cần thiết. Phân rã ma trận LU kết hợp với phương pháp lập trình cho ta một phương pháp giải hiệu quả các phương trình

trên. Từ những kết quả đã trình bày còn cho thấy sự cần thiết của việc kết hợp nghiên cứu phát triển các phương pháp Toán học và phương pháp lập trình trong giải các mô hình tuyến tính trong kinh tế và mở rộng cho các mô hình kinh tế khác.

TÀI LIỆU THAM KHẢO

- [1] Trung tâm tin học, Đại học Khoa học Tự nhiên (2019), *Mathematics and Statistics for Data Science*, Tài liệu lưu hành nội bộ.
- [2] Nguyễn Đình Trí (2014), *Bài tập Toán cao cấp tập 1*, Nxb Giáo dục Việt Nam.
- [3] Nguyễn Tiến Quang (2014), *Cơ sở Đại số tuyến tính*, Nxb Giáo dục Việt Nam.

Ngày nhận bài: 22-02-2021. Ngày biên tập xong: 05-5-2021. Duyệt đăng: 20-5-2021